Reconocimiento de la Expresión Facial en Secuencias de Video usando parámetros basados en LBP

Edwin Alberto Silva Cruz

Universidad Industrial de Santander
Facultad de Ingenierías Físico Mecánicas
Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones
Bucaramanga
2015

Reconocimiento de la Expresión Facial en Secuencias de Video usando parámetros basados en LBP

Edwin Alberto Silva Cruz

Trabajo de grado para optar al título de Doctor en Ingeniería, Área Electrónica

PhD. Arturo Plata Gómez, Director

Universidad Industrial de Santander
Facultad de Ingenierías Físico Mecánicas
Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones
Bucaramanga
2015

Resumen

Título: Reconocimiento de la expresión facial usando parámetros basados en LBP[1]

Autor: Edwin Alberto Silva Cruz

Palabras claves: Reconocimiento expresión facial, aprendizaje de máquina, LBP, POEM, TPOEM

En esta disertación se diseñó e implementó un sistema de reconocimiento de la expresión facial que realiza detección de rostro, extracción de parámetros, selección de parámetros y clasificación de la expresión con algoritmos eficientes de bajo costo de cálculo y memoria, con el fin de su aplicación en sistemas en tiempo real. Los parámetros extraídos son dinámicos, basados en patrones locales binarios y POEM (*Patterns of Oriented Edge Magnitudes*). El trabajo incluye la implementación de los algoritmos SFA-WM (*Sequential Feature Analysis for extraction of Weak Metaclassifiers*), que es un aporte original a la búsqueda de parámetros débiles, el algoritmo LC-NNMLE (*Local Clustering-Nearest Neighbor MLE*), para estimación de dimensión intrínseca, los códigos VPOEM y TPOEM, que probaron ser descriptores adecuados de la expresión facial. Además de pruebas de validación cruzada LSO, se realizaron pruebas de generalización entre bases de datos usando la base de datos KDEF. Los resultados fueron comparados con resultados del estado del arte, que muestran la validez de los parámetros y los sistemas de selección y clasificación con desempeño similar o superior al de la mayor parte de trabajos distinguidos en la bibliografía del tema.

[1]Trabajo de grado, Facultad de Ingenierías Físico Mecánicas, Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones, Doctorado en Ingeniería, área Electrónica

Abstract

Title: Facial Expression Recognition using LBP-based Parameters[2]

Author: Edwin Alberto Silva Cruz

Keywords: Facial expression recognition, machine learning, LBP, POEM, TPOEM

In this thesis a facial expression recognition system was designed and implemented. The system includes facial detection, parameter extraction, feature selection and expression classification using efficient algorithms in memory and calculation costs, which makes them viable for potential real time applications. The extracted parameters are dynamic, based on local binary patterns and POEM (Patterns of Oriented Edge Magnitudes). The work includes the implementation of the algorithms SFA-MW (Sequential Feature Analysis for extraction of Weak Metaclassifiers), which is an original contribution to the search of weak parameters; the algorithm LC-NNMLE (Local Clustering-Nearest Neighbor MLE), for the estimation of intrinsic dimension of clustered data in high dimensionality spaces, and the codification VPOEM and TPOEM, which proved to be efficient descriptors of facial expression. Additionally, besides crossed validation LSO, further tests were made for generalization of description using the KDEF database. The results were compared with state of the art results, showing the validity of the parameters and the classification system had similar or superior to most of the most recognized works in the bibliography.

[2]Trabajo de grado, Facultad de Ingenierías Físico Mecánicas, Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones, Doctorado en Ingeniería, área Electrónica

CONTENIDO

pág.

LISTA DE FIGURAS

pág.

LISTA DE TABLAS

pág.

1. Introducción

El reconocimiento de la expresión facial es un problema de interés creciente, sobre todo en los últimos años. Esto obedece al desarrollo de sistemas de interacción hombre-máquina de naturaleza más compleja, que requieren extraer la mayor cantidad de información posible del proceso de comunicación humana. Si bien la comunicación verbal usualmente contiene la mayor cantidad de información, la interacción humana es multimodal. Esto es, en un proceso de comunicación buena parte de la información es representada visualmente a través de gestos y expresiones y auditivamente a través del sonido y la entonación [18]. El reconocimiento de la expresión facial es una actividad realizada cotidianamente por los individuos, de manera recurrente y sin esfuerzo. De acuerdo con Mehrabian [108, 110], la contribución de la información verbal en la comunicación cara a cara está generalmente limitada a sólo el 7 % de la información total, mientras que señales tales como la acentuación, puntuación y ritmo contribuyen el 38 % y la expresión facial contribuye 55 %. Esta relación 55-38-7 ha sido discutida y cuestionada por Lappako [88], Hergstrom [67] y Burgoon [19], e incluso Mehrabian no implica que la información verbal exclusivamente contiene sólo el 7 % de la información global [109]. No obstante, ningún autor desprecia el importante aporte de la información no verbal en el proceso de comunicación.

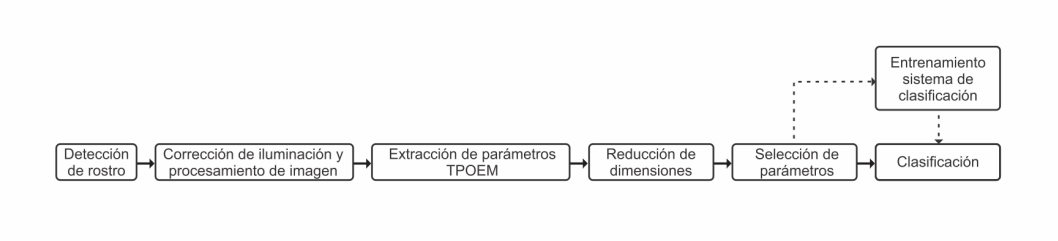
Debido a estas consideraciones y gracias al desarrollo de sistemas de computación y algoritmos para procesamiento de imágenes digitales, las condiciones para el desarrollo de sistemas automáticos de reconocimiento de la expresión facial son más propicias. De hecho, múltiples estudios [15, 30, 20] muestran que el reconocimiento de la expresión facial es importante en el desarrollo de sistemas de interacción hombre-máquina, por cuanto las expresiones faciales desempeñan un papel esencial en la coordinación del habla humana, a partir de información no verbal incorporada en la observación e interpretación de las expresiones faciales de otros individuos.

1.1. Descripción del sistema

En general, el reconocimiento de la expresión facial es un problema que puede ser formulado de la siguiente manera: a partir de una imagen o una secuencia de imágenes de un rostro, el objetivo es identificar la expresión facial realizada por el individuo o la neutralidad de la expresión, usando algún criterio de comparación, tal como los parámetros definidos para la universalidad de la expresión facial determinados por Ekman [44]. No obstante, el nivel de abstracción de la expresión facial desde una perspectiva humana no es igual que el nivel de abstracción que puede realizar un sistema de cómputo, de modo que debe haber una traducción de lenguaje. Esto es, determinar parámetros obtenidos por el sistema de cómputo que sean descriptores adecuados que permitan caracterizar y distinguir las expresiones faciales.

De manera global, un sistema de reconocimiento de la expresión facial se puede separar en distintas etapas: i. detección del rostro, ii. procesamiento de imagen, normalización geométrica y corrección de iluminación, iii. extracción de parámetros de

Figura 1. Sistema global de reconocimiento de expresión facial



expresión facial, iv. procesamiento de datos (reducción de dimensiones, selección de parámetros) y v. clasificación. El sistema completo se muestra en la figura 1.

La detección de rostro es la primera etapa del sistema completo. Se refiere a determinar la presencia de un rostro humano en la imagen o cuadro de video y obtener su localización espacial. En el entorno de este trabajo, este problema es además restringido por el requerimiento de cómputo en tiempo real, por cuanto algoritmos complejos pueden determinar con alta precisión y baja tasa de error la localización de rostros en una imagen. No obstante, la disponibilidad total de tiempo de cálculo para todas las etapas del sistema está limitada por el tiempo real requerido para el procesamiento. La segunda etapa se refiere a la manipulación de la imagen facial que facilite la extracción de parámetros adecuados para la representación de la expresión facial. En general, corresponde a corrección de iluminación, eliminación de artefactos y alineación geométrica. La tercera etapa del proceso es la extracción de parámetros. Los parámetros son datos obtenidos por el sistema que permiten representar la expresión facial, por ejemplo descriptores de formas y texturas, que si bien no necesariamente tienen un nivel de abstracción equivalente al humano, en principio deben cumplir el mismo objetivo: obtener datos que permitan separar entre las clases (expresiones) para realizar el reconocimiento de la expresión facial. En tanto que los datos extraídos pueden ser de alta dimensión dependiendo del tipo de codificación usado, en la siguiente etapa se realiza reducción de dimensiones, consistente en la proyección de los datos de alta dimensión a un espacio de menor dimensionalidad en el cual la información pertinente para la discriminación de la expresión facial sea preservada. En la quinta etapa se realiza selección de parámetros. Debido a que en la metodología seguida en este trabajo los parámetros extraídos se refieren a celdas faciales espaciales, es probable que el aporte de algunas de estas celdas no sea relevante, de manera que pueden ser descartadas para la etapa de selección. Hay que señalar, sin embargo, que esta etapa depende de la extracción de parámetros y de los resultados de clasificación, de manera que no es un proceso lineal. Así mismo, una vez se determine la relevancia de los parámetros extraídos, este proceso no se realiza nuevamente, sino que en la extracción de parámetros no se calculan parámetros que se determinaron irrelevantes. La última etapa es la clasificación, que corresponde al conjunto de algoritmos usados para determinar la expresión facial o neutralidad en la muestra de entrada, a partir de los parámetros de la muestra y los sistemas de clasificación desarrollados usando un subconjunto de entrenamiento sin traslape con el conjunto de datos usados para la validación.

Uno de los objetivos de este trabajo se refiere a la implementación de algoritmos que sean aplicables en tiempo real. Para este efecto, definimos tiempo real como el tiempo equivalente a la duración de la secuencia de video, de manera que el algoritmo esté en capacidad de detectar la expresión facial para cada cuadro diezmado antes de que el siguiente cuadro diezmado sea capturado/visualizado. Nuestro algoritmo TPOEM realiza diezmado 5 ó 7, equivalente a 167ms a 233ms, de manera que el algoritmo final completo debe estar en capacidad de realizar el procedimiento completo en esta disponibilidad de tiempo.

1.2. Descripción de la problemática

El reconocimiento de la expresión facial tiene numerosos desafíos por cuanto los rostros son objetos tridimensionales deformables y en general la información digital real es representada por imágenes o secuencias de video en dos dimensiones. Adicionalmente, si bien la expresión facial definida por Ekman pretende describir las características de la expresión facial, en realidad las variaciones interpersonales de la expresión facial son significativas, con frecuencia mayores que las variaciones determinadas por la calidad de la imagen (iluminación, resolución y presencia de artefactos) y las variaciones entre las clases. Por otra parte, existe un número limitado de bases de datos estandarizadas de la expresión facial, debido principalmente a la dificultad de la obtención de este tipo de bases de datos, ya que requieren de un conjunto preferiblemente numeroso de individuos, la realización artificial de la expresión facial con frecuencia no es trivial [1] y la validación estandarizada de las muestras obtenidas requiere de individuos entrenados altamente calificados. Adicionalmente, buena parte de los trabajos se basan en la presunción de la universalidad de la expresión facial, pero esto no es acogido por toda la comunidad científica. Por último, así como aún no es claro cómo el humano realiza la clasificación de la expresión facial, los descriptores numéricos obtenidos por un sistema automático no tienen similares niveles de abstracción, de modo que aún no hay consenso sobre qué tipos de descriptores son más adecuados para realizar la discriminación de la expresión facial.

1. *Variaciones intraclase e interclase:* En un problema ideal de clasificación, las muestras pertenecientes a distintas clases son perfectamente diferenciables de las muestras de otras clases usando uno o más descriptores obtenidos de las clases. No obstante, en un problema de reconocimiento de la expresión facial las variaciones entre individuos pueden ser considerablemente más grandes que las variaciones entre clases. Desde un punto de vista numérico (por ejemplo variaciones entre pixeles equivalentes), puede haber mucha más variación entre el rostro de un hombre con expresión de ira y una chica con la misma expresión que con el

[1] En [160] se mostró la diferencia notable entre las expresiones faciales naturales y las impostadas, y en [98], donde se hace la revisión de la base de datos extendida CK+, hay en promedio 4 muestras por individuo, de un total de 6 expresiones faciales, lo que implica que alrededor del 60 % de las expresiones impostadas fueron eliminadas en tanto que no eran representativas de la expresión facial etiquetada.

mismo hombre realizando la expresión de tristeza. De hecho, la mayor parte de descriptores adecuados para identificación facial tienen esta función: minimizar la distancia entre muestras del mismo individuo mientras que se maximizan las distancias con individuos distintos incluso si realizan una expresión facial similar. Desde un punto de vista de identificación facial, las variaciones entre individuos son deseables, por cuanto el problema justamente se refiere a distinguir entre dos o más individuos. Pero en un problema de reconocimiento de la expresión facial se requiere de lo contrario: minimizar las distancias entre individuos, maximizando las distancias entre las expresiones, de modo que el tipo de descriptores requeridos es, necesariamente, de naturaleza muy distinta. Adicionalmente, a diferencia de un problema de identificación facial, la expresión facial generalmente es representada por microexpresiones que modifican ligeramente el rostro a partir de la instancia neutral. Esto, además de constituir un problema de distancia entre individuos y distancia entre clases, además es un problema dependiente de la calidad de la imagen. Una imagen de resolución mediana o cuyo procesamiento atenúa las micro expresiones, como las arrugas del ceño fruncido, constituye un reto para la representación de expresiones tales como ira o tristeza. Es decir, nuevamente a diferencia de un problema de identificación facial en el cual es deseable que los parámetros describan la generalidad geométrica del rostro, en un problema de reconocimiento de la expresión facial los parámetros deben ser capaces de describir los micro gestos que generalmente son útiles en la diferenciación entre expresiones.

2. *Disponibilidad de bases de datos:* En la actualidad, las bases de datos estandarizadas de la expresión facial son limitadas, siendo las más representativas la base de datos CK+ [98],MMI [124] y JAFFE [101]. Estas tres bases de datos tienen en total menos de 200 individuos y aproximadamente 4 muestras en promedio por individuo, por cuanto buena parte de las muestras fueron eliminadas en el proceso de estandarización. Es decir, hay expresiones cuya representación en total en las bases de datos es considerablemente limitada. Adicionalmente, las bases de datos no son universales, debido a la demografía de los individuos (en general no hay individuos niños o adolescentes ni personas de alta edad, con limitada representación de minorías raciales y, en el caso de la base de datos JAFFE, únicamente mujeres jóvenes). Existen otras bases de datos de expresión no estandarizadas, pero la no estandarización es un problema relevante, por cuanto un buen número de las muestras no validadas no son representativas de la expresión facial etiquetada, lo cual constituye un doble problema de validación (cuando se valida con una muestra no representativa, es probable que un sistema automático clasifique la muestra como perteneciente a otra clase, con error en la matriz de confusión sin que sea error real) o en el entrenamiento (al entrenar con muestras no representativas, el sistema se conduce a error, por cuanto es factible que posteriormente clasifique muestras similares como pertenecientes a la clase de la muestra etiquetada incorrectamente, generando, así, múltiples errores de validación), de modo que este tipo de bases de datos no es muy adecuado para el desarrollo del sis-

tema. Sin embargo, en tanto que la mayor parte de trabajos usan descriptores multivariables con un elevado número de variables, esto implica que se requiere preferiblemente de una cantidad suficientemente numerosa de muestras para poder describir adecuadamente las clases, sobre todo teniendo en cuenta que los protocolos de validación implican el entrenamiento con una parte de las muestras y la validación con las muestras restantes, limitando así el conjunto de datos disponibles para entrenar el sistema.

3. *Universalidad de la expresión facial:* Generalmente se da como un hecho la universalidad de la expresión facial. Sin embargo, esta presunción no es en absoluto acogida por toda la comunidad científica. Darwin es el exponente inicial de la teoría de la universalidad de la expresión facial en [33]. Paul Ekman, el investigador de expresión facial más reconocido en el entorno académico, es uno de los principales defensores de esta postulación, con trabajos que fundamentan la propuesta en [45, 46]. Sin embargo, hay críticas acerca de la validez de estos estudios. La primera crítica se refiere a que la universalidad tiene como fundamento que los individuos tengan capacidad de reconocer expresiones espontáneas, pero los estudios fueron realizados con muestras actuadas [104, 116]. Por otra parte, los estudios generalmente son realizados mostrando al individuo distintas imágenes de expresión facial, con selección forzada. Sin embargo, esto es distinto al proceso natural de reconocimiento de expresión, por cuanto el humano no realiza reconocimiento de expresión facial a partir de muestras de distintas expresiones sino en relación a experiencias previas [158] y sin selección forzada [171]. Es decir, un individuo espontáneamente puede atribuir neutralidad a una expresión ligera, mientras que en selección forzada le atribuye expresión. Esto es fundamentado al revisar las tasas de reconocimiento humano de expresión facial, con relativamente baja tasa de reconocimiento de instancia neutral, debido a que los individuos tienden a asignar expresión a muestras que en otras circunstancias no etiquetarían de igual manera, tal como se muestra en [99] y como determinamos en nuestras propias pruebas mostradas en el capítulo 7, tabla 41. Si bien parte de estos problemas es inevitable, por cuanto en principio es imposible obtener muestras de expresión espontánea validadas en cantidad suficiente para realizar el trabajo, esto muestra que la expresión no necesariamente es universal y, en consecuencia, los resultados de validación con muestras distintas no representadas en las bases de datos estandarizadas, tales como de minorías raciales o culturales, pueden ser afectados.

4. *Abstracción de los descriptores:* Aún no existe claridad acerca del proceso realizado por los humanos al desempeñar reconocimiento de la expresión facial. Naturalmente, hay un buen componente determinado por la geometría de los objetos faciales, tal como se muestra en [21] al hacer representación básica de las expresiones mediante estructuras faciales. De hecho, al generar expresiones a partir de deformación geométrica y de textura de rostros neutrales y transformaciones entre expresiones, como en [22, 34, 128], se encontró que los humanos y los siste-

mas de clasificación basados en redes neuronales están en capacidad de realizar una discriminación y, de hecho, encontrar una frontera ambigua en la transformación dinámica artificial entre dos expresiones. Esto es un resultado esperado, por cuanto en una imagen o secuencia de video la información representada es de esta naturaleza. Sin embargo, esto aún no considera el nivel de abstracción realizado por el humano. En [103] se mostró cómo tanto los humanos como los sistemas de máquina tienen mejor capacidad de reconocimiento y de tiempo de respuesta en secuencias de video cuando la expresión involucra una mayor cantidad de movimiento de músculos faciales. Esto sugiere que los sistemas que incluyan información dinámica de forma o textura deben tener una capacidad mayor de descripción de la expresión, que es un objetivo de este trabajo. No obstante, no existe consenso acerca de qué tipo de aproximación de descriptores es más adecuada para el reconocimiento de la expresión. Debido a esto, en la bibliografía se encuentran trabajos que usan metodologías diversas, por ejemplo descripción geométrica, descripción de apariencia, ondeletas Gabor, modelos activos de apariencia y textura, patrones locales binarios, descriptores de Fourier, entre otras muchas técnicas. Es decir, es un problema aún abierto con diversas posibilidades aún no resueltas.

1.3. Principales contribuciones

El primer aporte importante de este trabajo es el diseño de un sistema de detección de rostro con ventanas tipo Haar y clasificación *boosted* usando una metodología distinta a la convencional de clasificadores débiles independientes. En vez de usar esta aproximación que implica redundancia, cada banco de clasificadores fue obtenido a partir de la salida de los bancos de clasificación precedentes. Es decir, un banco de clasificación posterior no necesariamente es útil para rechazar muestras que hubiesen sido en cualquier caso rechazadas en una etapa precedente. De esta forma cada nivel de filtrado con ventanas Haar es especializado, enfocándose únicamente en las muestras negativas que aún no han podido ser rechazadas por las etapas previas más simples de filtrado. Adicionalmente, se aprovechó la información estadística de presencia y localización de rostros en cuadros vecinos precedentes en las secuencias de video. En tanto que en general los cuadros vecinos en las secuencias son muy similares, se espera que los rostros permanezcan relativamente cercanos y las zonas sin rostro permanezcan estables. De esta forma no se hace búsqueda exhaustiva de rostros en todos los cuadros de video, sino que se usa una pirámide de filtrado truncada en zonas previamente rechazadas y una pirámide más simple en las zonas previamente aceptadas, reduciendo así el costo de cálculo de manera considerable salvo en secuencias con numerosos rostros de bajo tamaño que, en todo caso, no son del tipo de secuencias usado en este trabajo.

Posteriormente a la etapa de detección, se mostró cómo el uso de corrección de iluminación y procesamiento de imagen que permita obtener imágenes de mejor apariencia para un observador humano y probablemente para un sistema de reconocimiento

facial, en general no es conveniente para un sistema de reconocimiento de la expresión facial. En cambio, el sistema de filtrado con bancos por ecualización de contraste y transformación adaptativa de contraste limitado en cascada, con corrección final para atenuar variaciones de contraste, usado en nuestro trabajo, permitió obtener imágenes cuya calidad desde el punto de vista humano es defectuosa, pero permitieron aumentar considerablemente las tasas de clasificación usando distintas técnicas. En tanto que en la bibliografía no es común encontrar procesamiento digital del rostro previo a la extracción de parámetros para reconocimiento de expresión facial, en nuestra opinión la implementación de esta etapa constituye una herramienta adicional para mejorar la capacidad de discriminación, principalmente en métodos basados en parámetros por textura.

En este trabajo diseñamos las codificaciones VPOEM y TPOEM como descriptores dinámicos de la expresión facial, basados en los descriptores estáticos POEM [169]. Los descriptores POEM fueron diseñados inicialmente para el reconocimiento facial en imágenes estáticas. Sin embargo, en tanto que determinamos la gran capacidad de descripción de textura de los códigos POEM, generalmente más útil en caracterización de expresión que en caracterización de rostro, realizamos la extensión de esta metodología. Para ello usamos el principio POEM combinado con patrones locales binarios, pero la codificación no fue realizada de igual manera que en la codificación convencional, sino obtuvimos una tabla de codificación propia a partir de un conjunto de textura de entrenamiento. Posteriormente la idea fue extendida a tres dimensiones, incorporando la variable temporal. Algo similar se hizo en [194] con la codificación dinámica VLBP y LBP-TOP a partir de la codificación LBP convencional. Este trabajo mostró cómo la adición de información dinámica a los códigos LBP permite aumentar drásticamente su capacidad de descripción. No obstante, los códigos VLBP requieren de un elevado costo computacional y su longitud es considerable, de manera que nuestra aproximación para usar información dinámica es distinta, optimizando el cálculo por reciclaje de información cuadro por cuadro, usando una vecindad de tipo cilíndrica en vez de esférica y usando reducción de dimensión por codificación. Los códigos VPOEM y TPOEM desarrollados probaron ser efectivos descriptores de bajo costo, con capacidad de reconocimiento de la expresión facial incluso en condiciones que requieren de alta generalización, tales como la validación con muestras de una base de datos usando un sistema entrenado con otra base de datos, con desempeño similar o superior al de la mayor parte de resultados publicados en el estado de desarrollo teórico reciente, pese al uso de metodología LSO (*leave-subjects-out*) con una muestra por individuo por clase que, tal como mostramos en el capítulo 6, sección 6.4, es una metodología rigurosa, en desventaja con metodologías convencionales aleatorias cruzadas (*random n-folded*).

Los datos obtenidos por la codificación TPOEM son de considerable tamaño, pese a ser notablemente más cortos que los obtenidos por codificación VLBP. Sin embargo, es razonable considerar que existe información redundante o innecesaria, principalmente debido a la naturaleza redundante de los códigos basados en histogramas y a que los códigos son obtenidos a partir de celdas espaciales que no aportan igual información de expresión facial. No obstante, un protocolo de reducción de dimensiones básico, por

ejemplo usando PCA, no es de utilidad, por cuanto este tipo de reducción de dimensiones fue probado y eliminó buena parte de la información de textura representativa de la expresión facial. Debido a esto, en este trabajo desarrollamos el sistema novedoso LC-NNMLE (Local Clustering - Nearest Neighbor Maximum Likelihood Estimation) para estimación de dimensiones intrínsecas, mostrado en 4.3, basado en NNMLE. A diferencia del NNMLE convencional, la vecindad usada por punto no es estática, sino depende de la densidad de población alrededor del punto evaluado. Nuestras pruebas con datos simulados de alta dimensión, con distribución no homogénea, probaron que la estimación de dimensiones con LC-NNMLE fue más exitosa que usando NNMLE, especialmente cuando el número de dimensiones es elevado. Por otra parte, en este trabajo hicimos pruebas que permitieron mostrar cómo la reducción supervisada de dimensiones no necesariamente mejora el sistema global de clasificación. Por ejemplo, la reducción usando MCML convencional produjo datos reducidos, pero la tasa de clasificación usando estos datos fue notablemente inferior que usando datos completos sin reducir. En cambio, al usar reducción de dimensiones supervisada más cuidadosa al asignar ponderación específica a cada parámetro dependiendo de su capacidad de discriminación, los datos reducidos tuvieron mejor capacidad de clasificación que los datos completos.

Así mismo, en este trabajo desarrollamos el algoritmo novedoso SFA-WM (*Sequential Feature Analysis for extraction of Weak Metaclassifiers*), descrito en 5.3. Este algoritmo fue usado para reducir el número de parámetros TPOEM de 256 a 112 sin disminuir la tasa de reconocimiento de la expresión facial. A diferencia de los algoritmos de búsqueda convencional, nuestro algoritmo incluye información mutua entre parámetros débiles, de modo que parámetros que eventualmente no tienen capacidad individual de discriminación, no son descartados si se encuentran parámetros que permitan, en conjunto, mejorar la tasa de clasificación. Por otra parte, métodos de búsqueda exhaustiva a partir de 256 parámetros iniciales no son viables, dada la imposibilidad de realizar búsqueda completa con las posibles combinaciones de 256 parámetros. En cambio nuestro algoritmo probó ser de bajo costo de cálculo en comparación con otros algoritmos convencionales del estado del arte, obteniendo reducción drástica del número de parámetros, que a su vez implica reducción de costos de extracción de parámetros, reducción de dimensiones y clasificación.

La metodología general de validación usada en este trabajo es LSO (*leave-subjects-out*), tomando pliegues basados en individuos en vez de muestras. Es decir, también se puede denominar validación aleatoria cruzada con n-pliegues basada en muestras (*subject-based random n-folded*) entendiéndose que los pliegues no son aleatorios en función de todas las muestras sino en individuos. La mayor parte de trabajos del estado del arte usan metodologías basadas en validación aleatoria cruzada. En este trabajo mostramos cómo nuestra metodología de validación es la más rigurosa para un problema de múltiples clases, por cuanto en metodologías basadas en validación aleatoria cruzada puede haber problemas de sobre entrenamiento inadvertido (cuando se usan varias muestras por individuo por expresión) o entrenamiento negativo (el sistema conoce muestras de otras expresiones de individuos pertenecientes al conjunto de validación). En cambio la metodología LSO elimina estos posibles problemas metodológicos: en el

conjunto de validación no hay ninguna muestra de individuos que tengan muestras en el conjunto de entrenamiento. Para ilustrar este fenómeno, hicimos pruebas de entrenamiento y validación tanto con nuestra metodología normal como con metodología aleatoria cruzada basada en muestras. En tanto que en el estado del arte no es usual encontrar trabajos con distintas metodologías de validación y el análisis de los resultados, consideramos que esto constituye un aporte acerca de la importancia de la selección y rigor en la metodología de validación.

En el ámbito de sistemas de clasificación, desarrollamos algoritmos novedosos basados en la confiabilidad de los metaclasificadores individuales. Normalmente la fusión de parámetros se realiza de acuerdo con la confiabilidad determinada de cada clasificador. En nuestro caso, incluimos la técnica original APCC (*A Posteriori Confidence Classification*), basada en la confiabilidad de cada metaclasificador a partir de su respuesta. La adición de esta técnica probó mejorar sustancialmente los resultaos de clasificación comparado con metodología normal de fusión de clasificadores. Así mismo, realizamos pruebas con máquinas deep learning que obtuvieron resultados promisorios. A diferencia del uso convencional de deep learning, basado directamente en pixeles de la imagen, en nuestro caso usamos las máquinas para obtener representación jerárquica a partir de las respuestas de los metaclasificadores débiles mediante autoencoders apilados y clasificación softmax. No fue posible determinar si efectivamente la adición de las máquinas deep learning produce mejores resultados, debido al margen de confianza de la clasificación, pero los valores obtenidos sugieren que efectivamente éste es el caso y es posible profundizar más en esta alternativa.

Adicionalmente, si bien en [98] se sugirió la necesidad de la realización de pruebas de reconocimiento humano de la expresión facial en la base de datos CK+ y pruebas de reconocimiento automático en la base de datos KDEF, hasta nuestro conocimiento esto no se ha hecho. En este trabajo hicimos un estudio de la capacidad de reconocimiento humano de la expresión facial usando voluntarios de la Universidad Federal de Rio Grande del Sur, Universidad Industrial de Santander e individuos externos, usando la base de datos CK+. Así mismo, hicimos pruebas de generalización del sistema de clasificación usando clasificadores entrenados con la base de datos CK+ en la base de datos KDEF. Los resultados de estas pruebas fueron interesantes. En primer lugar, la comparación que se hace en [98] no es completamente válida, por cuanto la base de datos KDEF tiene mayor dificultad de clasificación debido a su naturaleza no estandarizada. Es decir, los sistemas automáticos de clasificación tienen mejor desempeño que los humanos, pero no con una diferencia tan grande como la sugerida en el trabajo citado. Por otra parte, nuestras pruebas en este escenario fueron en condiciones de elevada dificultad, ya que el sistema de clasificación fue entrenado sin usar ninguna muestra de la base de datos KDEF. No obstante, los resultados de clasificación fueron mejores que los obtenidos por un panel de humanos mostrados en [99]. Es decir, se probó de manera contundente que nuestro sistema de reconocimiento tiene mejor capacidad de clasificación incluso en condiciones adversas: desconocimiento de la base de datos, muestras de ciertas expresiones no representativas de la expresión estándar (por ejemplo ira caricaturesca, que es de fácil clasificación por parte de humanos, pero que el sistema

tiene inconvenientes porque no fue entrenado con ninguna muestra similar) y contra mujeres jóvenes, que son el subconjunto de clasificadores humanos de mayor tasa de éxito en la clasificación de la expresión facial [71].

1.4. Contenido

Este trabajo está enfocado principalmente en el desarrollo de algoritmos eficientes que permitan la codificación de la información de los rostros preservando los parámetros necesarios para describir la expresión facial. Por lo tanto, la mayor parte de la investigación se refiere a la codificación de datos, reducción de dimensiones, extracción de parámetros y sistemas de clasificación. Sin embargo, el proceso completo es más extenso e incluye detección de rostro, procesamiento de imagenes, reducción de dimensiones, extracción de parámetros, selección de parámetros, sistemas de clasificación y validación, así como pruebas adicionales de generalización. Para cada una de estas etapas introdujimos nuevas soluciones y aportes que permiten mejorar el desempeño al reducir los costos de cálculo o aumentar las tasas de clasificación.

La presentación de esta disertación no corresponde necesariamente con el desarrollo cronológico de esta investigación. Por ejemplo, en tanto que la efectividad de la reducción de dimensiones o de la selección de parámetros es evaluada con los resultados de clasificación, estas etapas no fueron desarrolladas previamente a los sistemas de clasificación, sino de manera paralela y realimentada. En general, todas las etapas del trabajo tuvieron revisión y corrección en diversos momentos del desarrollo de la investigación. No obstante, para facilidad de lectura hacemos la presentación con el orden del protocolo de reconocimiento de la expresión facial etapa por etapa.

La descripción de las etapas de detección de rostro, procesamiento de imagen y corrección de iluminación está descrita en el capítulo 2. En la primera parte del capítulo hacemos una descripción general del estado de desarrollo teórico. El contenido restante del capítulo se refiere a nuestro aporte de detección de rostro por arquitectura dependiente de clasificación por nivel, que es una propuesta novedosa respecto de la arquitectura independiente redundante característica de la detección convencional AdaBoost. Así mismo, desarrollamos procesamiento de corrección de iluminación cuyo objetivo es obtener imágenes con mejores características de micro gestos y expresiones, de modo que la obtención de parámetros de expresión facial se facilite incluso si la calidad desde un punto de vista de visión humano no sea aparentemente adecuada.

En el capítulo 3 hacemos inicialmente una descripción de las vertientes más usadas en la obtención de parámetros para expresión facial, con énfasis en las técnicas basadas en patrones locales binarios, debido a su reciente desarrollo y utilidad como descripción de textura. En particular, hacemos referencia a los parámetros POEM (*Patterns of Oriented Edge Magnitudes*), que son una adaptación de los parámetros LBP convencionales. Con los parámetros POEM modificados para codificar textura hicimos pruebas preliminares que mostraron su alto desempeño en el reconocimiento de la expresión en imágenes estáticas. Posteriormente presentamos los algoritmos novedosos VPOEM y

TPOEM que permiten extraer información dinámica de textura de las secuencias de imágenes, debido a la utilidad de esta información en la descripción de expresiones. La codificación TPOEM fue probada usando sistemas simples de clasificación y los resultados mostraron gran capacidad de discriminación de la expresión facial, comparable con resultados del estado teórico incluso usando aún metodologías básicas de clasificación.

Uno de los principales inconvenientes de la codificación basada en patrones locales binarios es el tamaño de los datos. Este tipo de codificación no reduce de manera significativa el tamaño de los datos de entrada, que suelen ser de tamaño elevado debido al volumen binario de una imagen o secuencia de video incluso en baja resolución. Por otra parte, una reducción de dimensiones convencional, por ejemplo usando PCA, elimina información detallada que justamente es crucial en la descripción de la expresión, de modo que otras alternativas deben ser evaluadas. En el capítulo 4 presentamos el análisis del problema, sus principales dos alternativas (reducción supervisada y no supervisada) y la discusión acerca de la problemática del uso de reducción no supervisada. Así mismo, incluimos el algoritmo novedoso LC-NNMLE para el cálculo de la dimensión intrínseca, cuyos resultados de estimación de dimensiones para datos en racimo (*clustered*) de alta dimensión, similares al tipo de datos extraídos por los codificadores TPOEM, son superiores a los de todas las técnicas evaluadas del estado del arte. Finalmente, implementamos la reducción supervisada de dimensiones que permitió recortar el tamaño de los datos preservando la información requerida para la discriminación entre expresiones faciales.

En el capítulo 5 hacemos el desarrollo de la metodología usada para la selección de parámetros. Inicialmente mostramos la fundamentación teórica y estado de avance en esta temática. Posteriormente hacemos la descripción del algoritmo novedoso SFA-WM (*Sequential Feature Analysis in floating search evaluation and extraction of Weak Metaclassifiers*). Nuestro protocolo SFA-WM probó ser una alternativa eficiente, de bajo costo y de buenas prestaciones para seleccionar parámetros a partir de un lote de parámetros débiles. El algoritmo SFA-WM fue probado con otros algoritmos recientes del estado de desarrollo teórico y los resultados mostraron su alto desempeño en la selección de parámetros sin eliminar parámetros importantes, pero sin incurrir en los prohibitivos costos de cálculo de un protocolo de búsqueda completa.

El capítulo 6 está enfocado en el desarrollo de los sistemas de clasificación. Posteriormente a la presentación teórica del estado de avance en los sistemas de clasificación, presentamos nuestra propuesta de sistema de fusión de metaclasificadores por FDA y SVM incluyendo información mutua entre parámetros. Así mismo, en tanto que las metodologías deep learning han probado recientemente su utilidad en la descripción categórica de diversos fenómenos, añadimos una etapa de clasificación por deep learning adaptada, con resultados promisorios. Por último, hacemos una comparación entre nuestros resultados de clasificación y los resultados de algunos de los trabajos más prominentes y trabajos recientes de reconocimiento de la expresión facial, mostrando que en general nuestra tasa de clasificación es similar o superior a la de casi todos los trabajos reseñados. En tanto que algunos trabajos muestran tasas de reconocimiento cercanas al 100 %, mostramos además cómo el uso de ciertas metodologías de validación

puede conducir a estas tasas de acierto muy por encima a lo esperado, pero con error de prueba de hipótesis.

En el capítulo 7 mostramos algunas pruebas adicionales. En primer lugar, en tanto que hasta nuestro conocimiento no ha habido pruebas directas de comparación de desempeño entre clasificación humana y clasificación por máquina, hacemos validación de clasificación con la base de datos KDEF, que fue validada por un panel compuesto por numerosos individuos, y validación de clasificación humana de la base de datos CK+, mostrando que en ambos casos la tasa de clasificación del sistema automático es superior a la clasificación humana, incluso en condiciones desventajosas que se refieren detalladamente en el capítulo. Por otra parte, en tanto que el desarrollo inicial de este trabajo se hizo con la base de datos CK, antes de obtener la autorización de uso de la base de datos CK+, algunas muestras raramente tenían clasificacion exitosa. No obstante, muchas de estas muestras no eran, desde una perspectiva humana, adecuadamente representativas de la expresión etiquetada y, de hecho, fueron eliminadas en la versión extendida CK+. En el capítulo mostramos cómo estas muestras fueron clasificadas dinámicamente por nuestros sistemas de reconocimiento y la diferencia de los vectores de puntaje entre este tipo de muestras y muestras más adecuadas para la expresión. Esto permitió concluir que por una parte probablemente el sistema de clasificación no estaba cometiendo errores, sino eran errores de las etiquetas en la base de datos CK, y por otra parte algunos trabajos del estado teórico que muestran tasas superiores al 95 % de clasificación con esta base de datos pueden tener errores metodológicos sea por el uso de metodología aleatoria cruzada basada en muestras, con sobreaprendizaje o aprendizaje negativo, o por descarte manual de muestras de difícil clasificación.

2. Detección de rostro y preprocesamiento de imágenes usando cascadas Haar mejoradas y árboles Bayesianos para detección *Boosted*

2.1. Introducción

Desde un punto de vista numérico, las variaciones de iluminación representan un importante inconveniente en el problema de itentificación facial o reconocimiento de la expresión facial. Idealmente se espera que las variaciones interclase (variaciones entre individuos o variaciones entre individuos ejecutando distintas expresiones faciales) sean menores que las variaciones intraclase para una métrica dada. Debido a las variaciones de iluminación esto no es necesariamente el caso. Esto se puede observar en la figura (2) con algunas imágenes de la base de datos Yale [52].

Las dos filas contienen imágenes de dos individuos diferentes. Sin embargo, si usamos una métrica simple tal como la comparación directa entre los pixeles, tal como se muestra en la ecuación (2.1) , es posible encontrar imágenes más cercanas entre dos grupos (clases) distintos, lo cual puede ser problemático para los algoritmos de clasificación.

$$d_{ij} = \sum_{x,y} \|I_i(x,y) - I_j(x,y)\| \tag{2.1}$$

Un algoritmo de clasificación no hace una separación entre clases simplemente como una comparación directa de los valores de los pixeles entre imágenes de distintas clases, pero el principal punto es que debido a los inconvenientes de iluminación dos imágenes pueden aparecer más cercanas entre sí para un humano o para un algoritmo de clasificación. Debido a esto, una etapa intermedia de procesamiento y corrección de iluminación es deseada, con el fin de obtener mejor desempeño en las etapas siguientes del sistema de clasificación.

Figura 2. Rostros de la base de datos Yale con diferente iluminación

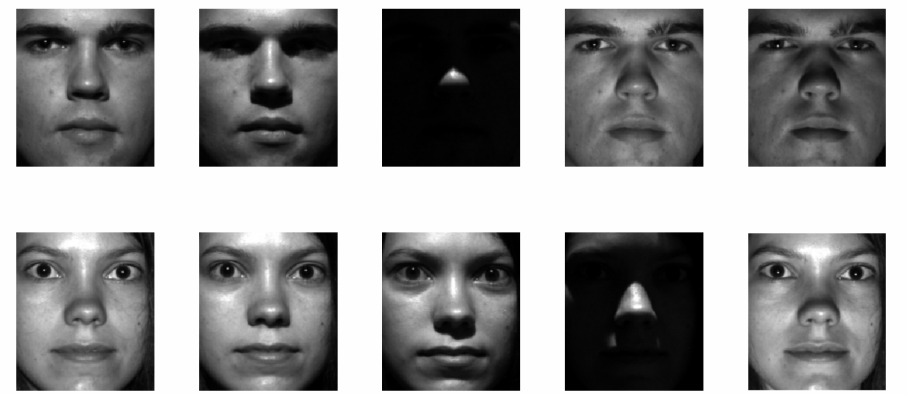
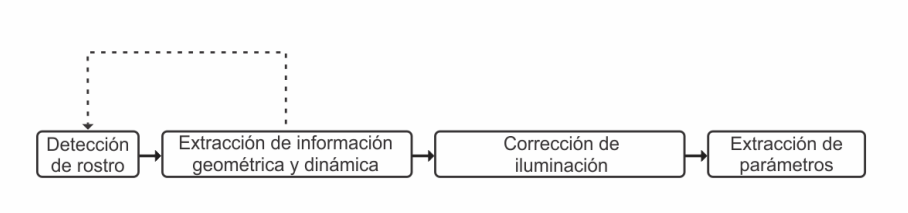


Figura 3. Sistema global de reconocimiento de expresión facial



El procesamiento y la corrección de iluminación pueden ser hechos en varias etapas del sistema global. Generalmente es deseable hacerlo antes de la detección de rostro, por cuanto la corrección de iluminación puede ayudar a detectar el rostro más fácilmente. En nuestro caso las condiciones relativamente controladas de imágenes y video hacen que la detección de rostro sea simple. Por tanto, se reducen costos de memoria y de ejecución al hacer la corrección de iluminación después de detectar el rostro. Por otra parte, realizar el procesamiento completo de corrección de iluminación sobre la imagen completa de 640×480 pixeles es mucho más costoso que sobre el rostro detectado y escalado a 128×128. La arquitectura del sistema de detección de rostro y corrección de imagen es mostrada en la figura 3.

En este capítulo haremos una revisión de la etapa de detección de rostro y corrección de iluminación, que corresponden al procesamiento efectuado en las imágenes/videos antes de que los parámetros faciales sean obtenidos.

En cuanto a la detección facial, el objetivo es definir con precisión las regiones en las cuales un rostro individual está presente en la imagen o en el cuadro de video, en oposición a detección de parámetros faciales, que se refiere a la extracción de características individuales en un rostro y generalmente se basan en la presunción de que hay un solo rostro de interés en la imagen [31] Éste es un problema no-lineal con diversos retos [182] [112] [187]. Los rostros tienen un alto grado de variación entre individuos debido a variaciones morfológicas, raza, género, edad y características particulares. Adicionalmente, hay alto grado de variación incluso entre imágenes del mismo individuo debido a pose, uso de accesorios y expresiones faciales. Por otra parte, condiciones fotográficas tales como resolución, exposición e iluminación agregan variabilidad al conjunto de posibles rostros en una imagen. Como tal, el problema tiene alta complejidad y se debe lidiar con diversos retos. Además, por cuanto este trabajo está dirigido hacia la obtención de un sistema de reconocimiento de expresión facial que sea aplicable para tiempo real, hay otra limitación importante relacionada con los costos de cálculo del algoritmo.

2.2. Fundamentos teóricos y estado del arte

Debido a la naturaleza del problema, la detección facial ha sido abordada desde diversas perspectivas, de manera que el desarrollo ha tenido contribuciones basadas en distintas técnicas. En esta sección haremos una breve revision del desarrollo histórico de algunos algoritmos de detección de rostro, fundamentación teórica, ventajas y

desventajas.

2.2.1. Detección facial usando PCA

La idea de la detección facial usando análisis de componentes principales (PCA) es que dado un conjunto de imágenes de entrenamiento X, una transformación PCA puede ser efectuada y los parámetros obtenidos son llamados eigenfaces [161]. Las *eigenfaces* representan parámetros de las imágenes originales embebidos en el espacio PCA. Debido a la naturaleza lineal de PCA, cualquier imagen del lote de entrenamiento puede ser reconstruida sin pérdida por la suma de sus *eigenfaces*, y cada *eigenface* contribuye en cierto grado a la reconstrucción de la imagen original. Por otra parte, algunas *eigenfaces* tienen mucha menor contribución que otras, entonces pueden ser eliminadas del conjunto. La detección consiste en el cálculo de los pesos de la región por explorar. Si los pesos son similares a los del conjunto de rostros de entrenamiento, se puede establecer similitud, entonces la región explorada es etiquetada como rostro; en otro caso, la región se etiqueta como no rostro. Para clasificar un nuevo objeto x, sus pesos son obtenidos según la ecuación (2.2)

$$w_k = u_k^T(x - \Psi), \;\; k = 1, ..., M \tag{2.2}$$

u son las *eigenfaces* obtenidas según el procedimiento previamente señalado, Ψ es el promedio de las imágenes de entrenamiento, M es el número de *eigenfaces*. Los pesos obtenidos son comparados con los pesos del conjunto de entrenamiento y si la métrica usada es menor que un umbral determinado, el nuevo objeto es determinado como no facial. El reconocimiento de rostros usando PCA tiene algunos inconvenientes [76]. PCA tiene alta dependencia de los niveles de los pixeles. Debido a ello, imágenes ruidosas o con problemas de iluminación tienen baja tasa de reconocimiento. Adicionalmente, para obtener un conjunto de *eigenfaces* adecuado se requiere de un significativo número de imágenes en el conjunto de entrenamiento. Esto conduce a un alto costo de cómputo. Por otra parte, al truncar el número de *eigenfaces* hay pérdida de información detallada y específica que puede conducir a detección inadecuada. Debido a esto, hay técnicas más poderosas para detección de rostro, tales como las basadas en LDA/FDA (discriminante lineal/discriminante Fisher) que usan clasificadores débiles con aprendizaje boosted learning. Otro inconveniente es que PCA no garantiza que los eigenvectores y *eigenfaces* sean descriptores adecuados de la expresión facial, y esto es probablemente evidenciado por la pobre tasa de reconocimiento. Esto es particularmente crítico en el estudio de expresiones faciales, pues las expresiones faciales típicamente se representan por detalles específicos del rostro, como micro expresiones, que posiblemente son descartados por su baja energía en la determinación de las *eigenfaces*, de manera que la detección de rostro por técnicas basadas en PCA fue descartada luego de estas pruebas preliminares. Sin embargo, PCA y avances derivados han sido usados en un amplio número de trabajos, tales como PCA simétrico [184], PCA 2-dimensional y 2-dimensional simétrico [188] [111], PCA 2D para reconocimiento de rostro [141], PCA diagonal [189], PCA multidimensional [113] y PCA+aprendizaje *boosted* híbrido [191].

2.2.2. Discriminante Fisher Lineal y SVM

El discriminante Fisher lineal (FLD) [50] tiene una notable ventaja sobre las *eigenfaces*. Al desarrollar FLD, las proyecciones tienen pequeña variación intraclase al tiempo que maximizan la variación interclase. Como consecuencia, FLD proporciona mejor separabilidad entre las regiones de rostro y no rostro. Como tal, la transformación dada por $y = W^T x$, donde W es la transformación Fisher y x es la nueva muestra a ser transformada, minimiza el radio entre la variación interclase y la variación intraclase dadas por las ecuaciones (2.3)(2.4).

$$S_B = \sum_{i=1}^{c} N_i(\mu_i - \mu)(\mu_i - \mu)^T \tag{2.3}$$

$$S_W = \sum_{i=1}^{c} \sum_{x_k \in X_i} (x_k - \mu_i)(x_k - \mu_i)^T \tag{2.4}$$

S_B es la matriz dispersa interclase, S_W es la matriz dispersa intraclase, μ_i es el vector medio de la clase i y N_i es el número de muestras por clase i. La proyección que maximiza la relación entre varianzas está dada por (2.5).

$$W_{opt} = \arg \max_W \frac{|W^T S_B W|}{|W^T S_W W|} \tag{2.5}$$

Ésta es la aproximación más simple de FLD en detección facial, pero hay una mejora notable llamada *Fisherfaces* [8] que resuelve el problema de matrices singulares S_W al previamente proyectar las imágenes en una dimensión menor usando PCA. Recientemente, técnicas basadas en Fisher han sido usadas con distintos parámetros, tal como en [152], que usa Fisher para separar clases basadas en parámetros Gabor de dimensión reducida con *AdaBoost*, pero siguen siendo técnicas poco usuales en detección de rostro.

La detección basada en máquinas de soporte vectorial SVM usa las ventajas de SVM en la maximización geométrica de la distancia interclases. Un hiperplano es construido para separar las regiones faciales y no faciales [121]. Sin embargo, tal como sucede con FLD, su aplicación en detección de rostro ha sido limitada hasta ahora.

2.2.3. Detección de rostro basada en redes neuronales

Las redes neuronales han sido usadas para construir clasificadores que separen regiones entre rostro y no rostro. El principal reto son las regiones de no rostro, por cuanto caracterizar estas regiones corresponde a todo el universo de posibles regiones de no rostro, que es un conjunto muy elevado de datos, teniendo en cuenta que en el universo de imágenes posibles, las regiones faciales son una fracción insignificante en comparación con todas las demás escenas potenciales. En [138] el problema se abordó añadiendo progresivamente imágenes al conjunto de entrenamiento durante el entrenamiento. Los datos de entrada son parches de 20×20 que corresponden a datos de rostro

y no rostro, incluyendo conjuntos de rostros del mismo tamaño por transformación de rostros (escala, translación, rotación y reflejo).

En general, las redes neuronales tienen inconvenientes serios en la clasificación de rostros, debido al gran tamaño del problema a caracterizar: están sujetas a *overfitting* si la red neuronal es de un tamaño considerable, pero pueden fácilmente ser pobremente entrenadas si la red es limitada. Más aún, en este tipo de problemas las redes neuronales tienen inconvenientes de optimización [181] [180]. Otra limitación de las redes neuronales en la detección de rostro, sobre todo si la arquitectura es piramidal, es el elevado número de cálculos requerido. El primer problema es posible que sea solucionado con los recientes avances en *deep learning* [11], pero el problema del costo computacional sigue presente, de manera que son pocos los trabajos relativamente recientes que usan redes neuronales para detección de rostro [32].

2.2.4. Detección basada en algoritmos *boosted*

Viola y Jones desarrollaron una técnica de detección de rostros basada en parámetros débiles [167], que derivó en las técnicas más exitosas a la fecha. La importancia de AdaBoost es tal que muchos de los más recientes desarrollos son basados en su idea general. Los algoritmos boosted learning se basan en la premisa de que existen parámetros que permiten medir propiedades específicas en una localización, escala y radio de aspecto de una imagen. Estos parámetros son obtenidos mediante filtros rectangulares tipo Haar que obtienen características geométricas y de textura. Debido a la simplicidad de los filtros, su capacidad de discriminación entre rostro y no rostro es pequeña. En consecuencia, el algoritmo seleccina en cada iteración un clasificador débil cuya tasa de error sea menor que 0,5 calculando los pesos y los umbrales que minimicen la función de error.

Al realizar este proceso iterativamente, se puede construir un clasificador final fuerte compuesto por la sumatoria del filtrado de todos los clasificadores débiles obtenidos. Las principales ventajas de esta metodología de detección de rostro son los bajos costos de memoria y de procesamiento, la baja tasa de error y la facilidad de modificación de funciones de castigo que permiten reducir errores (falsos positivos o falsos negativos) a cambio de un compromiso de aumentar el error complementario. El clasificador final es dado por la ecuación 2.6.

$$F(x) = sign(\sum_{t=1}^{T} f_t(x)) \tag{2.6}$$

donde x es la señal de entrada, $f_t(x)$ son los filtros obtenidos y T es el número total de filtros.

2.3. Detección de rostro propuesta

Los algoritmos basados en boosted learning tienen generalmente bajas ratas de falsos negativos, normalmente a costa de un incremento de falsas detecciones. Esto implica que dado un sistema propiamente entrenado es improbable que una subventana facial sea eliminada por alguna etapa del algoritmo, mientras que hay una alta certeza de que una subventana eliminada corresponde a una region no facial.

Esta tasa baja de falsos negativos sugiere que es posible desarrollar un algoritmo de detección de rostro en una secuencia de video con menor costo de ejecución. La mayor parte de cuadros en una secuencia de video tienen alta semejanza con sus vecinos cercanos. Como tal, las localizaciones faciales en un cuadro brindan información importante acerca de la probabilidd de localizaciones faciales en los cuadros cercanos, salvo dichos cambios de escenas.

2.3.1. Detección facial multi resolución

Con el fin de validar nuestra hipótesis, tomamos varias secuencias de video con dos o más escenas distintas y regiones faciales y no faciales. AdaBoost convencional fue usado para la detección de rostro en cada cuadro de video y las subventanas fueron almacenadas. Tal como se esperaba, debido a patrones específicos en el fondo de la imagen los falsos positivos fueron numerosos, pero la mayor parte de las regiones faciales fueron adecuadamente seleccionadas por el detector de rostro. Con el fin de tener un criterio de comparación confiable, las localizaciones reales de los rostros fueron manualmente seleccionadas en los cuadros de video usados. Adicionalmente, secuencias de video de las bases de datos CK y CK+ fueron incluidas.

Nuestra primera aproximación fue basada en la probable baja transición de las regiones del fondo entre cuadros de video. En esta prueba se esperaba que una región descartada por una etapa completa de detección de rostro permanecería eliminada en la misma localización o al menos muy cercana en el siguiente cuadro. Debido a que las regiones rechazadas son no faciales con alta certeza gracias a la baja tasa de falsos negativos del algoritmo AdaBoost, consideramos que un menor costo de procesamiento puede ser obtenido al considerar que la probabilidad de aparición de un nuevo rostro en estas regiones es limitada.

Dado un subconjunto $f_{i,k}$ con la localización de las K sub ventanas no eliminadas en el cuadro i, el siguiente paso es reducir la resolución de las ventanas eliminadas. La idea es ejecutar un algoritmo de detección facial igualmente riguroso en las regiones no eliminadas y al mismo tiempo reducir los costos de cálculo en las regiones descartadas en los cuadros precedentes. El protocolo consiste en transformar la imagen en el cuadro $i+1$ en una imagen de menor resolución I'_{i+1}, desarrollar detección facial en esta versión de menor resolución y sobre las regiones $f_{i,k}$ de la imagen completa I_{i+1}. En cuanto el tamaño de las regiones no rechazadas generalmente es una pequeña fracción de la imagen total, los costos de cálculo son reducidos y la precisión se mantiene elevada, salvo en casos en los que la pérdida de resolución ocasione detección fallida de un rostro

previamente detectado. Posteriormente veremos cómo este inconveniente es atenuado.

El entrenamiento de los nodos de clasificación fue hecho usando la arquitectura convencional de AdaBoost. Actualmente hay técnicas eficientes para entrenar los clasificadores débiles, tales como aproximaciones probabilísticas y clasificación basada en SVM, pero decidimos no incluirlas en este trabajo debido a varias consideraciones. Primero, el desarrollo de técnicas de punta de detección facial es una investigación completa en sí misma, mientras que el principal foco de este trabajo es el reconocimiento de la expresión facial, que requiere de otras tareas importantes de extracción de parámetros, procesamiento de datos, clasificación y validación. Segundo, este trabajo está basado en la base de datos CK+ y otras bases de datos de imágenes frontales, así como secuencias de video de rostros individuales capturados por el autor, de manera que la calidad de las imágenes y cuadros de video hace que la detección de rostro sea simplificada. Por último, buena parte de estos desarrollos recientes fue publicada en el transcurso de esta tesis doctoral, cuando otras etapas más avanzadas del trabajo estaban en implementación.

La principal desventaja de esta aproximación multi resolución es que puede ignorar la aparición súbita de rostros en regiones que no incluían rostros en el cuadro inicial, la detección inicial de rostro no es válida para cuadros lejanos en el dominio temporal y hay cambios de escena en el video. Para resolver o atenuar los inconvenientes, la detección completa de rostro se ejecuta nuevamente cuando se cumple una de dos condiciones: un número predefinido de cuadros ha transcurrido desde la última búsqueda completa o cuando la búsqueda de rostros en regiones no eliminadas tiene resultados considerablemente distintos comparados con los de cuadros precedentes (cambio significativo de posición de los rostros o desaparición de rostros previamente detectados). Se usaron como parámetros iniciales 10 cuadros de video entre búsquedas exhaustivas y al menos 5 % de cambio de posición normalizada entre rostro y rostro detectado para realizar unan nueva búsqueda.

En promedio el costo de cálculo se redujo 23 %, con mayor reducción aumentando el umbral de tiempo entre búsquedas completas, a cambio de ligeramente peor desempeño (muy inferior al error estadístico con confianza 95 %, de manera que lo consideramos irrelevante). Posteriormente se mejoraron notablemente los resultados, sobre todo en reducción de falsos positivos, al incluir información de cuadros vecinos en la clasificación de una región como rostro o no rostro. Dado $F = \{f_{i,k}\}$, correspondiente a las regiones candidatas detectadas, $D = \{d_{i,k}\}$ son las regiones finales, donde i es el número de imagen en la secuencia de video y k es el índice de cada región. Para determinar si una región $f_{i,k}$ es incluida como región final se usa el procedimiento descrito en el pseudoalgoritmo 1.

donde T es el número de cuadros, α_n son las ponderaciones de los cuadros temporales vecinos asignadas empíricamente usando los vecinos $i-1$ y $i+1$ para reducir costos de cálculo, K_i es el número de regiones detectadas en el cuadro i y h es un umbral estimado empíricamente en 0.8. Adicionalmente, se implementó un detector simple de cambio de escena por comparación directa entre cuadro y cuadro y cuando hay cambio de escena se reinicia F con una búsqueda completa, al igual que con los criterios previamente utilizados. Esta modificación incrementa ligeramente el costo de cálculo debido a

Algorithm 1 Detección ponderada de rostros

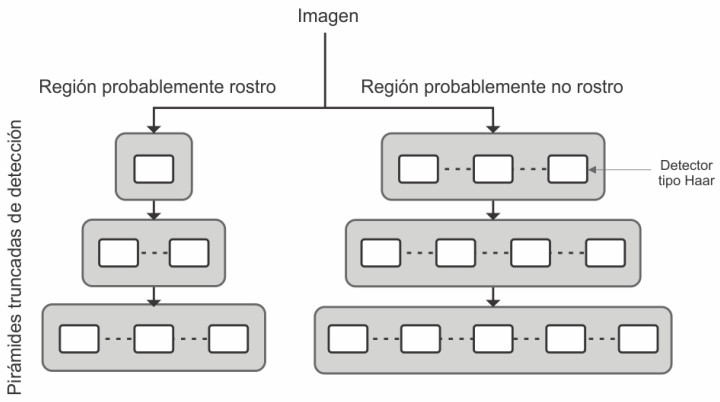
procedure BÚSQUEDA DE REGIONES FACIALES
  $D = \{d_{i,k}\}$
  for $i \leftarrow 1, T$ **do**
    $t \leftarrow t + 1$
    Calcular $F = \{f_{i,k}\}$
    for $k \leftarrow 1, K_i$ **do**
      $s_{i,k} = \alpha_1 f_{i,k} + \alpha_2 f_{i-1,k} + \alpha_3 f_{i-2,k} + ...$
      if $s_{i,k} > h$ **then**
        $D \leftarrow f_{i,k}$
    if *Cambio de escena detectado* **then**
      Reset, F
    if $t > nt$ **then**
      Reset, F, $t \leftarrow 0$
    if $d(f_{i,k}, f_{i-1,k}) > dt$ **then**
      Reset, F

la inclusión de los cuadros adyacentes. No obstante, el procedimiento de costo computacional más elevado es aún la detección de rostros, de manera que el incremento no es sustancial. Con esto se consiguió mejorar los resultados precedentes, especialmente en el número de falsos positivos, que se reduce debido a la baja probabilidad de aparición de artefactos similares a rostros en cuadros consecutivos.

2.3.2. Detección *Boosted* sin estructura de cascada lineal

Las cascadas de Algoritmo de aprendizaje de máquina Adaptive Boosting, basado en fusión en cascada de clasificadores débiles (AdaBoost) convencionales son de naturaleza lineal con complejidad creciente con el fin de eliminar sub ventanas candidatas con mayor precisión en cada etapa de clasificación. De esta forma el primer clasificador es simple, normalmente con pocos clasificadores que rechazan la mayor parte de las regiones no faciales, y los clasificadores siguientes tienen mayor complejidad, pero no deben hacer clasificación sobre la imagen completa. Esto implica, sin embargo, redundancia en la selección de regiones faciales, por cuanto una región facial debe pasar por todos los clasificadores. En una secuencia de video una región facial $f_{i,k}$, donde i es la variable temporal y k es el índice de cada región, probablemente se mantiene estable en cuadros cercanos y esta información puede ser útil en la reducción de estas clasificaciones redundantes. Nuestra propuesta es entonces usar dos distintos lotes de clasificadores de acuerdo con la etiqueta de la clasificación AdaBoost en cuadros previos cercanos. Las regiones rechazadas previamente pasan por los primeros nodos de clasificación, mientras que las regiones detectadas pasan únicamente por los últimos nodos de clasificación. Dada la naturaleza independiente de los nodos de clasificación AdaBoost convencional, una región facial que no sea rechazada por los últimos nodos

Figura 4. Etapas truncadas de filtrado de regiones faciales y no faciales



de clasificación no habría sido rechazada por los nodos iniciales, de manera que se evita redundancia. Esta característica de AdaBoost es conveniente cuando no se dispone de información previa, por cuanto progresivamente se eliminan regiones y sólo una región pequeña de la imagen atraviesa los nodos más complejos, pero en nuestro caso disponemos de información *a priori* sobre la probabilidad de pertenencia a rostro de ciertas regiones. La arquitectura es mostrada en la figura 4. Como se observa, esta arquitectura hace que las regiones no faciales sólo deban atravesar etapas cortas de clasificación, mientras que las regiones probablemente faciales no tengan que ser sujetas a todos los clasificadores iniciales simples, que añaden costos de cálculo.

Con el algoritmo completo usando pirámide de detección truncada se consiguió una ligera disminución del costo de cálculo de en promedio 3 % en la detección de rostros en secuencias de video naturales en las cuales la cantidad de regiones faciales es normalmente pequeña en comparación con el tamaño completo de la imagen, pero se logró una reducción de alrededor de 22 % de costo de cálculo en secuencias de video de expresión facial, en las cuales las regiones detectadas normalmente son una fracción importante de la imagen. Adicionalmente, en estas bases de datos es posible implementar detección de rostro menos exigente (es decir, usar nodos con menor número de parámetros (*features*) en las etapas finales) debido a la naturaleza controlada y simple de estas secuencias, pero decidimos no implementar esta adaptación por cuanto probablemente representa deterioro en la capacidad de detección de rostros en secuencias de video naturales.

En la tabla 1 mostramos los resultados obtenidos en la detección de rostros en secuencias de video usando la base de datos CK [98] y secuencias de video de rostros frontales recopiladas por el autor. El tiempo de cálculo es reducido con las técnicas de multi resolución, cuadros ponderados y árboles no lineales de detección truncados. El compromiso es, por otra parte, que aumenta el número de falsas detecciones, aunque en nuestra aplicación específica esto no es un inconveniente, puesto que casi todas las falsas detecciones son eliminadas automáticamente. En el equipo de cómputo usado, con procesador i7 de 2.4 GHz y 4GB de memoria RAM, con algoritmos implementados

en Python, estos resultados equivalen a 11.7ms promedio por imagen. El protocolo TPOEM, descrito en el capítulo 3, realiza diezmado cada 5 ó 7 cuadros de video, dependiendo de los parámetros usados. Esto equivale a 167ms-233ms, de manera que usar 11.7ms promedio en la detección de rostro por cuadro no constituye un porcentaje sustancial del tiempo disponible para ejecución en tiempo real.

Tabla 1. Comparación de detección de rostro por AdaBoost y por nuestras propuestas

AdaBoost convencional	**Rostros totales**	**Rostros detectados**
Detección de rostros	1874	1873
Falsos positivos	27	
Tiempo de procesamiento	28.335s	
Multi resolución	**Rostros totales**	**Rostros detectados**
Detección de rostros	1874	1858
Falsos positivos	20	
Tiempo de procesamiento	25.620s	
Cuadros ponderados	**Rostros totales**	**Rostros detectados**
Detección de rostros	1874	1863
Falsos positivos	21	
Tiempo de procesamiento	23.317s	
Árboles de detección truncados	**Rostros totales**	**Rostros detectados**
Detección de rostros	1874	1872
Falsos positivos	36	
Tiempo de procesamiento	21.875s	

Los resultados entre AdaBoost convencional y árboles de detección truncado muestran que las dos aproximaciones son adecuadas para la detección de rostro, con mayor a 99.9 % capacidad de detección. En nuestro caso la tasa de falsos positivos, no obstante, fue más elevada. Sin embargo, esto no es un inconveniente importante, debido a que los falsos positivos ocurrieron en su totalidad en zonas pequeñas y por lo general alejadas del centro de la imagen. En tanto que las condiciones de las secuencias de video son conocidas previamente, estos falsos positivos son eliminados automáticamente debido a su posición y tamaño. Adicionalmente, el costo de cálculo usando nuestra propuesta fue considerablemente inferior, 22.8 % en esta prueba específica.

2.4. Corrección de iluminación

En este trabajo hay dos retos importantes: corregir los problemas de iluminación y usar algoritmos de bajo costo aplicables al problema del tiempo real, tal como lo definimos en el capítulo 1. Estos dos retos se comprometen mutuamente, por cuanto

no es posible usar corrección de iluminación sofisticada, por ejemplo con modelamiento de luz y objetos, manteniendo costos de cálculo compatibles con tiempo real. Debido a esto, se intentó desarrollar corrección de iluminación lo más adecuada posible pero manteniendo tiempos de ejecución discretos.

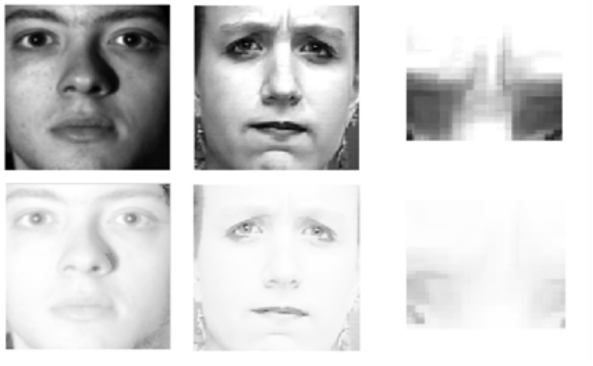
Una de las ventajas del uso de los descriptores basados en patrones locales binarios es que su capacidad de descripción de textura es fuerte, inlcuso con considerables problemas de iluminación y ruido en la imagen. Debido a ello, es suficiente que la corrección de iluminación produzca imágenes no necesariamente perfectas, pero suficientes para poder extraer parámetros descriptores de la expresión facial. Sin embargo, es imposible determinar *a priori* qué tan exigente debe ser la corrección de iluminación, de modo que si bien esta sección corresponde al orden lógico de ejecución del sistema completo, buena parte del desarrollo presentado a continuación fue realizado posteriormente a la implementación del resto de etapas del sistema. Como tal, no se tuvo en cuenta la restricción de costo de cálculo en los primeros algoritmos de corrección de iluminación implementados y posteriormente se realizaron simplificaciones en la medida en que la clasificación no fuese perjudicada. Por otra parte, pruebas preliminares con sistemas de clasificación relativamente simples mostraron adecuado desempeño incluso sin realizar corrección de iluminación previa. Es decir, los algoritmos de detección de rostro estuvieron en capacidad de realizar detección facial sobre las imágenes y secuencias originales, de modo que la ejecución de la corrección de iluminación probó ser más importante para acentuar los gestos faciales, requeridos para la descripción de la expresión facial. De esta forma se obtuvieron resultados satisfactorios sin requerimiento computacional elevado.

Una consideración importante en el desarrollo de corrección de iluminación para extracción de parámetros es que las imágenes requeridas no necesariamente corresponden a imágenes de buena calidad para un observador humano y también dependen del tipo de clasificación. Por ejemplo, para un observador humano una imagen con iluminación corregida adecuada es generalmente una imagen suave, con transiciones ligeras. Así mismo, para un sistema de clasificación facial este tipo de imágenes puede ser adecuado. En el caso del reconocimiento de la expresión facial, sin embargo, este tipo de corrección de iluminación puede eliminar detalles finos que son descriptores de expresiones específicas, de modo que es un procesamiento indeseable.

2.4.1. Retos de la corrección de iluminación

Para un número considerable de imágenes es relativamente simple diseñar un algoritmo que produzca una imagen corregida que sea mejor desde el punto de vista de un observador humano. Sin embargo, esto no significa que la imagen sea más apropiada para la extracción de parámetros que una imagen no procesada. Esto ocurre porque la mayoría de algoritmos de corrección de iluminación producen pérdida de información y es posible que parte de la información perdida fuese importante para la descripción del problema. Adicionalmente, buena parte de los algoritmos de corrección de iluminación incluyen el uso de filtros pasa bajo que pueden mejorar la calidad percibida de la ima-

Figura 5. Corrección de iluminación por mapeo exponencial del histograma



gen, pero eliminan información detallada que puede ser importante. Mostraremos esto usando dos imágenes de las bases de datos Yale y Cohn-Kanade que fueron procesadas usando mapeo exponencial del histograma con el fin de mejorar la normalización de iluminación. La salida se observa en la figura (5).

En la fila superior se muestran las imágenes no procesadas más detalles del ceño de la segunda imagen. En la fila inferior las imágenes procesadas. Se puede observar que este procedimiento mejora notablemente las regiones con sombras y la apariencia general de las imágenes desde un punto de vista humano (y probablemente para un sistema de identificación facial). Sin embargo, los detalles del ceño que son bastante claros en la imagen superior se pierden casi por completo en la imagen inferior. Esta pérdida producida por el mapeo del histograma es importante, por cuanto la imagen superior del medio, etiquetada como enojo en la base de datos, fácilmente puede ser etiquetada como tristeza o neutral por un observador humano o un sistema de clasificación en la imagen inferior del medio. Esto ilustra cómo algunos algoritmos convencionales de corrección de iluminación pueden empeorar el desempeño del sistema.

Por otra parte, hay alternativas interesantes para mejorar las condiciones de iluminación sin perder información de patrones detallados en la imagen. Una idea común es el uso de imágenes de cociente (*quotient images*). La idea general del SQ (*self-quotient*) [172] es obtener el radio entre las imágenes y las imágenes suavizadas usando kernels.

$$Q = \frac{I}{I * H} = \frac{I}{\hat{I}} \tag{2.7}$$

$$H = WG \tag{2.8}$$

En la ecuación (2.7) Q es la imagen de auto cociente, I es la imagen original. En la ecuación 2.8 H es un filtro de suavizado obtenido con la matriz de pesos W y un filtro G. Los filtros pueden ser obtenidos usando aproximaciones anisotrópicas relacionadas con el modelo Lambertiano de iluminación, pero para aplicaciones prácticas kernels Gaussianos son adecuados debido a los costos de ejecución. Los resultados de SQ dependen del tamaño del kernel. Si el kernel es muy pequeño, la imagen suavizada es muy parecida a la imagen original, pero un kernel grande produce halos y la información

Figura 6. Corrección de iluminación por MSQ

de iluminación es muy global. Una alternativa es el uso de distintos kernels por MSQ (*multi-scale self quotient*) [173].

$$Q = \frac{I}{\frac{1}{N} W_k H_k * I} \tag{2.9}$$

En la ecuación (2.9) N es un parámetro de normalización, W_k son los pesos por filtro y H son los filtros, de modo que la imagen final MSQ es una combinación de la imagen original con el conjunto de filtros con distintos tamaños y pesos. En la figura (6) se muestra la misma imagen de la base de datos Cohn-Kanade procesada usando MSQ.

MSQ produce imágenes cuya salida puede ser percibida como de peor calidadd para un observador humano. No obstante, nuestras pruebas experimentales y los resultados en [89] muestran que el resultado es más apropiado para extraer parámetros válidos. Se observa que la imagen es menos natural que la imagen original, pero se acentúan los rasgos característicos de la expresión facial. Sin embargo, ocurre otro inconveniente. MSQ requiere de un elevado costo de cálculo debido al uso de múltiples imágenes *self-quotient* con distintos tamaños de *kernel*. En consecuencia, no es una técnica apropiada para tiempo real, al menos no en el equipo usado para el desarrollo de este trabajo, por cuanto MSQ usó en promedio más de 1.5s por imagen para hacer la corrección de iluminación. Entonces el reto fundamental es obtener imágenes adecuadas pero sin incurrir en esos costos de cálculo elevados.

2.4.2. Manipulación de imagen

La primera etapa del procesamiento de imagen fue la manipulación del histograma de la imagen. El objetivo de esta etapa es mejorar la forma del histograma para producir una imagen cuyo rango dinámico global y local sea más apropiado para la extracción de parámetros. Normalmente cuando el requerimiento es la obtención de una imagen visualmente atractiva se puede despreciar hasta cierto grado la información local en cambio de una imagen visualmente más atractiva, pero en nuestro trabajo esto no es deseable. No se requiere de una calidad visual de la imagen, sino que se preserve y acentúe la información relacionada con la expresión facial. En consecuencia, buena parte de las técnicas de manipulación de histograma no son aceptables para extracción de parámetros. En este trabajo los mejores resultados fueron obtenidos cuando el

Figura 7. Corrección de iluminación por ecualización de histograma e histograma adaptativo limitado por contraste

Figura 8. Corrección de iluminación por implementación simplificada de histograma adaptativo

histograma es ecualizado y posteriormente se realiza una trasformación adaptativa de contraste limitado. Un ejemplo es la figura (7).

La imagen en la izquierda es la imagen sin procesar. La segunda imagen es después de ejecutar ecualización de histograma. La tercera imagen es obtenida luego de transformación adaptativa truncada de histograma [129]. La cuarta imagen es obtenida al realizar consecutivamente los ecualización de histograma y transformación de histograma adaptativo limitado por contraste (CLAHE)[199]. Si bien el resultado final muestra los gestos acentuados, CLAHE tiene el inconveniente de que se debe dividir la imagen en una grilla de pequeñas celdas, ejecutar histograma adaptativo sobre cada celda, truncado iterativo y combinar los resultados por interpolación bilineal de cada celda con las celdas vecinas, y este procesamiento tiene un elevado costo de máquina. Una alternativa usada para nuestro trabajo fue aplicar transformación adaptativa de histograma y reducir el número de iteraciones. La imagen final fue filtrada para truncar las variaciones locales exageradas en el contraste.

En la figura (8) se muestra el resultado al realizar este procedimiento simplificado.

La imagen obtenida es muy similar a la obtenida usando el procedimiento precedente. Hay ligeras diferencias en las variaciones locales de intensidad cuando los valores de los pixeles cambian sobre un umbral, pero a cambio el costo del procesamiento total para corrección de iluminación fue reducido en promedio 76 %, con costo completo de 14ms. En términos visuales la imagen puede parecer de menor calidad que la obtenida con otras manipulaciones de histograma, pero el acentuamiento de los contrastes locales la hace más apropiada para procedimientos basdos en patrones y parámetros locales.

Resultados similares en tiempo de máquina fueron conseguidos al usar bancos de fil-

Figura 9. Corrección de iluminación por ecualización de histograma y técnica isotrópica de difusión

tros emuladores de la visión animal por cascada de filtros no lineales, tal como se sugiere en [168] y el desempeño de estos filtros retinales en condiciones pobres de iluminación es mejor. Sin embargo, nuestra propuesta fue más satisfactoria en condiciones normales y buenas de iluminación, por cuanto, a diferencia de los filtros retinales, mantiene más adecuadamente las variaciones locales en la imagen.

A modo de prueba usamos algoritmos más avanzados despreciando el relacionado costo de ejecución de estos algoritmos. Los mejores resultados en términos de extracción de parámetros para expresión facial fueron obtenidos usando una combinación de ecualización de histograma y técnica isotrópica basada en difusión [69]. Posiblemente si las condiciones de la imagen incluyeran bordes degradados o estructuras de una dimensión cerradas, técnicas anisotrópicas serían más adecuadas, tal como se sugiere en [179]. Usando estos tipos de técnicas se pueden obtener resultados tales como se muestra en la figura

La difusión isotrópica es basada en el suavizado de la imagen usando difusiones, siendo la imagen original la condición inicial del problema. Las ecuaciones de difusión son versiones discretas de las ecuaciones de conducción de calor.

$$I_{t+1}(x,y) = [I(x,y) + \frac{1}{\Omega}\sum_{d=1}^{\Omega} \nabla I_d(x,y)]_t \tag{2.10}$$

En (2.10)$I_{t+1}(x,y)$ es la imagen difusa, ω es el número de difusiones de orientación y $I_d(x,y)$ es la derivada direccional de la imagen en el pixel (x,y). En el ejemplo mostrado en la figura

2.5. Conclusiones

En este trabajo probamos cómo los clasificadores débiles basados en AdaBoost son muy exitosos para la tarea de la clasificación de rostro. Pese a que en [96] se muestra cómo los clasificadores basados en boosted learning tienen problemas cuando hay alguna fracción no nula de muestras etiquetadas incorrectamente en el entrenamiento (y, por extensión, muestras correctamente etiquetadas pero que no sean representativas de la clase), en el caso de detección de rostro la etiqueta de las muestras no es un problema. Basados en *boosting* conseguimos implementar un algoritmo de detección de rostro

con reducido número de falsos positivos y bajo tiempo de máquina aprovechando la información probabilística de localización de rostro en una secuencia de video. Nuestra hipótesis de que dada una secuencia de video la localización de rostros y no rostros en cuadros adyacentes/cercanos podría ayudar a la precisión de detección y tiempo de cálculo fue probada usando bases de datos e imágenes recopiladas por el autor. La mayor parte del tiempo de cálculo en el detector *boosted* convencional es usada al rechazar regiones que no son eliminadas por los detectores hasta una etapa tardía de filtrado o si el número de regiones de rostros en una imagen es elevado. Una región difícil de rechazar probablemente tenga que pasar por un buen número de nodos de clasificación antes de ser finalmente rechazada (o aceptada como falso positivo), mientras que una región facial pasa por todos los nodos. Si esto se ejecuta cuadro por cuadro, el costo de cálculo se mantiene similar sin importar la información proporcionada por los cuadros previos/futuros. Propusimos e implementamos una nueva aproximación, en la cual la información facial y no facial de un cuadro es considerada por el cuadro siguiente. En el nuevo cuadro, las regiones previamente detectadas como no rostro atraviesan una pirámide de clasificación recortada, mientras que las regiones previamente detectadas como rostro atraviesan sólo las etapas finales de clasificación. Esto fue posible debido a la independencia entre cada nodo de clasificación en la propuesta de Viola y Jones. Sin embargo, hay un problema inherente en la pirámide recortada para las regiones previamente detectadas como no rostro. Es posible que algunas de estas regiones eran difíciles de rechazar en los cuadros precedentes y luego, en una clasificación más simple, fueran en consecuencia aceptadas. Para evitar con ello la aparición de falsos positivos incluimos una tercera opción, al retirar temporalmente la región de los no rostros, pero sin etiquetarla como rostro, y en el siguiente cuadro de video la región incierta es tratada como rostro (es decir, clasificación rigurosa), para determinar con mayor certeza si efectivamente es rostro o era una región difícil, pero no facial. Gracias a que con esta aproximación se reduce el tiempo de cálculo tanto de las regiones no faciales como de las regiones faciales, se consiguió aumentar la disponibilidad de tiempo para las otras etapas del sistema.

Es importante señalar que nuestra propuesta es válida para la estructura piramidal convencional en la cual cada nodo de clasificación es independiente de los demás. En [23] ha sido sugerido el uso de nodos dependientes entre sí, tal que en teoría un nodo no debe preocuparse por rechazar regiones no faciales que de todas formas serían rechazadas por un nodo previo, y así enfocarse únicamente en las eventuales regiones no rechazadas que alcancen este nodo. Si bien esta aproximación puede conducir a una mejor precisión al especializar cada nodo, no sería apropiada para nuestra propuesta, pues no habría garantía de que un nodo sofisticado descarte una región no facial que hubiese sido descartada previamente.

Algunas posibles mejoras a nuestra propuesta que no fueron implementadas son por ejemplo que si una región es rechazada en un nodo relativamente tardío, podría pasar únicamente por nodos avanzados en la siguiente detección, asumiendo que hay alta probabilidad de que pueda evadir ser rechazado por los nodos menos sofisticados. Ocasionalmente esto podría empeorar el costo, por ejemplo si en la nueva clasificación

hubiese sido descartado rápidamente por un clasificador simple, pero creemos que en general el costo sería disminuido al evitar redundancia. Adicionalmente, consideramos que una aproximación Bayesiana en la cual las regiones no sean etiquetadas de forma binaria rostro/no rostro podría ayudar con una estimación *a priori* para la clasificación.

El problema de la corrección de iluminación presentó varios retos. El más importante probablemente es que la tarea no necesariamente es intuitiva. Según nuestras pruebas, diversas técnicas como manipulación de histograma, modelamiento de escena, corrección fotométrica y modelos retinales producen imágenes con diversa calidad visual. Sin embargo, para extracción de parámetros la calidad visual no es el criterio importante, sino qué tan adecuada sea la imagen procesada para extraer la información requerida para la clasificación. De esta forma, la corrección de la imagen está estrechamente relacionada con la etapa de extracción de parámetros y clasificación. Esto es crucial sobre todo cuando parte de la información específica de textura o detalles puede ser eliminada por algunos métodos de corrección de iluminación. Por ejemplo, en el caso particular de reconocimiento de la expresión facial algunas expresiones faciales tales como disgusto, tristeza e ira típicamente presentan gestos detallados específicos que deben ser preservados e, idealmente, acentuados en la etapa de procesamiento.

En nuestras pruebas con la base de datos CK+ presentamos imágenes procesadas elegidas según calidad visual determinada por observadores humanos. Luego el conjunto original y el conjunto procesado fueron usados para hacer una prueba de clasificación humana por parte de voluntarios. Tal como se esperaba, en las imágenes cuya calidad visual para los observadores humanos es mejor, la tasa de reconocimiento de la expresión facial es inferior. Las imágenes con mejor calidad para clasificación automática de expresión facial fueron basadas en técnicas de difusión y autocociente, en oposición a problemas de reconocimiento facial en las cuales el suavizado y la reducción de micro gestos ayuda en la tarea de reconocimiento [93]. Esto es consistente con las conclusiones de otros autores [172] [140] [61] y es razonable por cuanto los detalles locales son acentuados al usar este procedimiento. Sin embargo, los costos de procesamiento de técnicas basadas en *self-quotient* son un poco elevados aún para su aplicación actual en sistemas de tiempo real.

Dadas las restricciones del problema, incluyendo la preservación de parámetros descriptores de la expresión facial y reducido costo de cálculo, nuestra propuesta de detección de rostro y corrección de iluminación fue satisfactoria para el problema. En los capítulos posteriores veremos cómo este procesamiento fue conveniente para una adecuada extracción de parámetros usados en la clasificación de la expresión. Adicionalmente, observamos cómo el uso de algoritmos basados en MSQ más filtros retinales fuertemente mejora la información local sin incluir artefactos producidos por SQ y probablemente tengan aplicación futura, así como las técnicas de difusión, cuando se consigan tiempos de cálculo reducidos, quizás con el uso de procesamiento con GPU.

3. Extracción de parámetros de la expresión facial por codificación *Temporal Patterns of Oriented Edge Magnitudes* y *Volumetric Patterns of Oriented Edge Magnitudes*

3.1. Introducción

La extracción de parámetros es un tipo específico de reducción de dimensiones. El objetivo general es tomar un conjunto de datos de cierta dimensión y procesarlos con el fin de obtener un nuevo conjunto generalmente de menor dimensión, pero con mayor capacidad de representación del problema a describir [59]. Naturalmente, un algoritmo de extracción de parámetros no puede crear información a partir de un conjunto de datos inicial, pero la meta es obtener una nueva representación que sea más viable para la clasificación en términos de tamaño, ruido y relevancia de la información. De manera formal, la extracción de parámetros se puede definir como la transformación $f : X \rightarrow X'$ donde X es el conjunto original de datos y X' es el conjunto de datos transformados, generalmente de menor dimensión.

La etapa de extracción de parámetros de una imagen o una secuencia de video para el reconocimiento de la expresión facial supone la transformación de la información de los pixeles a un nuevo espacio dimensional, con mejor capacidad de representación. Esto es fundamental, por cuanto incluso en una sola imagen, aún de relativo bajo tamaño correspondiente a la región facial detectada, tiene una gran cantidad de información, buena parte de ella inútil e incluso perjudicial para solucionar el problema. En [178] se muestra cómo los datos de alta dimensión representan retos y dificultades para un problema de clasificación. Debido a esto, un sistema de clasificación que se base directamente en los datos de todos los pixeles no es práctico, pues la cantidad de información relevante de cada pixel es variable, el número de variables es considerable (128×128 datos por región facial extraída de este tamaño, para 16384 datos) y hay considerable ruido en la información. Entrenar un sistema de clasificación con los datos sin procesar es complejo y la clasificación es sujeta a diversas fuentes de error, tales como incapacidad del clasificador de modelar el problema, pobre convergencia del sistema de clasificación en el entrenamiento (problema usual en el uso de redes neuronales) y overfitting, cuando el sistema de clasificación es suficientemente poderoso para ajustarse bien a los datos de entrada de alta dimensión, pero con baja capacidad de generalización del sistema. En consecuencia, la etapa de extracción de parámetros debe obtener un conjunto de datos cuya capacidad de representación sea suficientemente adecuada para discriminar las expresiones faciales de los individuos, idealmente debe reducir el tamaño del conjunto de datos y, en el caso de nuestro objetivo de tiempo real, tener un costo de cálculo que se ajuste a la disponibilidad de tiempo para el cumplimento del tiempo real.

En este capítulo se presentará brevemente el fundamento teórico de técnicas de extracción de parámetros usadas en representación de rostros en 3.2. En 3.3 se mostrará el desarrollo y la implementación de los parámetros usados en este trabajo. En 3.4 se

mostrarán algunos resultados preliminares. No son resultados finales por cuanto en esta etapa del trabajo los sistemas de clasificación usados fueron bastante simples, con los parámetros extraídos sin procesar, pero el objetivo era determinar si los parámetros eran promisorios para la representación de la expresión facial. Finalmente, en 3.5 se muestran las conclusiones generales del trabajo desarrollado.

3.2. Fundamentos teóricos y estado del arte de extracción de parámetros para representación facial

En la figura se muestran ejemplos de variación dinámica de la expresión facial en la base de datos CK+. El objetivo de la extracción de parámetros es extraer información pertinente para la representación de las clases en el problema. En la representación de rostros y, en general, en la representación de texturas de una imagen, hay dos tipos de parámetros comúnmente usados, los parámetros basados en geometría y los parámetros basados en apariencia. Los parámetros basados en geometría son descriptores de la geometría del rostro dado un conjunto de puntos fiduciales, de manera que se caracteriza la forma del rostro y las deformaciones producidas en la expresión facial dependiendo de la localización relativa de estos puntos. El segundo tipo son los parámetros basados en apariencia, que describen texturas en la imagen que pueden ser descriptores de gestos, arrugas y pliegues en la piel producidos por los movimientos de los músculos faciales.

3.2.1. Parámetros basados en geometría

Los parámetros basados en geometría se usan para describir la localización y la forma del rostro en la imagen. Generalmente se detectan y siguen puntos fiduciales en localizaciones importantes específicas del rostro, con el fin de obtener la forma y tamaño de la boca, ojos y cejas, que son parámetros no transitorios, además de arrugas y gesto del ceño, que son parámetros transitorios. La diferencia entre los parámetros transitorios y no transitorios es que los parámetros transitorios no aparecen todo el tiempo en el rostro, sino que están temporalmente presentes debido a alguna activación muscular, mientras que los parámetros no transitorios están presentes todo el tiempo, pero su forma, localización y tamaño pueden cambiar debido a la expresión facial particular del individuo.

La mayor parte de trabajos usan modelos activos de apariencia [42] o técnicas similares para describir la localización y variación de los puntos fiduciales. Para describir las características geométricas del rostro, las distancias normalizadas entre puntos, tamaños relativos, ángulos y formas pueden ser usados para obtener un vector de parámetro característico, que es usado en el sistema de clasificación. Para detectar y seguir los puntos fiduciales se puede realizar de forma automática o con asistencia humana total o parcial. Idealmente el objetivo es que el sistema sea autónomo, entonces la meta es el seguimiento automático de los puntos fiduciales, pero esta tarea en tiempo real es aún complicada, pues se debe sacrificar precisión (menos puntos fiduciales y de localización automática más sencilla) o tiempo de estimación (menos cuadros por segundo).

Figura 10. Ejemplos de la base de datos CK+

En [123] [164] se usó una combinación de detectores y seguidores de puntos fiduciales basados en que cada clase de detector tiene buen desempeño bajo ciertas circunstancias, de manera que una combinación de detectores proporcionó un mejor resultado de seguimiento global. El rostro es descrito por el uso de 19 puntos frontales y 10 puntos laterales (valles y picos del perfil del rostro). El modelo propuesto fue adecuado para la descripción de 28 AUs (unidades de acción facial) y la tasa global de reconocimiento de 6 expresiones fue de 90.1 %, excluyendo 2 % de rostros cuyos puntos no fueron detectados. En [14] los puntos fiduciales son seguidos usando un algoritmo especializado de seguimiento de puntos y el reconocimiento de expresión es hecho usando un modelo de movimiento. Las variaciones dinámicas son descritas por el cambio de las medidas geométricas en el conjunto de datos. Los resultados muestran reconocimiento de expresión facial aceptable incluso con oclusión parcial del rostro. En [84] el usuario introduce manualmente los nodos de una malla *Candide* [2] en el primer cuadro de una secuencia de video y el sistema deforma la malla para que coincida con las imágenes de la secuencia. La clasificación se realiza con el rostro con mayor variación respecto del inicial con SVM o con unidades de acción facial, con 99.7 % de clasificación de 6 expresiones. En [25] un modelo activo de apariencia (AAM) fue usado con un perceptrón multi capa, con cerca de 99 % de reconocimiento. Más recientemente, en [53] se obtuvo cerca del 97 % de reconocimiento usando 52 puntos de referencia seguidos con *Elastic Graph Matching* (EGM)[162]. Los parámetros usados fueron las distancias y ángulos entre los puntos en un cuadro determinado y el cuadro inicial, tomado como instancia neutral del rostro.

3.2.2. Parámetros basados en apariencia

Los métodos basados en apariencia se fundamentan en la descripción de texturas en el rostro detectado. La idea general es que las texturas son descriptores adecuados para la representación de la expresión del rostro. La principal ventaja del uso de métodos basados en apariencia es que no requieren de la detección y el seguimiento de puntos fiduciales, cuyo cálculo es regularmente costoso y en algunos trabajos requiere de asistencia manual. Actualmente no hay consenso acerca de cuál aproximación es más adecuada, por cuanto se han conseguido resultados interesantes usando métodos basados en geometría y en apariencia.

El principal objetivo de los métodos usados en apariencia es transformar la información del rostro en una representación de menor dimensión que contenga información referente a la textura de las diferentes expresiones faciales. Algunas técnicas usadas son PCA, Análisis de componentes independientes (ICA) [28], Análisis de componentes principales con kernel (kPCA) [146], *wavelets*, parámetros Gabor [48] y códigos LBP [119]. PCA es de fácil ejecución con matrices de pequeña dimensión y produce información de reconstrución que, al ser truncada, sirve para comprimir los datos a un tamaño reducido. Los principales inconvenientes son que PCA es muy sensible al ruido y el nuevo espacio dimensional no necesariamente representa información local adecuada para la representación del problema. De hecho, la información codificada con PCA puede

eliminar detalles de la imagen que corresponden a micro gestos que son importantes para la expresión facial, pero que son descartados en PCA porque no representan un aporte energético sustancial para la reconstrucción. kPCA es un avance sobre PCA, por la linealización del *manifold*, pero el inconveniente de la representación del problema persiste. ICA, en cambio, es más adecuado para el problema, por cuanto intenta separar la información fuente que produce más datos locales, de manera que es más robusto ante cambios de iluminación, de perspectiva y oclusiones parciales. En [39] se implementaron y probaron varios métodos basados en apariencia, tales como PCA, ICA, Análisis local de parámetros (IFA)[126], Análisis lineal discriminante (LDA) y ondeletas Gabor y la conclusión fue que la representación Gabor produjo los mejores resultados. Como consecuencia, los parámetros relacionados con Gabor han sido usados en varios trabajos de reconocimiento de expresión facial, incluyendo trabajos recientes [95] [4] [57]. El principal inconveniente de las ondeletas Gabor es el alto costo de cálculo y los requisitos de memoria debido al banco de filtros usados en este procedimiento. Otra tendencia reciente en los métodos basados en apariencia es el uso de códigos de patrones locales binarios (LBP) y afines, cuyo cálculo es rápido y la representación es adecuada para definir el problema de la expresión facial en [3] [150] [49] [114]. En [193] se desarrollaron los códigos LBP para volúmenes (VLBP) como una extensión del LBP convencional que incluye información de un número de cuadros vecinos y así añadir información de transformaciones temporales en la secuencia. En [194] se introdujo LBP-TOP, una construcción LBP sobre tres planos ortogonales en la secuencia de imágenes. Los resultados usando LBP-TOP+VLBP están en el rango 88.77 %-96.26 % dependiendo de la complejidad del código usado y, hasta donde sabemos, son los mejores resultados obtenidos usando códigos basados en LBP.

3.2.3. Codificación LBP

Los patrones locales binarios fueron desarrollados por Ojala et al. en [119] [120]. En la codificación LBP el valor de un pixel es comparado con el de sus vecinos y el resultado es un código binario dependiente de la comparación. Los códigos LBP básicos son números binarios cuya longitud es igual al número de vecinos P considerados en la ecuación. El cálculo de LBP es mostrado en la ecuación 3.1.

$$LBP_{P,R} = \sum_{p=0}^{P-1} f(g_p - g_c)2^p \tag{3.1}$$

$$f(x) = \begin{cases} 1, \text{ si } x \geq 0 \\ 0, \text{ en otro caso} \end{cases} \tag{3.2}$$

donde P es el número de vecinos, R es el radio de la vecindad, g_p es el valor de intensidad de los vecinos, g_c es la intensidad del pixel central. En la codificación LBP original se usaban los 8 vecinos adyacentes del pixel central, pero en LBP general el número de vecinos y el radio usado dependen de la aplicación.

Los códigos LBP son considerablemente redundantes en la descripción de textura. Por ejemplo, hay códigos diferentes si la imagen es rotada o reflejada, pero un descriptor de textura debería ser invariante a estas transformaciones. Más aún, texturas muy similares podrían ser codificadas por distintos códigos LBP, lo cual no sólo es redundante, sino puede producir problemas de clasificación. Consecuentemente, la primera aproximación para abordar estos inconvenientes fue la inclusión de patrones uniformes en la codificación LBP. Si el número de transiciones entre 0 y 1 y viceversa es igual o menor que dos, el código LBP es considerado uniforme y todos los códigos uniformes tienen su propia etiqueta, mientras que todos los códigos no uniformes son representados por la misma etiqueta, de manera que el número de códigos LBP es considerablemente reducido. Mapeos más sofisticados sirven para agrupar texturas similares en el mismo código LBP, con el fin de reducir más la dimensión del vector de salida, pero también para prevenir problemas de sobreajuste (overfitting) o de pobre generalización en la etapa de clasificación si se tiene que diferentes códigos corresponden a entradas muy similares y códigos similares corresponden a entradas muy distintas.

Una vez el código LBP es obtenido para cada pixel en la imagen, una técnica generalmente usada es dividir la imagen en una grilla de celdas. La estructura de la grilla es variable, pero generalmente es aconsejable usar tamaños mayores que el posible número de códigos LBP, de manera que haya información estadística útil sobre cada celda. El histograma de códigos LBP se calcula según la ecuación 3.3. El histograma obtenido puede ser usado como un descriptor de textura para un sistema de clasificación o para segmentación de imágenes [118].

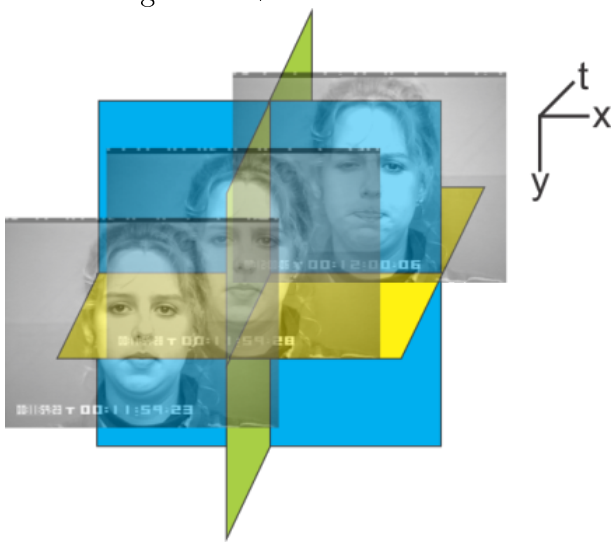
$$H_i = \sum_{x,y} I(LBP(x,y) = i), \quad i = 0, ..., n-1 \tag{3.3}$$

$$f(x) = \begin{cases} 1, & \text{if } x \text{ es verdadero} \\ 0, & \text{en otro caso} \end{cases} \tag{3.4}$$

LBP ha probado ser un descriptor eficiente de textura, de maner que ha sido ampliamente usado en diferentes aplicaciones directamente o con adaptaciones. En [175] se usó LBP integral en combinación con Histogramas de gradientes orientados (HOG) para implementar un sistema de detección de humanos. En [68] un LBP modificado llamado LBP con simetría central (CS-LBP) es usado como un parámetro local por transformación invariante en escala (SIFT) [97] obteniendo un descriptor de region cuyo desempeño es mejor que con SIFT convencional. En [190] el reconocimiento facial fue realizado usando LBP en subventanas escalables y métodos *boosted* para aprender la discriminación entre rostros. En [192] se implementó una variación de LBP adecuada para describir variaciones temporales en una secuencia de video usando LBP con histograma de Fourier (LBP-HF) para representación estática e implementación invariable a rotación del algoritmo LBP-TOP.

En [194] se implementó el algoritmo VLBP, para el uso en secuencias de video. El objetivo de VLBP es incluir información temporal en el descriptor. Una secuencia de video puede ser descrita como una matriz tridimensional $\{X, Y, T\}$ donde X e Y

Figura 11. Orientaciones ortogonales VLBP



son los componentes espaciales y T es el componente temporal. De manera similar a LBP, se define una vecindad, incluyendo vecinos temporales en los datos. Una vez una vecindad tridimensional es definida usando un patrón determinado, por ejemplo una esfera alrededor del punto central, se usa LBP invariante a rotación. En [195] se incluyó el uso de patrones uniformes y no uniformes en el cálculo de VLPB. La principal desventaja de VLBP es que los cálculos son demandantes al requerir interpolación 3D para obtener la vecindad y la aplicación de codificación LBP invariante en una vecindad extensa requiere de considerable tiempo de cómputo adicional comparado con las vecindades LBP típicamente reducidas. La modificación LBP-TOP descrita en [193] usa una vecindad tridimensional más simple que en VLBP. Dado XY el plano espacial convencional y XT y YT los planos espacio-temporales, una vecindad puede ser definida para cada uno de estos planos. La forma de la vecindad no necesariamente es cuadrada, por cuanto si bien los planos X e Y tienen dimensiones espaciales, no hay una relación directa entre estos planos espaciales y la dimensión del espacio temporal T. Los planos ortogonales se muestran en 11

De acuerdo con la bibliografía consultada, usando la codificación VLBP+LBP-TOP se ha obtenido la más alta precisión hasta ahora usando parámetros basados en LBP. Los descriptores usados en los mejores resultados son $LBP-TOP^{u2}_{8,8,8,3,3,3} + VLBP_{3,2,3}$. Los subíndices indican que VLBP es realizado con intervalo de tiempo 3, 2 vecinos por cuadro y radio 3, mientras que LBP-TOP fue ejecutado con 8 vecinos por plano, para un tamaño de codificación total de 776 bits por código.

3.2.4. Codificación POEM

La codificación POEM [169] fue desarrollada para combinar características deseables de métodos orientados a parámetros locales detallados y métodos más orientados

globalmente. El parámetro POEM es construido usando la idea de la auto similaridad en las codificaciones LBP con la distribución del borde local en distintas orientaciones espaciales. La combinación de la información de bordes y formas locales con la relación entre regiones vecinas proporciona una adecuada caracterización de los objetos en la imagen. Adicionalmente, el cálculo de los parámetros POEM es relativamente sencillo, especialmente comparado con variaciones sofisticadas basadas en LBP, de manera que es aplicable en sistemas de tiempo real.

Los parámetros POEM son calculados al usar las magnitudes de los gradientes orientados por acumulación de un histograma local de direcciones del gradiente sobre los pixeles de una región espacial llamada celda. Este cálculo es realizado sobre un número determinado de orientaciones espaciales alrededor del pixel central.

La diferencia de POEM con los algoritmos HOG es que los parámetros HOG son calculados sobre una malla densa para obtener la representación de la celda, mientras que en POEM el histograma local de gradientes es calculado alrededor del pixel central para obtener la representación del pixel. Como tal, los algoritmos POEM caracterizan objetos en una pequeña escala y luego el algoritmo basado en LBP es usado para codificar regiones más grandes, en oposición a las técnicas convencionales LBP+global, que caracterizan información global y luego codifican los detalles usando algoritmos basados en LBP.

El primer paso para extraer los parámetros POEM es calcular la imagen gradiente en un número de orientaciones espaciales. La salida es un arreglo de matrices con intensidades que definen la magnitud del gradiente. Posteriormente se obtiene el histograma local de bordes orientados sobre las celdas. La ponderación usada (peso) es la magnitud de la intensidad del pixel. La importancia del pixel central puede ser enfatizada usando una ventana tal como un filtro Gaussiano, aunque en nuestro caso la precisión no mejoró al usar este tipo de procedimiento. Esto sucede porque un filtrado es conveniente para eliminar detalles finos y microgestos, conveniente para el reconocimiento facial, que es el objetivo inicial de la codificación POEM. Sin embargo, el objetivo del reconocimiento de la expresión facial requiere de estos detalles finos y microgestos, de modo que atenuarlos con filtrado Gaussiano dificulta la resolución del problema, por cuanto atenúa o incluso destruye microexpresiones relevantes para la diferenciación de la expresión facial. La codificación es realizada sobre cada orientación de magnitudes de gradiente, de manera que el código POEM final es la concatenación de cada orientación POEM.

La codificación completa es descrita por la ecuación (3.5).

$$POEM_{W,\theta_n,P,R} = \sum_{j=0}^{P-1} s(S(i_{p,\theta_n} - i_{c,\theta_n}))2^j \tag{3.5}$$

donde W es el tamaño del bloque, θ_n son las orientaciones de gradiente, P es el número de vecinos, R es el radio de la vecindad, $f(\cdot)$ es equivalente a la función s en LBP descrita previamente, $S(\cdot)$ es la función de semejanza e i_{p,θ_n} e i_{c,θ_n} son los gradientes orientados espacialmente.

Las principales características de POEM son:

1. El cálculo de los parámetros POEM es rápido por cuanto se basa en las magnitudes de bordes orientadas acumuladas, cuyo cálculo es rápido con el uso de imágenes integrales, y codificación LBP, de bajo costo de cálculo.

2. La codificación POEM es flexible, con la opción de modificar el número de orientaciones y el tamaño de las celdas, así como los parámetros convencionales LBP, lo que permite usar parámetros especializados según la aplicación.

3. Los magnitudes de bordes orientados son resistentes a cambios de iluminación. Debido a esta característica, la etapa de corrección de iluminación y procesamiento del rostro puede ser simplificada, lo que reduce lor requerimientos de cálculo del sistema completo.

4. POEM calcula disimilitudes entre celdas, así que puede ser usado para obtener semejanzas entre regiones de la imagen. Esto hace a POEM un parámetro adecuado para tratar problemas inherentes en el reconocimiento de la expresión, tales como rotaciones, translaciones y oclusiones en regiones de la imagen.

3.3. Codificación VPOEM y TPOEM

El reto más importante en este trabajo es la restricción impuesta por los cálculos en tiempo real, especialmente dado el número de etapas del sistema global (captura, detección de rostro, procesamiento de imagen, extracción de parámetros, procesamiento de parámetros y clasificación). De esta forma, el uso de parámetros descriptores de geometría fue descartado, por cuanto generalmente su costo de máquina es muy exigente y probablemente prohibitivo para alcanzar el objetivo de tiempo real. Los trabajos que se basan en descriptores geométricos pueden producir excelente precisión en el reconocimiento de la expresión facial. No obstante, el compromiso es en algunos casos el requerimiento de asistencia humana en algunas etapas del proceso y el seguimiento de un elevado número de puntos fiduciales, de alto costo de procesamiento. En oposición, el uso de códigos basados en LBP permite obtener alta precisión manteniendo los costos computacionales bajos.

En [169] [170] POEM fue desarrollado y adaptado como descriptor para la identificación de rostros, con adecuados resultados. La codificación POEM fue estudiada y adaptada para la descripción de la expresión facial. Si bien las tareas de identificación de rostros y reconocimiento de la expresión facial pueden parecer similares, en realidad son muy diferentes, de manera que se requieren abordar por técnicas distintas. La identificación facial requiere principalmente descriptores relativamente globales, como formas y posiciones de objetos en el rostro, descartando información muy detallada como gestos finos o elementos transitorios en la imagen. En cambio el reconocimiento de la expresión facial debe eliminar las variaciones inter individuo (esto es, reconocer la expresión sin importar la persona), para lo cual es necesario obtener elementos

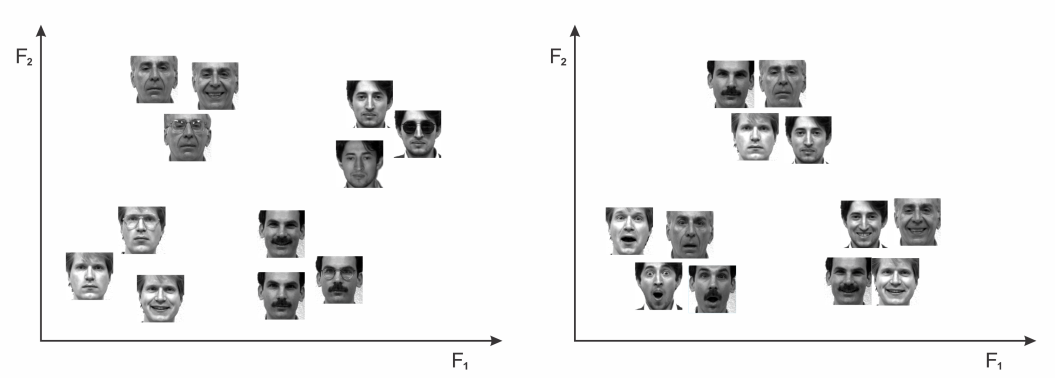
descriptivos de la expresión facial, que generalmente son detalles en el rostro. Debido a esto, la secuencia de histograma con PCA blanqueado (WPCA) POEM (*Whitened* PCA-POEM-*Histogram Sequence*, WPCA-POEM-HS) usada en [170] para la identificación facial no es apropiada para el reconocimiento de la expresión facial. De acuerdo con nuestras pruebas, la reducción de dimensiones con WPCA consigue una reducción del tamaño de datos de aproximadamente 50-60 %. Sin embargo, el conjunto de datos transformado no mejoró las tasas de clasificación y, en el caso de algunas expresiones, empeoró notablemente los resultados. Esto ocurre por la características previamente reseñadas: el reconocimiento facial no requiere de información fina de textura, al contrario de los requisitos de reconocimiento de expresión. Es así, entonces, que la tarea de identificación facial es relativamente más simple incluso con imágenes suavizadas, mientras que el reconocimiento de la expresión facial es un problema más complicado con este tipo de imágenes, por ejemplo imágenes que han sido procesadas con corrección de iluminación que incluya filtrado pasa-bajo o imágenes de muy baja resolución espacial.

Otras modificaciones que se realizaron a los parámetros POEM originales fueron la eliminación del filtrado suavizado Gaussiano, que atenúa gestos finos detallados en el rostro, y la realización de otro tipo de codificación. La codificación POEM original es codificación uniforme. Sin embargo, para este trabajo usamos codificación no uniforme con mapeo basado en entrenamiento por texturas. Para este procedimiento usamos bancos de distintas texturas, obtuvimos los códigos POEM correspondientes para las distintas texturas y el mapeo fue realizado a partir de la información extraída, de manera que se garantizó que texturas similares tienen el mismo código POEM y texturas muy parecidas generalmente tienen códigos parecidos en distancia Euclidiana. El desarrollo fue automático, usando un millón de texturas generadas por un algoritmo, de modo que no fue necesario etiquetar manualmente las texturas y así se simplificó el tiempo de desarrollo de la nueva codificación.

Adicionalmente, los descriptores POEM se basan en imágenes estáticas, de manera que no incluyen información temporal, que es información poderosa para la descripción de la expresión del rostro. Detectar y evaluar estas transformaciones sirve para mejorar las posibilidades de clasificación de la expresión. Tal como se reseñó en la sección (3.2.3), extensiones temporales de los algoritmos LBP como VLBP y LBP-TOP han sido muy apropiadas para la descripción de las texturas dinámicas, de manera que en esta sección hacemos pruebas preliminares de uso de códigos POEM adaptados para reconocimiento de expresión facial y posteriormente desarrollamos codificación que incluye información temporal, así como algunas pruebas iniciales de resultados con sistemas básicos de clasificación.

Es importante señalar que los parámetros extraídos que pueden ser valiosos en un problema de identificación facial, eventualmente no son convenientes para un problema de reconocimiento de la expresión facial y viceversa. Esto sucede porque las características que permiten evaluar la expresión facial son distintas que las necesarias para distinguir entre distintos rostros. En general, los parámetros globales son adecuados para distinguir rostros, mientras que los parámetros de texturas y microgestos son más

Figura 12. Reconocimiento facial y reconocimiento de la expresión facial



importantes para reconocer las expresiones faciales. Ilustramos esto en la figura 12.

En la figura se usan muestras tomadas de la base de datos Yale [52]. En la izquierda se muestra un problema de reconocimiento facial. En este problema se espera que los parámetros (en este caso, para efectos de visualización, 2 parámetros, F_1 y F_2) minimicen las distancias intraclase (imágenes del mismo individuo) mientras que las distancias interclase (imágenes de distintos individuos) sean maximizadas. En la derecha el problema es de reconocimiento de la expresión facial. En este caso los parámetros extraídos deben obtener distinto tipo de información, por cuanto el atributo deseado en el problema de reconocimiento facial se convierte en algo indeseable y los parámetros ahora deben minimizar distancias entre individuos con la misma expresión facial, maximizando distancias entre individuos con distinta expresión facial, así pertenezcan a la misma persona. Esto hace que si bien en principio el problema de reconocimiento de la expresión facial pueda ser directamente relacionado con el problema de reconocimiento facial, en realidad las características de los descriptores son muy distintas y requieren de una aproximación diferente. Debido a esto la necesidad de implementar parámetros de otra naturaleza y, para los parámetros POEM, usar distinto mapeo de codificación especializado en expresión facial.

3.3.1. Pruebas de los parámetros POEM en la caracterización de la expresión facial

Con el fin de determinar si los parámetros POEM modificados son descriptores válidos de la expresión facial, el conjunto de pruebas iniciales realizadas fue usando la base de datos CK, aunque posteriormente, en cuanto se recibió autorización de uso de la base de datos CK+, se repitieron estas pruebas con el fin de ser consignadas en la sección 3.4, aún cuando ya se había alcanzado el objetivo de determinar la validez de los parámetros como descriptores de la expresión. [1]. En esta parte del desarrollo

[1]En esta etapa del trabajo aún no se había recibido autorización de descarga y uso de la base de datos CK+, de manera que las pruebas preliminares se hicieron usando la base de datos Cohn-Kanade y posteriormente, cuando se obtuvo la autorización, el resto del trabajo se hizo con la base de

se tomaron 3 imágenes por individuo por expresión, más 1 imagen correspondiente a una instancia neutral. En cada muestra se obtuvieron códigos POEM sobre una región facial detectada y convertida a tamaño 128×128. Los parámetros iniciales fueron fijos, con 8×8 celdas, 8 número de vecinos (posteriormente reducidos a 5 para el algoritmo TPOEM final), 3 orientaciones de gradientes y codificación no uniforme. La salida es 192 vectores codificados por imagen. Los datos fueron etiquetados según la expresión y el individuo y se obtuvieron los conjuntos de entrenamiento y validación usando LSO [2]. El conjunto de datos de entrenamiento fue definido como $\Omega_1(x)_{i,k,n}$, donde x es el vector POEM, k es la celda POEM y n es el número de la muestra, i es la expresión (1, ira; 2, disgusto; 3, miedo y así sucesivamente). x, i, k, n serán omitidos de ahora en adelante en la nomenclatura por simplicidad. El conjunto de validación es un conjunto disjunto denominado Ω_2, usando LSO. Con el conjunto de entrenamiento definido, se obtuvo el código POEM prototipo como el promedio mostrado en la ecuación (3.6).

$$Av(x)_{i,k} = \frac{1}{N} \sum_{n=1}^{N_i} \Omega_1 \tag{3.6}$$

donde N_i es el número de muestras por expresión i

Clasificación chi-cuadrado: La métrica usada para la validación fue chi-cuadrado. Cada imagen del conjunto de validación $Val(x)_{k,n}$ fue comparada con el conjunto de entrenamiento por expresión y los resultados fueron almacenados en el vector $D_{i,k,n}$, tal como se muestra en la ecuación (3.7).

$$D_{i,k,n} = chi_i^2(Val_{k,n}, Av_{i,k}) = \sum_x \frac{(Av_{i,k} - Val_{k,n})^2}{Av_{i,k} + Val_{k,n}} \tag{3.7}$$
$$i = 1, ..., 7;\ k = 1, ..., K;\ n = 1, ..., N_i$$

La salida de la ecuación es un vector de distancias chi-cuadrado entre cada código POEM por celda en la imagen de validación y la celda correspondiente en cada conjunto de entrenamiento por expresión. Se espera que los valores más pequeños correspondan a alta similitud entre la celda de la imagen y la expresión respectiva.

---

datos mejorada y extendida CK+. Incidentalmente, las principales modificaciones de la base de datos CK+ consistieron en eliminar muestras que estaban erróneamente etiquetadas, sea porque pertenecían a alguna expresión distinta a la etiqueta inicial o porque en realidad no eran representativas de la expresión facial. Nuestras pruebas iniciales, realizadas con la base de datos Cohn-Kanade original, etiquetaron "incorrectamente" la mayor parte de las muestras que habrían de ser eliminadas en la base de datos CK+. Es decir, nuestro sistema inicial tuvo la capacidad de, sin saberlo, descartar muestras no representativas de la expresión.

[2] Tal como se verá en el capítulo 7, en muchos trabajos la validación se realiza usando *random 10-folded*, pero esta metodología puede conducir a tasas artificialmente elevadas de clasificación por aprendizaje negativo o por *overtraining*. En el capítulo referido hacemos el análisis y las pruebas correspondientes.

El clasificador simple es obtenido sumando la distancia por celda tal como se muestra en la ecuación (3.8).

$$M_{i,n} = \sum_{k=1}^{K} D_{i,k,n} \tag{3.8}$$

La decisión de clasificación es dada por (3.9).

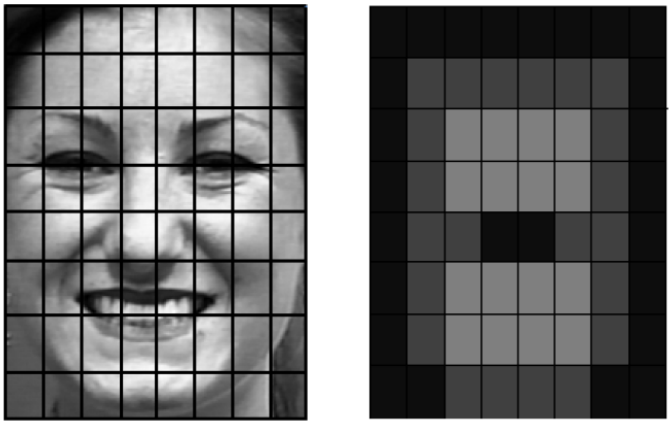
$$c_n = \underset{i}{\text{argmín}}\, M_{i,n} \tag{3.9}$$

La tasa global de acierto para clasificación de 7 expresiones es de 79.9 % y para 6 expresiones es de 90.4 %. En la sección (3.4) se muestran con mayor profundidad los resultados de clasificación con este sistema básico. En este trabajo se entrenan y validan sistemas de clasificación tanto de 7 expresiones como de 6 expresiones. En nuestra opinión, la clasificación de 6 expresiones es por lo menos defectuosa en su fundamento, pues implica que el rostro es clasificado como perteneciente a alguna expresión facial, lo cual no necesariamente es el caso la mayor parte del tiempo y, por lo tanto, la aplicabilidad real de la clasificación de 6 expresiones sin incluir instancia neutral es limitada. Sin embargo, incluimos este tipo de clasificación por cuanto numerosos trabajos en el área usan clasificación de 6 expresiones, de manera que sea posible hacer comparación.

Es de señalar que el uso de un vector prototipo promedio por expresión para realizar la clasificación es una técnica muy simple y sujeta a error, pero esto fue deliberado por varias razones. Inicialmente, porque se pretendía determinar la validez de los parámetros POEM adaptados como descriptores de la expresión del rostro, y como tal la idea era probar su desempeño incluso con clasificadores muy limitados. Por otra parte, si bien esta metodología de clasificación es muy básica, no es susceptible de overfitting. Al contrario, al promediar las muestras por expresión para obtener el conjunto prototipo, las muestras *outliners*, que suelen producir dobleces y singularidades en los sistemas de clasificación expertos, son en cambio homogeneizadas, lo cual, sumado a la metodología LSO, prácticamente elimina la posibilidad de sobreentrenamiento. Por último, en este capítulo se usan los datos completos extraídos por los algoritmos desarrollados, sin hacer ningún procedimiento de extracción y detección de parámetros o reducción de dimensiones. Estas etapas del desarrollo se abordan en los capítulos correspondientes.

Ponderación manual de pesos de las celdas: La clasificación previa muestra que todas las distancias por celda son igualmente importantes en la clasificación. Es decir, en el sistema de clasificación todas las celdas espaciales tienen igual ponderación. Sin embargo, esto es una simplificación drástica que puede deteriorar los resultados de la clasifación, por cuanto es razonable asumir que algunas celdas proporcionan más información sobre la expresión facial. En consecuencia, se realizó una modificación sencilla, consistente en asignar pesos manuales a las celdas espaciales. Si bien hay un grado de arbitrariedad al asignar pesos manualmente, para la expresión facial tenemos información empírica adicional, pues podemos determinar *a priori* con cierta certeza cuáles

Figura 13. Asignación manual de pesos por codificación POEM



regiones del rostro son más importantes en la definición de la expresión del rostro.

En la figura 13 se muestra la asignación manual de los pesos por celdas espaciales. En negro, ponderación 1; en gris oscuro, ponderación 2, y en gris claro, ponderación 4. La tasa de acierto de reconocimiento de 7 expresiones es de 81 % y de reconocimiento de 6 expresiones es de 91 %. Es decir, incluso una modificación elemental como asignar manualmente pesos de las celdas puede mejorar los resultados de clasificación. En la sección (3.4) se muestran los resultados completos.

Estimación de los pesos por celda POEM: Para las celdas más relevantes en la clasificación, $D_{i,k,n}$ debería producir valores pequeños promedio para la clase correcta (i.e. cuando $c(n) = i$) y valores grandes para el resto de clases (cuando $c(n) \neq i$). Dado esto, el error normalizado por celda para la etiqueta correcta dado por (3.10)

$$\epsilon_k = \sum_{n \in \Omega_2} \frac{D_{c(n),k,n}}{\sum_{i=1}^{I} D_{i,k,n}}, \tag{3.10}$$

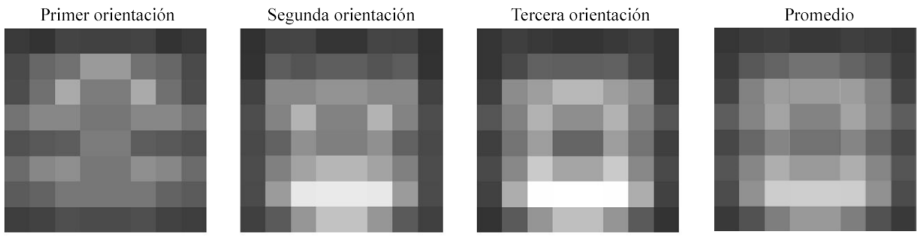
debería producir el valor ideal cero para las celdas relevantes y valores cercanos a uno para las celdas con pobre capacidad de discriminación. Para determinar el valor w_k de cada celda, escogimos una función monotónicamente decreciente en función de ϵ_k y normalizamos los valores tal que $\sum w_k = 1$, de acuerdo con (3.11)

$$w_k = \frac{1}{\sum_{k=1}^{K} e^{-\alpha\epsilon_k}} e^{-\alpha\epsilon_k}, \tag{3.11}$$

donde α controla el decaimiento de la función exponencial.

Finalmente, teniendo en cuenta que en el rostro hay cercana simetría con el eje vertical, se obtuvo simetría de los pesos w_k al promediar los pesos originales con una versión horizontalmente reflejada de los pesos. Los resultados se muestran en la figura 14. Nótese cómo tanto en las tres orientaciones como en el promedio las regiones de mayores coeficientes corresponden en general a la boca, los ojos, el ceño y regiones

Figura 14. Estimación de pesos por orientación POEM



cercanas a la boca, lo cual es esperado por cuanto son regiones importantes en la descripción de la expresión facial.

Usando estimación de pesos de las celdas, los resultados de clasificación de 7 expresiones son de 83.3 % y de 6 expresiones son de 93.1 %, mejorías notables comparado con los resultados iniciales de 79.9 % y 90.4 % respectivamente.

Debemos apuntar, sin embargo, que para la estimación de los pesos de las celdas siempre se usaron datos de Ω_1. Esto es muy importante por cuanto si en la estimación de la ponderación se usan datos del conjunto completo o al menos datos que estén posiblemente incluidos en el conjunto de validación Ω_2, se constituye un error metodológico. Esto ocurre porque los pesos tienden a ser menores cuando el error es mayor, lo cual es natural, pues es efectivamente lo que se pretende. Sin embargo, si en la obtención de los pesos hay muestras que eventualmente van a ser usadas en la validación, estas muestras ya le han servido al clasificador para aprender en la estimación de pesos en cuáles celdas tienen error elevado, contribuyendo así a la reducción de estos pesos y, posteriormente, siendo posiblemente correctamente clasificadas pese a que de otra manera habrían sido clasificadas erróneamente. Si bien esta observación puede parecer obvia y redundante, existen numerosos trabajos tanto en reconocimiento de expresión facial como en sistemas de clasificación en general en los cuales etapas previas a la clasificación, tales como reducción supervisada de dimensiones, estimación de variables o extracción/detección de parámetros, son realizadas incluyendo muestras que participan también del proceso de validación, obteniendo así valores artificialmente elevados de clasificación debido al error metodológico.

3.3.2. *Volume Patterns of Oriented Edge Magnitudes* (VPOEM)

El reconocimiento de la expresión facial puede ser implementado mediante la evaluación de una imagen o un cuadro individual en una secuencia de video. Sin embargo, siendo la expresión facial una transición temporal entre una instancia neutral y una expresión pico (o entre una expresión y otra expresión), se puede determinar que hay información adicional significativa en la secuencia de video. Los resultados preliminares han sido basados en cuadros individuales de video. En esta sección introducimos el nuevo algoritmo *Volume Paterns of Oriented Edge Magnitudes* VPOEM, que es una extensión en volumen del algoritmo POEM convencional. En VPOEM información de vecinos temporales es incluida en la codificación. Se espera que al incluir información

de otros cuadros cercanos, la codificación VPOEM sea adecuada para introducir información concerniente a transiciones temporales de formas y texturas y, como tal, codificar la naturaleza dinámica de la expresión facial con el fin de obtener una mejor representación de la expresión.

La vecindad usada en la codificación VPOEM está dada por la ecuación 3.12.

$$V = \begin{pmatrix} i_{t_c-T,c,\theta_0}, i_{t_c-T,0,\theta_0}, \cdots, i_{t_c-T,P-1,\theta_0}, \\ i_{t_c,c,\theta_0}, i_{t_c,0,\theta_0}, \cdots, i_{t_c,P-1,\theta_0}, \\ i_{t_c+T,c,\theta_0}, i_{t_c+T,0,\theta_0}, \cdots, i_{t_c+T,P-1,\theta_0}, \cdots, \\ i_{t_c-T,c,\theta_{N-1}}, i_{t_c-T,0,\theta_{N-1}}, \ldots, i_{t_c-T,P-1,\theta_{N-1}}, \\ i_{t_c,c,\theta_{N-1}}, i_{t_c,0,\theta_{N-1}}, \cdots, i_{t_c,P-1,\theta_{N-1}}, \\ i_{t_c+T,c,\theta_{N-1}}, i_{t_c+T,0,\theta_{N-1}}, \cdots, i_{t_c+T,P-1,\theta_{N-1}} \end{pmatrix} \tag{3.12}$$

i_{t_c-T,c,θ_n} es el valor del gradiente acumulado del pixel central en el cuadro $t_c - T$, donde T es un número de cuadros definido para la vecindad, y orientación espacial θ_n. i_{t_c-T,p,θ_n} es el gradiente acumulado del pixel vecino p en el cuadro $t_c - T$, orientación θ_n, en ambos casos para $n \in \{0, ..., N-1\}$, donde N es el número de orientaciones espaciales, en el pixel $p \in \{0, ..., P-1, c\}$ perteneciente a una vecindad de radio R. Para obtener el gradiente acumulado de P vecinos se usa un círculo de radio R alrededor del pixel central. Dado un pixel central localizado en $(x_c, y_c, t_c, \theta_n)$, el valor de los P vecinos está dado por la ecuación 3.13.

$$i_{t_c,p,\theta_n} = i_{t_c,x_c+R\cos(2\pi p/P),y_c-R\sin(2\pi p/P),\theta_n} \tag{3.13}$$

En tanto que la secuencia de video es de naturaleza discreta, los valores de vecindad que no corresponden a localizaciones exactas de pixeles son interpoladas para obtener una aproximación. Una vez la vecindad V es obtenida, el valor del gradiente acumulado central por orientación espacial es restado del vector de volumen V, similar al proceso de LBP convencional en el cual el valor del pixel central es restado del valor de los pixeles, tal como se muestra en la ecuación 3.14.

$$V = \begin{pmatrix} i_{t_c-T,c,\theta_0} - i_{t_c,c,\theta_0}, i_{t_c-T,0,\theta_0} - i_{t_c,c,\theta_0}, \cdots, \\ i_{t_c-T,P-1,\theta_0} - i_{t_c,c,\theta_0}, 0, i_{t_c,0,\theta_0} - i_{t_c,c,\theta_0}, \cdots, \\ i_{t_c,P-1,\theta_0} - i_{t_c,c,\theta_0}, i_{t_c+T,c,\theta_0} - i_{t_c,c,\theta_0}, \\ i_{t_c+T,0,\theta_0} - i_{t_c,c,\theta_0}, \cdots, i_{t_c+T,P-1,\theta_0} - i_{t_c,c,\theta_0}, \cdots, \\ i_{t_c-T,c,\theta_{N-1}} - i_{t_c,c,\theta_{N-1}}, i_{t_c-T,0,\theta_{N-1}} - i_{t_c,c,\theta_{N-1}}, \ldots, \\ i_{t_c-T,P-1,\theta_{N-1}} - i_{t_c,c,\theta_{N-1}}, 0, i_{t_c,0,\theta_{N-1}} - i_{t_c,c,\theta_{N-1}}, \cdots, \\ i_{t_c,P-1,\theta_{N-1}} - i_{t_c,c,\theta_{N-1}}, \\ i_{t_c+T,c,\theta_{N-1}} - i_{t_c,c,\theta_{N-1}}, i_{t_c+T,0,\theta_{N-1}} - i_{t_c,c,\theta_{N-1}}, \cdots, \\ i_{t_c+T,P-1,\theta_{N-1}} - i_{t_c,c,\theta_{N-1}} \end{pmatrix} \tag{3.14}$$

Si se asume que la diferencia entre las magnitudes acumuladas de los gradientes en los pixeles vecinos y el pixel central es independiente, de manera similar al procedimiento LBP, se pueden eliminar los valores cero del vector. En realidad no hay independencia entre los dos valores, pero la ventaja de hacer esta simplificación es que

la nueva codificación es más robusta a cambios de iluminación en la secuencia original. En tanto que las anteriores ecuaciones son un poco extensas, introducimos una notificación simplificada dado que el proceso de obtención de la vecindad VPOEM ya está establecido previamente. Incluimos las ecuaciones anteriores pese a su extensión para evitar que la simplificación pueda introducir confusiones en la determinación de la vecindad VPOEM. Dados los dos cuadros vecinos en $t-T$ y $t+T$, el vector puede ser definido de manera simplificada según la ecuación 3.17.

$$V(\theta_n, t) = [V_{\theta_n,t-T}\ V_{\theta_n,t}\ V_{\theta_n,t+T}], \text{where} \tag{3.15}$$

$$V_{\theta_n,t\pm T} = [i_{t\pm T,p,\theta_n} - i_{t,c,\theta_n}],\ \text{for } p \in \{0, ..., P-1, c\} \tag{3.16}$$

$$V_{\theta_n,t} = [i_{t,p,\theta_n} - i_{t,c,\theta_n}],\ \text{for } p \in \{0, ..., P-1\} \tag{3.17}$$

$n = 0, ..., N-1$. Tal como se puede observar, la longitud de $V(\theta_n, t)$ es $N(P+1) + NP + N(P+1) = N(3P+2)$. Una vez la textura volumétrica es obtenida, el código VPOEM es calculado de acuerdo con:

$$VPOEM_{T,P,R,N}(\theta_n, t) = \sum_{q=0}^{N(3P+2)-1} f(v_{\theta_n,q})2^q, \tag{3.18}$$

donde $v_{\theta_n,q}$ es el q-th elemento de $V(\theta_n, t)$, y f es la función definida en 3.1.

En [193] se introdujo una modificación en volumen invariante a rotación. Si bien una aproximación similar puede ser desarrollada para VPOEM, decidimos descartarlo por varios inconvenientes. En principio, el cálculo de rotaciones circulares de los códigos requiere de un considerable costo de cálculo. Adicionalmente, se espera que la rotación de las texturas para el reconocimiento de la expresión facial sea pequeña debido a la alineación del rostro, que típicamente es vertical o cercano a vertical en una imagen (y en caso de no serlo, un proceso de alineación de rostro previo es más expedito que la aplicación de diversas rotaciones circulares en los códigos). Por último, los resultados en [193] mostraron que la introducción de códigos invariantes a rotación para texturas dinámicas no mejoró clasificación sino en casos muy específicos, mientras que el costo de cálculo se incrementa.

En nuestro trabajo se usó vecindad cilíndrica en vez de vecindad esférica. La principal razón de ello es mantener costo computacional bajo control: cuando se usa una región esférica, toda la textura dinámica V debe ser recalculada cuando el cuadro analizado t cambia. Con la vecindad propuesta, t depende de $t-T$ y $t+T$. Es decir, la vecindad actual presente y futura t y $t+T$ de un cuadro en t incluye la vecindad pasada y presente para el cuadro $t+T$, cosa que no sucede si la vecindad es esférica. Si adicionalmente avanzamos en pasos de valor T en vez de 1, que es además recomendable por cuanto la expresión facial no cambia significativamente cuadro a cuadro en una secuencia convencional de 30 fps, el siguiente cuadro $t+T$ depende de t y $t+2T$. En tanto las secciones cruzadas (*cross-section*) de nuestro volumen son constantes, todos los magnitudes de gradientes acumulados en t y $t+T$ pueden ser reutilizadas en un procedimiento incremental mostrado en el algoritmo 2.

Algorithm 2 Pseudoalgoritmo para la extrcción de la textura volumétrica

procedure VPOEM
    Calcular $V(\theta_n, t)$
    Sustraer i_{t,p,θ_n} de $V(\theta_n, t)$
    para obtener la textura volumétrica inicial
    Almacenar los valores i_{t,p,θ_n} y i_{t+T,p,θ_n}
Iterativamente
    Calcular $V(\theta_n, t+T)$
    Almacenar i_{t,p,θ_n} y i_{t+T,p,θ_n}
    $i_{t+T,p,\theta_n} \rightarrow i_{t,p,\theta_n}$
    $i_{t,p,\theta_n} \rightarrow i_{t-T,p,\theta_n}$
    Sustraer i_{t+t_0,c,θ_n} de $V(\theta_n, t+T))$
    para obtener la textura inicial
    $t = t + T$
end

Usando este procedimiento, el cálculo de los códigos VPOEM es apenas ligeramente más costoso que los cálculos de los códigos POEM, con requerimientos mayores de memoria, que no son limitantes para un sistema de cómputo actual, ya que es menor que el equivalente del almacenamiento de dos imágenes monocromáticas de tamaño 128×128. Adicionalmente, en tanto que el procedimiento no se hace sobre todos los cuadros de video de la imagen, no se necesita que el tiempo global de todo el sistema sea menor que 30ms para 30fps. Esto no es inconveniente, por cuanto la duración de la expresión facial típicamente es de varios cuadros (generalmente mayor que 300ms en la base de datos y en los videos compilados por el autor), entonces diezmar la secuencia con el protocolo reseñado no es inconveniente. El algoritmo VPOEM fue probado usando el mismo esquema de clasificación por chi-cuadrado mostrado previamente, dividiendo cada imagen de 128×128 pixeles en 8×8 celdas, con parámetros $P = 5$, número de orientaciones $N = 3$, $T = 5$. El tamaño de la grilla es altamente dependiente del tamaño del rostro detectado: si el número de celdas es relativamente alto, no hay un adecuado número de pixeles para codificar, de manera que el histograma es pobremente construido; si el número de celdas es muy pequeño, la información descrita por los histogramas es muy global y posiblemente no caracteriza bien la expresión facial. El valor de tiempo T óptimo para representación de la expresión ápice fue $T = 7$ según nuestras pruebas preliminares. Sin embargo, al extender el análisis a otros cuadros de las secuencias de video se encontró que es preferible usar una transición temporal ligeramente menor, de $T = 5$. Esto es porque en algunos casos los saltos de $T = 7$ pueden ocasionar que sólo se evalúen cuadros con expresión muy tenue, particularmente cuando el individuo ejecuta la expresión facial muy rápidamente, especialmente con muestras de expresión natural compiladas por el autor. En la sección 3.4 se muestran algunos resultados que fundamentan la elección de estos parámetros.

En general, la transición entre dos cuadros consecutivos en una secuencia capturada

Figura 15. Variación entre cuadros cercanos en una secuencia de expresión facial

a 30 fps es muy pequeña. En la figura 15 se muestra en la parte superior dos cuadros consecutivos y en la parte inferior dos cuadros separados aproximadamente 200ms. Se puede determinar claramente que la variación entre los cuadros superiores es muy pequeña, de manera que la información dinámica representada por su transición no es considerable. En cambio en los cuadros inferiores esta variación es mucho más notable, de manera que la codificación que incluye estas transformaciones dinámicas puede proporcionar mayor información acerca de la expresión facial.

3.3.3. *Temporal Patterns of Oriented Edge Magnitudes* (TPOEM)

Nuestra definición de gradiente acumulado hasta ahora se basa en el gradiente en una dirección espacial. En esta sección incluimos una propuesta novedosa denominada TPOEM que agrega el gradiente acumulado en la dirección temporal, que en principio debería aportar información importante para la caracterización de la expresión facial. En consecuencia, en vez de usar el conjunto de orientaciones espaciales θ_n, $n \in \{0, ..., N-1\}$, incorporamos la dimensión temporal. Denominando θ_z a esta nueva dimensión temporal, la ecuación simplificada está dada por:

$$TPOEM_{T,P,R}(\theta_z, t) = \sum_{p=0}^{P-1} f(i_{t,p,\theta_z} - i_{t,c,\theta_z}) 2^j \tag{3.19}$$

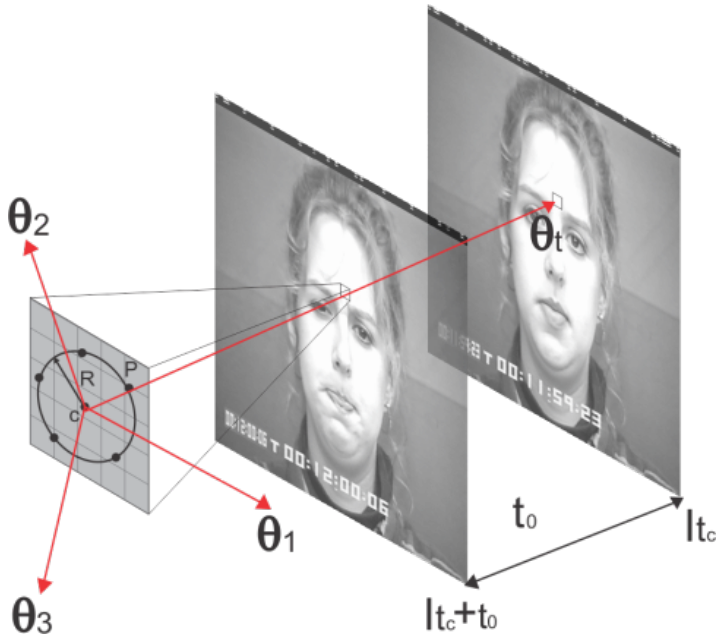
Finalmente, la codificación completa TPOEM es la concatenación del vector VPOEM con las orientaciones espaciales y la codificación parcial TPOEM con la dimensión temporal añadida se muestra en la ecuación 3.20.

$$TPOEM_{T,P,R,\{1,...,N\}} \leftarrow VPOEM_{T,P,\{1,...,N\}} \tag{3.20}$$

$$TPOEM_{T,P,R,N+1} \leftarrow TPOEM_{T,P,R} \tag{3.21}$$

En la figura 16 se muestra con mayor claridad la idea general de la codificación TPOEM. En el caso de la figura, la codificación tiene 3 orientaciones espaciales (θ_1, θ_2 y θ_3) y la orientación temporal θ_t), con espacio entre cuadros vecinos definido por t_0.

Figura 16. Codificación TPOEM



3.4. Resultados

En esta sección consignamos los resultados de clasificación preliminares usando la base de datos CK y posteriormente las pruebas realizadas con la base de datos CK+ en cuanto fue recibida la autorización de uso de esta última, así como resultados que fundamentan la elección de ciertos parámetros.

En primera instancia, el número de vecinos P para la vecindad POEM. Para ello realizamos pruebas con número variable de vecinos en un problema de clasificación de 6 clases y de 7 clases usando clasificación simple chi-cuadrado. Los resultados de una de estas pruebas se muestran en la figura 17. Para la clasificación de 7 clases la forma de la curva es similar. Se puede determinar que el valor pico de clasificación se alcanza con alrededor de 10-11 vecinos espaciales, pero con crecimiento apenas marginal a partir de 4 vecinos. Debido a ello nuestra elección de 5 vecinos espaciales, pues es un buen compromiso que muestra adecuada capacidad de descripción de la expresión facial y reducido tamaño que atenuará posteriormente las dificultades del trabajo en espacios de alta dimensión.

Los resultados de clasificación para 6 expresiones y 7 expresiones (6 expresiones más instancia neutral) para el algoritmo POEM usando la base de datos Cohn-Kanade se muestran en las tablas 2 y 3.

La siguiente prueba fue usando ponderación manual de pesos de las celdas. Los resultados se muestran en 4 y 5.

Las siguientes pruebas fueron realizadas usando el procedimiento mostrado en este capítulo para determinar el peso óptimo de las celdas en la clasificación. Los resultados se muestran en 6 y 7.

Figura 17. Tasa de clasificación contra número de vecinos P

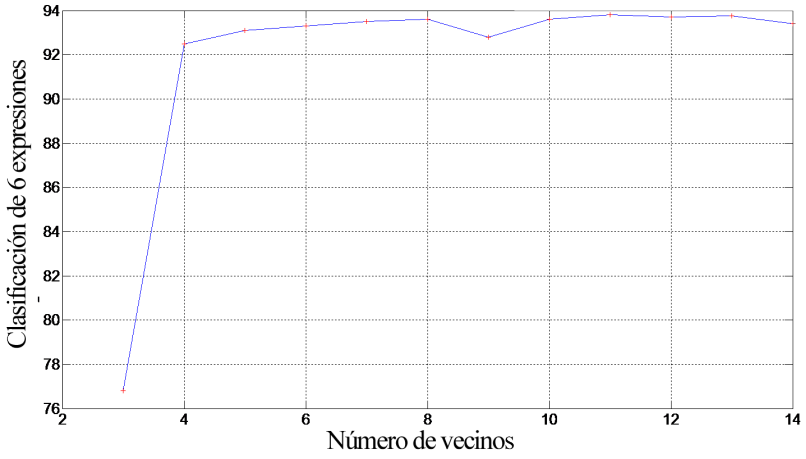


Tabla 2. Reconocimiento de 7 expresiones faciales usando POEM y métrica chi-cuadrado

	Ira	Dis.	Mie.	Ale.	Tri.	Sor.	Neu.
Ira	**70.9**	12.7	0.0	0.0	0.9	0.0	15.5
Disgusto	2.5	**93.3**	0.8	0.0	0.8	0.0	2.5
Miedo	0.6	3.9	**65.6**	11.1	3.3	1.7	13.9
Alegría	0.0	0.0	2.9	**92.9**	0.0	0.0	4.3
Tristeza	3.5	0.0	0.4	1.3	**63.0**	2.2	29.6
Sorpresa	0.0	0.0	0.0	0.0	0.0	**85.6**	14.4
Neutral	2.4	0.9	0.5	0.2	7.5	0.0	**88.0**

Estos resultados preliminares muestran cómo incluso con un sistema de clasificación simple los descriptores POEM son adecuados para representar la expresión facial, así como el uso de ciertas adaptaciones básicas tales como la ponderación por celda permiten incrementar notablemente los resultados obtenidos.

En cuanto los resultados hasta ahora mostrados son obtenidos usando la base de datos Cohn-Kanade, en la tabla 8 mostramos la comparación con resultados encontrados en la revisión bibliográfica usando la misma base de datos o la base de datos CK+ (cuya clasificación es más sencilla, debido a la eliminación de muestras defectuosas), incluyendo los resultados de mayor precisión del estado del arte usando parámetros basados en apariencia.

[3]En este trabajo la clasificación se realizó usando únicamente 4 expresiones: alegría, sorpresa, disgusto y neutral, que son las expresiones de más fácil clasificación en los resultados vistos hasta ahora. Es razonable que esta tasa de acierto sea mucho menor si se incluyen las demás expresiones, cuya

Tabla 3. Reconocimiento de 6 expresiones faciales usando POEM y métrica chi-cuadrado

	Ira	Dis.	Mie.	Ale.	Tri.	Sor.
Ira	**83.9**	15.1	0.0	0.0	1.1	0.0
Disgusto	2.3	**95.6**	0.9	0.0	0.9	0.0
Miedo	0.7	4.5	**76.1**	12.9	3.9	1.9
Alegría	0.0	0.0	3.0	**97.0**	0.0	0.0
Tristeza	4.9	0.0	0.6	1.9	**89.5**	3.1
Sorpresa	0.0	0.0	0.0	0.0	0.0	**100**

Tabla 4. Reconocimiento de 7 expresiones faciales usando POEM y ponderación manual de pesos

	Ira	Dis.	Mie.	Ale.	Tri.	Sor.	Neu.
Ira	**75.5**	10.0	0.0	0.0	1.8	0.0	12.7
Disgusto	1.7	**93.3**	0.8	0.0	0.8	0.0	3.3
Miedo	1.1	2.8	**65.0**	10.6	3.3	3.3	13.9
Alegría	0.0	0.0	2.1	**93.9**	0.0	0.0	3.9
Tristeza	3.5	0.0	0.0	1.3	**63.9**	3.0	28.3
Sorpresa	0.0	0.0	0.0	0.0	0.0	**85.6**	14.4
Neutral	2.4	0.2	0.5	0.5	6.4	0.4	**89.6**

Las tasas de clasificación con nuestro sistema simple de clasificación y parámetros POEM muestran valores adecuados para considerar que son buenos descriptores. Es

clasificación es típicamente mucho más difícil.

[4]En este trabajo la validación se hizo usando *leave-subjects-out*, con un 20 % de las muestras aleatorias. Es decir, no se hizo basado en individuos sino basado en muestras. En nuestra opinión, un sistema de clasificación experto multiclase en la cual haya varias muestras por individuo por clase debe ser validado basado en individio y no en muestra, para evitar entrenamiento negativo. Esto se confirma posteriormente en el capítulo 6, donde hacemos validación *n-folded random* y entrenamiento experto con SVM-RBF y obtuvimos tasa de clasificación de 7 expresiones de 99.14 %. Debido a esto, consideramos que el criterio de validación debe ser cuidadoso para obtener resultados reales y no inflados por aprendizaje negativo o por overfitting. Nuestra hipótesis además es apoyada por el hecho de que estos resultados cercanos al 100 % fueron obtenidos usando la base de datos Cohn-Kanade, no la CK+. En la base de datos Cohn-Kanade muchas muestras no están etiquetadas correctamente o tienen etiqueta incorrecta, por cuanto no tiene sentido que un sistema automático de clasificación tenga clasificación casi perfecta pese a ello.

[5]La clasificación en este trabajo es con SVM, usando validación *10-folded random*, también basado en muestras, como en el trabajo anterior, de manera que el mismo comentario es aplicable.

Tabla 5. Reconocimiento de 6 expresiones faciales usando POEM y ponderación manual de pesos

	Ira	Dis.	Mie.	Ale.	Tri.	Sor.
Ira	**86.5**	11.5	0.0	0.0	2.1	0.0
Disgusto	1.7	**96.6**	0.9	0.0	0.9	0.0
Miedo	1.3	3.2	**75.5**	12.3	3.9	3.9
Alegría	0.0	0.0	2.2	**97.8**	0.0	0.0
Tristeza	4.9	0.0	0.0	1.8	**89.1**	4.2
Sorpresa	0.0	0.0	0.0	0.0	0.0	**100**

Tabla 6. Reconocimiento de 7 expresiones faciales usando POEM y estimación de pesos por celda

	Ira	Dis.	Mie.	Ale.	Tri.	Sor.	Neu.
Ira	**78.2**	8.2	0.0	0.0	0.0	0.0	13.6
Disgusto	1.7	**92.5**	0.8	0.0	0.8	0.0	4.2
Miedo	1.1	1.7	**71.7**	8.9	3.3	1.7	11.7
Alegría	0.0	0.0	1.4	**93.9**	0.0	0.0	4.6
Tristeza	2.6	0.0	0.0	1.3	**70.4**	2.2	23.5
Sorpresa	0.0	0.0	0.0	0.0	0.0	**85.2**	14.8
Neutral	2.0	0.0	0.7	0.5	5.1	0.2	**91.5**

también de resaltar que los resultados con codificación POEM y pesos empíricos son considerablemente superiores a los resultados con codificación LBP, también con pesos empíricos. Esto muestra que los códigos POEM adaptados que usamos para este trabajo son poderosos descriptores de la expresión del rostro. Por último, las tasas de clasificación no son lejanas de la clasificación con LBP-TOP+VLBP, pese a que la longitud de esta codificación es mayor, usa varios cuadros en la estimación dinámica y la clasificación es experta, con má quinas de soporte vectorial, además de nuestras observaciones acerca del uso de *random 10-folded* en oposición a *leave-subjects-out* para este tipo de problema de clasificación.

Una vez estos resultados confirman nuestra idea de la validez de los parámetros usados, hicimos pruebas con los algoritmos TPOEM desarrollados en este capítulo. Para entonces ya teníamos autorización de uso de la base de datos CK+. Si bien repetimos todas las pruebas previas con esta base de datos, por razones de espacio nos limitaremos a consignar los resultados de validación de 7 expresiones con POEM con pesos estimados y validación de 7 expresiones con TPOEM con pesos estimados, con el fin de mostrar

Tabla 7. Reconocimiento de 6 expresiones faciales usando POEM y estimación de pesos por celdas

	Ira	Dis.	Mie.	Ale.	Tri.	Sor.
Ira	**90.5**	9.5	0.0	0.0	0.0	0.0
Disgusto	1.7	**96.5**	0.9	0.0	0.9	0.0
Miedo	1.3	1.9	**81.1**	10.1	3.8	1.9
Alegría	0.0	0.0	1.5	**98.5**	0.0	0.0
Tristeza	3.4	0.0	0.0	1.7	**92.1**	2.8
Sorpresa	0.0	0.0	0.0	0.0	0.0	**100**

Tabla 8. Comparación de resultados de clasificación con metodología similar de clasificación y con trabajos significativos del estado del arte

	7 expresiones	6 expresiones
Parámetros geométricos + TAN[26]	73.2	–
LBP+ template matching [151]	79.1	84.5
POEM, template matching	79.9	90.4
POEM, template matching, pesos empíricos	81.0	91.0
POEM, template matching, pesos estimados	81.2	90.8
TPOEM, template matching, pesos estimados	83.7	92.4
Redes bayesianas, SVM y árboles de decisión [148] [3]	–	91.89 (4 expresiones)
Grillas Candide y PFEG, SVM multiclase [84] [4]	–	99.45
LBP-TOP+VLBP [193] [5]	–	95.19

cómo los nuevos algoritmos de codificación representan una mejora sustancial en la clasificación, incluso usando aún sistema simple de clasificación.

En la tabla 9 se muestran los resultados de la validación de clasificación de 7-expresiones con la base de datos CK+ usando algoritmo POEM convencional.

Por último, se muestran los resultados de la clasificación usando TPOEM, estimación de pesos, con la base de datos CK+, en la tabla 10.

Los parámetros usados para la codificación TPOEM de estos resultados son $P = 5$, $T = 5$, $N = 2$, $R = 5$. Con estos parámetros, los códigos TPOEM tienen una longitud de 40 bits en total, en comparación con los códigos POEM usados de longitud 24, con $P = 8$ y $N = 3$. Sin embargo, el cálculo de los códigos TPOEM en una secuencia de video es en promedio 32 % más corto que los códigos POEM en la misma secuencia, gracias a la reutilización de datos en el cómputo VPOEM. Los resultados muetran clasificación global de 92.4 % en comparación con 91.24 % obtenido con codificación POEM adaptada. Si bien no pareciera una mejora drástica, al observar la matriz de confusión se puede determinar que TPOEM contribuyó de manera notable a aumentar la

Tabla 9. Reconocimiento de 7 expresiones faciales usando POEM y estimación de pesos por celda, base de datos CK+

	Ira	Dis.	Mie.	Ale.	Tri.	Sor.	Neu.
Ira	**92.01**	0.33	0.00	0.00	0.29	0.00	7.29
Disgusto	0.08	**93.55**	1.65	0.58	0.00	0.00	4.14
Miedo	0.00	0.00	**81.43**	1.03	0.25	6.52	10.77
Alegría	0.00	0.00	0.00	**99.99**	0.00	0.00	0.01
Tristeza	0.42	0.00	0.40	0.00	**85.68**	0.00	13.50
Sorpresa	0.00	0.00	0.74	0.00	0.00	**92.57**	6.69
Neutral	3.07	0.56	1.07	0.33	1.98	0.83	**92.16**

Tabla 10. Reconocimiento de 7 expresiones faciales usando TPOEM y estimación de pesos por celda, base de datos CK+

	Ira	Dis.	Mie.	Ale.	Tri.	Sor.	Neu.
Ira	**93.52**	0.00	0.00	0.00	0.00	0.00	6.48
Disgusto	0.00	**91.58**	2.35	2.35	0.00	0.00	3.62
Miedo	0.00	0.00	**89.33**	0.00	0.00	3.33	7.33
Alegría	0.00	0.00	0.00	**100.00**	0.00	0.00	0.00
Tristeza	0.00	0.00	0.00	0.00	**87.00**	0.00	13.00
Sorpresa	0.00	0.00	0.83	0.00	0.00	**94.75**	4.42
Neutral	2.21	1.07	1.43	0.00	3.93	0.71	**90.64**

tasa de acierto de la clasificación de miedo, que es la expresión de más difícil clasificación tanto para un humano como para un sistema automático (ver más en 7), de 81.43 % a 89.33 %. Adicionalmente, con tasas de acierto superiores a 90 % es difícil mejorar notablemente el valor absoluto de la clasificación, pero en términos relativos se redujo el error promedio en un 15.26 %, que es un valor notable, especialmente teniendo en cuenta los clasificadores elementales que se desarrollaron.

Nótese que la introducción de la base de datos CK+ implica una reducción significativa del error, principalmente en la clasificación de ira, miedo y tristeza, sin haber realizado ninguna modificación a la metodología de clasificación. Esto ocurre porque una fracción no despreciable de las muestras de la base de datos original fue eliminada por cuanto no era representativa de la etiqueta correspondiente. Es decir, muchos de los supuestos errores de clasificación con la base de datos Cohn-Kanade no eran errores en realidad y esto, a su vez, sugiere problemas metodológicos en trabajos que mostraron resultados cercanos al 100 % de clasificación con esta base de datos.

3.5. Conclusiones

Los parámetros basados en patrones locales binarios han probado ser eficientes descriptores de textura, de manera que su uso en el reconocimiento de la expresión facial es interesante. En nuestro trabajo usamos adaptación de los parámetros POEM (*Patterns of Oriented Edge Magnitudes*) e introdujimos ideas novedosas que mejoraron la capacidad de representación. Nuestras pruebas preliminares mostraron que el uso de códigos POEM adaptados muestra mejores reultados que el uso de codificación LBP en las mismas o muy similares condiciones: igual resolución facial, igual o muy similar longitud de código y el mismo sistema de clasificación, de manera que se concluyó que la codificación POEM sirve como descriptor adecuado de la expresión facial, con resultados incluso superiores que los de LBP que ya han sido probados como eficientes en esta tarea.

En el cálculo de POEM convencional los códigos fueron reducidos usando codificación uniforme. Sin embargo, nuestras pruebas mostraron mejores resultados al usar codificación no-uniforme tanto con un sistema de clasificación simple, como el mostrado en este capítulo, como con sistemas de clasificación más complejos, tal como está descrito en cite(capítulo). Debido a ello, nuestra adaptación incluyó el uso de codificación no-uniforme y el desarrollo de un mapeo específico para descripción de expresión facial. Para realizar este mapeo se tuvo en cuenta la información de textura por celda espacial, con el objetivo de intentar no codificar texturas representativas de distintas expresiones faciales con el mismo código POEM.

Una característica importante de la codificación no-uniforme previa al mapeo es que la longitud de los códigos crece linealmente con el número de orientaciones y de vecinos usados en el cálculo. Como tal, una desventaja de esta codificación es que no es posible usar un elevado número de vecinos y de orientaciones espaciales sin reducir la dimensión de los datos y, de hecho, a partir de ciertos valores de parámetros el tamaño de los datos se incrementa. Sin embargo, si bien esto puede ser un inconveniente al diseñar un sistema de clasificación debido a la alta dimensionalidad de los datos, en este capítulo se mostró que incluso con relativa alta dimensión de los datos y una métrica de clasificación simple por distancias chi-cuadrado se pueden obtener resultados adecuados. Esto último es muy importante, por cuanto la codificación POEM y, en general, cualquier codificación basada en LBP, tiene la particularidad de que no necesariamente en la comparación de dos códigos distintos con un código prototipo la distancia chi-cuadrado es menor para el código cuya textura es aparentemente más parecida a la textura prototipo, sobre todo luego de usar un mapeo uniforme. Es por esto que los resultados obtenidos con estas pruebas preliminares son alentadores acerca de la potencialidad de los patrones POEM y TPOEM en la resolución del problema.

En este trabajo desarrollamos la implementación de los códigos VPOEM basados en la representación volumétrica de una textura dinámica. VPOEM sirve para codificar la información volumétrica de textura alrededor de un pixel central usando vecinos temporales y espaciales. Si bien la idea más lógica para representar una vecindad en estas dimensiones es el uso de una esfera alrededor del pixel central, en nuestro trabajo

probamos y descartamos esta alternativa. El uso de una vecindad esférica implica que la vecindad para el pixel p_{x,y,t_c} en general no comparte vecinos, salvo singularidades, con la vecindad del pixel p_{x,y,t_c+t_0}. Debido a esto, la vecindad debe ser calculada en cada pixel para todos los cuadros usados de la secuencia, lo cual es muy demandante en términos de tiempos de cálculo. En consecuencia, usamos una vecindad cilíndrica tal que los datos de gradientes acumulados usados en el cómputo de VPOEM para un cuadro específico puedan ser reutilizados para cálculos de VPOEM en la misma localización espacial de vecinos pasados y futuros. Gracias a ello se logró una reducción drástica del costo de procesamiento del sistema.

La codificación VPOEM desarrollada fue optimizada tal que el costo de cálculo es inferior que el costo de cálculo de POEM para una secuencia de video dada, con la ventaja de que además VPOEM incluye información temporal. La optimización fue conseguida no sólo al usar la vecindad cilíndrica reseñada previamente, sino al almacenar los gradientes acumulados sin el *offset* del valor del pixel central, tal como se hace en LBP o POEM convencional. De esta forma, el cálculo iterativo de VPOEM por pixel en la secuencia simplemente determina la vecindad para el pixel particular, resta el umbral dado por el gradiente acumulado en el pixel y ejecuta la codificación, mientras que el mismo proceso para POEM o LBP cuadro a cuadro requiere del cálculo completo de todo en cada cuadro de la secuencia.

Posteriormente introdujimos la idea novedosa de TPOEM. TPOEM añade el gradiente en dirección temporal, en vez del gradiente acumulado espacial calculado por VPOEM. En tanto que una secuencia de video de expresión facial muestra una variación dinámica de texturas en el rostro, se esperaba que el gradiente temporal acumulado pudiese evaluar estas transiciones. El código TPOEM final es la concatenación del código previo VPOEM y el cálculo con la dimensión temporal añadida. Debido a que en una secuencia de video normal de 30 fps la variación entre cuadro y cuadro es bastante pequeña, hicimos pruebas con distinta longitud del parámetro T de vecindad TPOEM. Nuestros resultados mostraron que el valor con mejor clasificación de expresiones ápice es $T = 7$, pero al incluir cuadros con expresiones no ápice (transición entre instancia neutral y expresión ápice o expresión en decaimiento) observamos que los resultados globales eran mejores con $T = 5$, especialmente porque la duración de la expresión facial puede hacer que el uso de un valor elevado de T ocasione que no se evalúe ningún cuadro con expresión fuertemente marcada.

En nuestras pruebas usamos un sistema de clasificación simple por distancia chi-cuadrado entre las muestras de validación y las muestras de entrenamiento. Este sistema de clasificación es bastante limitado, pero esto fue deliberado, por cuanto el objetivo de esta etapa era determinar la validez de los parámetros obtenidos, lo cual puede ser ofuscado si el sistema de clasificación es muy poderoso. Tuvimos particular cuidado en usar validación por *leave-subjects-out* en vez de *random n-folded* convencional. La idea es que el sistema de clasificación no conozca ninguna muestra del mismo individuo (persona), así sea de otra clase (expresión), por cuanto esto puede conducir a entrenamiento negativo. Si bien la clasificación por chi-cuadrado no es propensa a entrenamiento negativo, ya que las expresiones prototipo son promedios y, en consecuencia, no son muy

afectadas por *outliners*, seguimos este criterio de validación durante todo el trabajo. En el capítulo 6 se mostrará cómo usar *random n-folded* con sistemas de clasificación experto condujo a resultados de clasificación cercanos al 100 %, pero, por supuesto, sin gran validez metodológica.

El objetivo de este capítulo de desarrollar e implementar parámetros eficientes descriptores de la expresión facial basados en LBP fue conseguido y los resultados de esta etapa del sistema fueron usados exitosamente en las etapas sucesivas de este trabajo, tal como se mostrará en los siguientes capítulos.

4. Análisis de Datos y Reducción de Dimensiones

4.1. Introducción

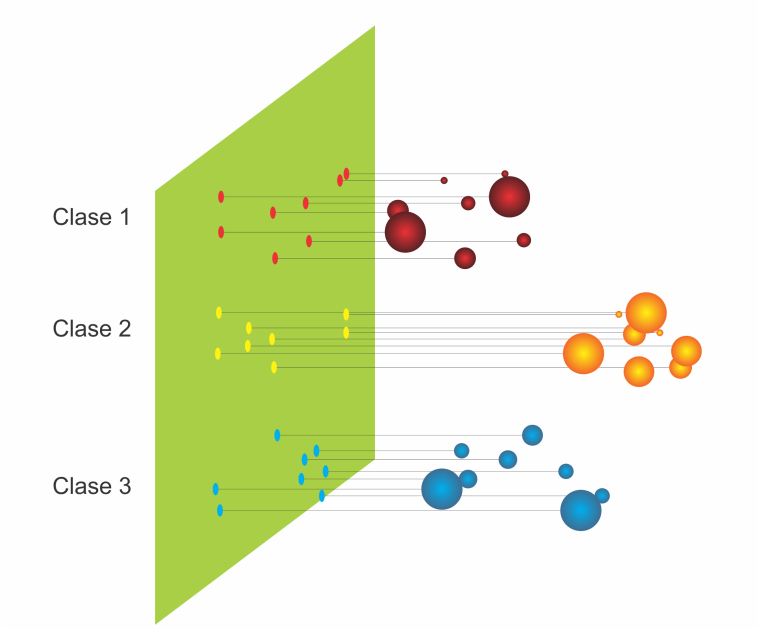
Los datos extraídos en el capítulo 3 son adecuados como descriptores de la expresión facial incluso usando metodologías de clasificación bastante básicas. Una idea normalmente usada consiste en utilizar los datos completos en un sistema de clasificación, considerando que los datos que puedan ser ruidosos o redundantes en todo caso no deteriorarían considerablemente los resultados de clasificación. Por ejemplo, se espera que si existe un conjunto de N variables de las cuales un subconjunto M no es relevante para la clasificación, el sistema ideal de clasificación descarte o atenúe estas variables, sea al eliminarlas por completo como criterio de clasificación o al asignarles una ponderación diminuta (por ejemplo pesos pequeños en una red neuronal o en un sistema deep learning. Esto, sin embargo, no necesariamente es cierto. En [58] se muestra cómo introducir datos redundantes puede ayudar en un proceso de clasificación al atenuar el efecto de variables ruidosas, pero en [96] se demostró cómo incluir variables ruidosas en un entorno de clasificación débil no sólo puede perjudicar la clasificación en términos de clasificación reducida, sino que puede incluso afectar el desempeño en grado tal que se obtengan resultados similares a los obtenidos por clasificación aleatoria.

Por otra parte, el uso del conjunto de datos completos sin procesar tiene otras desventajas. En primer lugar, si existen variables inútiles en la descripción del problema, es recomendable no capturarlas en primera instancia, para reducir los costos globales de procesamiento (aunque este procesamiento no se aborda ahora, sino en el capítulo 5). Adicionalmente, reducir las dimensiones de los datos puede implicar que los algoritmos de clasificación sean más simples y rápidos, que son beneficios añadidos al desempeño global. Por último, aunque en nuestro caso esto no es una limitación, se pueden reducir los requerimientos de memoria, que en algunas aplicaciones es una restricción importante, por ejemplo cuando se usan métricas sísmicas en procesos de análisis de suelos.

En la figura 18 se muestra un ejemplo de reducción supervisada de dimensiones. Los datos están embebidos en un espacio en tres dimensiones. Sin embargo, con una proyección adecuada, se pueden reducir a datos de dos dimensiones, tal como en la figura, o incluso a una dimensión, sin perder capacidad de discriminación entre las clases, de manera que los algoritmos de clasificación tengan menor complejidad y estén menos propensos a sobre ajuste.

Lamentablemente, la reducción de dimensiones implica una suerte de compromisos importantes. Teniendo en cuenta que la reducción de dimensiones es básicamente la proyección de los datos de dimensión p a una dimensión q tal que $q < p$, se espera que los datos en esta dimensión reducida preserven la información relevante para la descripción del fenómeno a caracterizar. Es decir, una reducción de dimensiones es inútil si elimina información necesaria para la discriminación entre clases. Más aún, es importante no sólo que la información se preserve, sino que además la forma del nuevo manifold sea conveniente para que la clasificación no sea problemática (de hecho, en

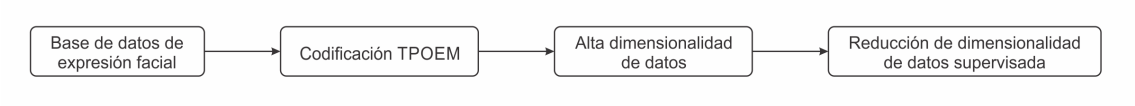
Figura 18. Proyección de datos a un espacio de menor dimensión



algunos sistemas de clasificación no se reduce la dimensión, sino que se aumenta, con el fin de proyectar los datos a un nuevo espacio dimensional en el cual la clasificación sea más simple). Por último, el proceso de reducción supervisada de dimensiones puede conducir a errores metodológicos inadvertidos. Esto ocurre cuando al implementar los algoritmos de reducción supervisada de dimensiones se usan datos que habrán de ser utilizados también en el proceso de validación del sistema. La reducción supervisada de dimensiones puede considerarse como una etapa previa de clasificación, por cuanto tiene en cuenta la etiqueta de las muestras, lo cual implica que usar muestras tanto en reducción supervisada de dimensiones como en clasificación conduce a resultados de clasificación artificialmente elevados. No obstante, esto representa otra limitación importante y en muchos casos crítica para la reducción de dimensiones supervisada: por una parte se requiere de un adecuado número de datos para describir el *manifold* n-dimensional con el fin de hacer reducción de dimensiones, pero por otra parte se requiere que estos datos no sean usados en la etapa de validación, que es un compromiso muchas veces insuperable y consiste en un reto sustancial en el procesamiento de los datos.

Otra dificultad inherente en los problemas con datos de alta dimensión es que los datos en estos espacios tienen propiedades de agrupamiento y distancias distintas que las encontradas en los espacios de pequeña dimensión. Este problema es claramente explicado en [13]. En general, los volúmenes en espacios de alta dimensión son muy grandes, de manera que los datos tienden a aglomerarse en las esquinas de los espacios de alta dimensión, dejando las regiones centrales despobladas. Esto hace que la mayor parte de métricas convencionales usadas en espacios de dimensiones relativamente pequeñas, tales como distancias Euclidianas, sean frecuentemente inútiles en espacios de

Figura 19. Reducción de dimensiones en un proceso de clasificación



alta dimensión, de modo que distancias Chi-cuadrado o Mahalanobis son más frecuentes (aunque las distancias Mahalanobis tienen otra suerte de problemas con conjuntos de datos limitados). Debido a esto mismo, manipular datos de alta dimensión para reducirlos a otra dimensión menor, pero que siga siendo relativamente grande, es un problema importante, por cuanto es factible que los datos en la nueva dimensión estén agrupados de modo que las clases sean equívocas y la clasificación sea imposible. Esto es frecuente en el uso de técnicas no supervisadas de reducción de dimensiones.

En este capítulo veremos la fundamentación teórica de la reducción de dimensiones en la sección 4.2, la metodología de este trabajo en la sección 4.3, incluyendo nuestra propuesta de estimación de dimensión intrínseca en 4.3.1 y pruebas con datos extraídos con reducción no supervisada de dimensiones en 4.3.2 y reducción supervisada de dimensiones en 4.3.3. Finalmente, las conclusiones del trabajo están consignadas en 4.4.

La pertinencia del proceso de reducción de dimensiones en el entorno del sistema completo es mostrada en la figura 19

4.2. Fundamentación teórica

Las imágenes y las secuencias de video contienen una gran cantidad de información. Por ejemplo, una imagen de resolución media de 640×480 pixeles tiene 307.200 bytes para incluir el universo de posibles $256^{307,200}$ imágenes en este espacio. Un algoritmo de reconocimiento de la expresión facial no puede lidiar con datos de este tamaño sin enfrentarse a diversos problemas, tales como los costos de cálculo, overfitting, costo de memoria, reducción de precisión con alta dimensionalidad o la maldición de la dimensionalidad [9] [40]. Una etapa preliminar de localización y extracción de la región facial reduce considerablemente el tamaño de los datos, por ejemplo en nuestro caso con tamaño normalizado de 128×128, pero el tamaño de los datos y los parámetros pueden ser aún elevados, de modo que es considerable el uso de reducción de dimensiones.

4.2.1. Reducción no supervisada de dimensiones

Los principales usos de la reducción de dimensiones son mejorar la efectividad de los métodos de aprendizaje de máquina, comprimir adecuadamente los datos, reducir o eliminar ruido estadístico, obtener o aproximar el valor de la dimensión intrínseca de los datos y en algunos casos tener la capacidad de visualizar en espacios 2D o 3D datos cuya dimensión original es mayor. En esta sección mostraremos algunos métodos típicamente usados en la reducción de dimensiones.

El esquema de la reducción de dimensiones dado un conjunto de datos d en el

espacio p con datos individuales $d_1, d_2, ..., d_N$ consiste en representar los datos en un nuevo espacio de menor dimensión q 4.1:

$$d_i \in \Re^p \rightarrow d_i' \in \Re^q \tag{4.1}$$

$$q < p \tag{4.2}$$

La dimensión intrínseca se refiere a un espacio tal que los datos en d puedan ser embebidos en un manifold con dimensión q tal que si q es la dimensión intrínseca de los datos originales, la reducción de dimensiones es realizada óptimamente. El objetivo de una reducción de dimensiones exitosa para un sistema de clasificación es reducir adecuadamente las dimensiones de los datos sin perder información esencial y al mismo tiempo mantener o reducir las distancias intraclase mientras las distancias interclase se mantienen o aumentan. Para conseguir esto último, generalmente se requiere de un proceso de reducción de dimensiones supervisado, de modo que la información de clase es incluida en el proceso.

Debe resaltarse que la dimensión intrínseca, si existe, y la geometría del *manifold* de datos no son conocidas, de modo que el problema de reducción de dimensiones sólo puede ser resuelto si se asumen ciertas propiedades de los datos [165].

Reducción de dimensiones lineal: El análisis de componentes principales (PCA) es de lejos el método más usado para realizar reducción lineal de dimensiones [125] [79]. El objetivo de PCA es reemplazar los datos de entrada con datos transformados tal que la correlación lineal entre las variables es minimizada. Consecuentemente, la reducción de dimensiones es más drástica si existe alta correlación lineal entre las variables iniciales. La reducción es obtenida al encontrar la transformación lineal que mantiene la mayor varianza posible. La más alta varianza está representada en el primer componente principal, la segunda más alta en el segundo componente y así sucesivamente, de manera que los últimos componentes pueden ser eliminados debido a su pequeña contribución. Se asume que debido a que su contribución a la varianza global es reducida, no tienen información importante [1].

En PCA un mapeo M es obtenido tal que $M^T cov(X)M$ es maximizado, donde X son los datos de entrada y $cov(X)$ es la covarianca de la matriz X, restringida a $|M| = 1$. Al usar multiplicadores de Lagrange, la minimización del problema no restringido $M^T cov(X)M + \lambda(1 - M^T M)$ se obtiene cuando $cov(X)M = \lambda M)$. Una vez

[1] No obstante, una contribución reducida a la varianza global no necesariamente indica que no hay información relevante. Por ejemplo, una reducción de dimensiones por PCA en imágenes faciales usando eigenfaces permite caracterizar rostros en una dimensión notablemente menor que la dimensión de las imágenes originales, con capacidad de reconocimiento facial debido a que las variaciones globales son producidas por variaciones interpersonales. Sin embargo, las variaciones locales, de mucha menor contribución energética -ceño, texturas en el mentón y alrededor de los ojos, por ejemplo-, contienen información requerida para la discriminación entre expresiones faciales, de manera que una aproximación por reducción de dimensiones usando PCA simple conduce a problemas de pérdida de información de expresión.

λ es obtenido, los principales a eigenvalores son usados, de modo que la reducción de dimensiones es tal como se muestra en la ecuación 4.3 :

$$Y = (X - \bar{X})M \tag{4.3}$$

PCA tiene ciertos inconvenientes, sin embargo. Si la dimensión de los datos de entrada es muy grande, el cálculo de los eigenvectores puede ser muy costoso y con gran requerimiento de memoria. Adicionalmente, PCA es un buen método de reducción de dimensiones si hay una correlación lineal entre las variables de los datos. Si existe correlación pero de una naturaleza no lineal, PCA no puede lograr el objetivo incluso si hay una dimensión intrínseca teórica más pequeña. Finalmente, PCA garantiza que los mayores valores de varianza son mantenidos, pero es posible que parte de la información eliminada sea crucial en el problema de clasificación, principalmente porque una aproximación por eigenfaces, cuyos resultados de discriminación entre individuos son notables, atenúa o elimina componentes de variación local importantes para la discriminación entre expresiones.

PCA intenta minimizar el error de reconstrucción al escoger los componentes principales más importantes. En algunas aplicaciones el error de reconstrucción no es crucial, sino la maximización de la independencia entre las direcciones de proyección. En análisis independiente de componentes (ICA) [28] este objetivo es alcanzado al minimizar la información mutua entre los datos 4.4

$$I(Y) = \left[\sum_{j=1}^{J} H(Y_i)\right] - H(Y) \tag{4.4}$$

$$\begin{vmatrix} y_1(t) \\ \vdots \\ y_m(t) \end{vmatrix} = W \begin{vmatrix} x_1(t) \\ \vdots \\ x_n(t) \end{vmatrix} \tag{4.5}$$

Otra alternativa es la minimización de la gaussianidad de la transformación de datos $y = Wx$ o la creación de una función de costo para calcular W e iterativamente minimizar la información mutua y la gaussianidad de la transformación.

ICA es importante en aplicaciones en las cuales la separación de las fuentes en los datos de entrada, por ejemplo cuando se requiere separar el ruido del resto de información o cuando hay más de una fuente de información presente en los datos [74].

Reducción no lineal de dimensiones: La reducción no lineal de dimensiones es un problema estudiado más recientemente, de manera que aún hay investigación importante en en desarrollo y no hay consenso completo sobre los métodos más apropiados. Adicionalmente, la reducción de dimensiones puede ser requerida para objetivos diversos. Por ejemplo, algunos métodos intentan mantener las propiedades locales de los datos en el nuevo espacio de menor dimensión, mientras que otras técnicas intentan mantener las propiedades sobre todo el conjunto de datos. A continuación haremos una breve reseña sobre estas técnicas.

Isomap: Isomap asume el hecho de que los puntos pueden estar sobre o cerca un *manifold* de cierta forma. El ejemplo más típico es el rollo suizo, pero en general puede ser cualquier *manifold* n-dimensional. La teoría subyacente implica que puede haber dos o más puntos que no están muy cercanos en el espacio de alta dimensión pero que podrían ser cercanos en el nuevo *manifold* construido, de manera que en realidad son vecinos. Al mismo tiempo dos puntos pueden estar cerca en el espacio de alta dimensión, pero la distancia geodésica sobre el nuevo *manifold* es alta, así que no son considerados vecinos.

Para construir el *manifold*, para los datos en d_i, para cada punto de los datos se encuentran sus K vecinos más cercanos en el conjunto completo D y se construye un grafo de distancia entre los puntos. Ésta es la principal diferencia entre *Multidimensional Scaling* (MDS) e Isomap [157]. Una vez el grafo es calculado, una aproximación de la distancia entre los puntos y el *manifold* supuesto es obtenida usando un algoritmo de búsqueda de grafos tal como el algoritmo Dijkstra [37]. Los más importantes n eigenvectores de la matrix de distancias geodésicas son usadas para construir el nuevo espacio dimensional.

Isomap ha sido útil en un gran número de aplicaciones. No obstante, tiene algunos inconvenientes. Es esencial que el número de puntos en el conjunto de datos D sea suficientemente adecuado para describir el *manifold*. Para problemas de alta dimensionalidad, la maldición de la dimensionalidad puede ser suficientemente problemática para impedir por completo el uso de Isomap. Por otra parte, los eigenvectores en la matrix de distancias geodésicas son muy sensibles al ruido, de manera que cierta dispersión alrededor del *manifold* puede conducir a una proyección muy diferente. Debido a esto, el valor de K (número de vecinos) debe ser elegido con cuidado. Un alto valor de K hace que el método sea muy propenso a errores de corto circuito, pero valores muy pequeños de K pueden hacer que la silueta del *manifold* no sea adecuadamente definida. Otro problema es la presencia de agujeros en el *manifold*, producidos porque no hay puntos cercanos en cierta región para definir adecuadamente su forma. Esto es un problema muy importante en espacios de muy alta dimensionalidad, pues definir las formas de estas altas dimensiones requiere de un conjunto exponencialmente grande de datos. Por último, Isomap tiene inconvenientes en la definición de *manifolds* complejos no convexos.

Mapas de difusión: Los mapas de difusión están basado en los sistemas dinámicos de transferencia de calor y caminos aleatorios en una cadena de Markov [27] [149]. El objetivo es que al hacer un camino aleatorio hay una métrica que proporciona información acerca de la vecindad de un punto en el conjunto de datos. Es asumido que el conjunto de datos está en un espacio de alta dimensión pero puede ser embebido en un espacio de menor dimensión, donde las distancias de difusión se mantienen tan cercanas como sea posible a sus valores originales.

Un kernel K es construido tal que define una afinidad de difusión en el grafo formado por los puntos en el conjunto de datos. Una posibilidad general es usar un kernel Gaussiano, tal como se muestra en la ecuación 4.6

$$k_{i,j} = \begin{cases} \frac{e^{-\left\|d_i - d_j\right\|}}{\sigma^2}, \text{if } x_i \approx x_j \\ 0, \qquad\quad \text{otherwise} \end{cases} \tag{4.6}$$

$d_i \approx d_j$ significa que d_i es un vecino cercano de d_j; de otra forma el grafo sería la matrix de grafos de distancias euclidianas del conjunto completo. En realidad, se requerirían las distancias geodésicas, pero el *manifold* no es conocido, de manera que se usan distancias euclidianas como una estimación de las distancias geodésicas locales. Tal como se observa en la ecuación, si la distancia entre un par de puntos d_i y d_j es grande, el valor $k_{i,j}$ correspondiente es pequeño, idealmente cero, mientras que si la distancia entre el par de puntos es pequeña el valor de $k_{i,j}$ tiende a uno. La idea es modular σ de manera que se obtengan aproximadamente estos dos valores de acuerdo con las distancias. La matrix K aún no es una cadena de Markov. Para obtenerla, se normaliza la matrix K así:

$$p_{i,j} = \frac{w_{i,j}}{\sum_{k=1}^{N} w_{i,k}} \tag{4.7}$$

El valor en $p_{i,j}$ representa la probabilidad de transición entre x_i y x_j en un simple paso. La matrix P representa, entonces, información geométrica del conjunto de datos D y, en oposición a PCA, la información de puntos distantes d en el conjunto de datos no es considerada. Como tal, puede verse que la probabilidad de transición entre un par de puntos está dada por $P^{(t)} = (P)^t$ y, de acuerdo con esto, la distancia de difusión está dada por 4.8:

$$D^{(t)}(d_i, d_j) = \sqrt{\sum_{k=1}^{N} \frac{\left(p_{i,k}^{(t)} - p_{i,k}^{(t)}\right)^2}{\psi(d_k)^{(0)}}} \tag{4.8}$$

$\psi(d_k)^0 = \frac{m_i}{\sum_j m_j}$, donde $m_i = \sum_j p_{i,j}$. Se puede observar que $\psi(d_k)^0$ es mayor cuando hay una mayor densidad de puntos alrededor de d_k, mientras que si la región es relativamente poco densa, el valor de $\psi(x_k)^0$ disminuye.

Finalmente, para transformar los datos en un espacio de menor dimensión, se resuelve el eigenproblema $P^{(t)}v = \lambda v$ para los primeros a eigenvalores excluyendo λ_1 y la representación se hace por reconstrucción por eigenvalores convencional.

***Local Linear Embedding*:** Pese a su nombre, LLE es una técnica no lineal. LLE es lineal únicamente en regiones locales. La idea de LLE es preservar las relaciones lineales entre vecinos cercanos, de manera que la linealidad es sólo asumida en regiones locales [137]. De esta manera, *manifolds* complejos, incluso no convexos, pueden ser proyectados en espacios de menor dimensión.

En LLE la descripción de la geometría local es realizada al aplicar coeficientes lineales para reconstruir cada punto a partir de sus vecinos cercanos. En la técnica tradicional LLE, se usan los K vecinos más cercanos, pero algunas mejoras pueden ser agregadas,

tal como usar una vecindad local de determinado tamaño o dependiente de una métrica local.

La transformación lineal para obtener el punto d_i a partir del punto d_j es denominada $w_{i,j}$. Los pesos en W son calculados de modo que se minimice una función de costo con dos restricciones: las reconstrucciones de d_i dependen únicamente de los vecinos cercanos, de manera que $w_{i,j} \neq 0$ si $d_i \approx d_j$, pero si los puntos no son vecinos cercanos, $w_{i,j)=0}$. La segunda condición es que la suma de W en una fila es uno: $\sum_j wi, j = 1$.

Los pesos obtenidos son invariantes a rotaciones, traslaciones y cambios de escala para cada punto y sus vecinos. Así, W caracteriza las propiedades geométricas locales de los datos D. Consecuentemente, las propiedades geométricas locales son preservadas en el nuevo espacio de menor dimensión.

En la siguiente etapa de LLE, los datos en D son proyectados a Y, con una menor dimensión. El tamaño de la nueva dimensión es escogido de manera que se minimice la función de costo dada por la ecuación 4.9:

$$\Phi(Y) = \sum_i \left| \overline{y_i} - \sum_j w_{i,j} \overline{y_j} \right|^2 \tag{4.9}$$

Esta función de costo es similar a la usada para obtener W, pero ahora W es una constante y la dimensión del nuevo conjunto de datos Y es optimizada. La solución de este problema es dada al minimizar la forma cuadrática por sus eigenvectores no nulos.

4.2.2. Reducción supervisada de dimensiones:

Las técnicas de reducción de dimensiones mostradas previamente, tanto lineales como no lineales, son no supervisadas. Esto quiere decir que las metodologías no tienen un conocimiento *a priori* de las etiquetas o clases de cada muestra o, si esta información está disponible, no es tenida en consideración durante la etapa de reducción de dimensiones. Las técnicas de reducción no supervisada de dimensiones pueden tener ciertos problemas. Por ejemplo, el conjunto de datos puede ser embebido a un espacio de menor dimensión, pero las proyecciones pueden carecer de información importante para separar las diferentes clases, e incluso si eso no pasa el nuevo *manifold* puede ser geométricamente más complejo, lo que dificulta la posterior clasificación. Por otra parte, es posible que las métricas para medir las distancias entre los puntos sean alteradas en el nuevo conjunto de datos, de manera que muestras fácilmente distinguibles en el conjunto inicial se conviertan vecinas cercanas en el nuevo conjunto de datos, de modo que métricas convencionales tales como distancias euclidianas o Mahalanobis no funcionen [137]. De esta forma, hay otra aproximación para la reducción de dimensiones que incluye las etiquetas de cada muestra. A continuación haremos una reseña de algunas de estas técnicas.

Análisis discriminante lineal supervisado: El LDA supervisado es una técnica básica de reducción de dimensiones para separar clases. La separación es entre dos

clases, pero puede ser generalizada para múltiples clases usando diferentes técnicas, tales como 1 vs. 1, 1 vs. todos o máquinas expertas [147]. La idea del LDA supervisado es proyectar los datos en un espacio de dimensión $N-1$, donde N es el número de clases, y la separación entre clases está dada de acuerdo a un umbral, generalmente cero, tal como se muestra en 4.10.

$$\begin{aligned} & X \to Y \\ & y = c'x \\ & if\ y_i \geq T,\ \ x_i \in class\ a \\ & if\ y_i < T,\ x_i \in class\ b \end{aligned} \tag{4.10}$$

donde c es la proyección al nuevo espacio dimensional.

De acuerdo con esto, el objetivo del LDA supervisado es encontrar la proyección c que maximiza la distancia interclases entre los datos de la clase a y la clase b. Normalmente la línea de proyección es calculada tal que se minimiza una función de error, pero esta línea puede ser modificada adicionalmente de acuerdo con una función de castigo, por ejemplo cuando falsos positivos sean menos severos que falsos negativos en sistemas de diagnóstico.

Maximum Collapsing Metric Learning **(MCML):** La idea fundamental en muchos métodos de reducción supervisada de dimensiones es que alguna métrica, tal como la distancia Mahalanobis, sea preservada luego de que la transformación $X \to Y$ sea realizada. Como tal, si muestras pertenecientes a la misma clase están agrupadas de manera cercana en el espacio $X \in \Re^p$, se espera que las muestras permanezcan abrupadas en el espacio $Y \in \Re^q$, donde $p > q$. MCML tiene otro objetivo ideal para maximizar la discriminación entre las clases [54]. Si todos los puntos que pertenecen a una clases pudiesen ser transformados a un solo punto en el nuevo espacio dimensional, el problema de clasificación sería trivial, pues cada clase estaría representada por un solo punto en el nuevo conjunto de datos.

En MCML, dado un conjunto de datos X representado por (x_i, c_i), donde las muestras son x_i, $x_i \in \Re^p$, y c_i son las etiquetas, $c_i \in \{1, 2, ..., k\}$, una métrica de similitud entre las muestras es obtenida. La métrica generalmente usada es la distancia Mahalanobis, como en la ecuación 4.11:

$$d\left(x_i, x_j|\ A\right) = d_{ij}^A = \left(x_i - x_j\right)^T A \left(x_i - x_j\right) \tag{4.11}$$

El escenario ideal es considerar que cada punto en X que pertenezca a una clase específica debe ser proyectado al mismo punto en Y, de modo que la distancia Mahalanobis es cero, mientras que cada punto perteneciente a una distinta clase debe ser proyectado a una distancia infinitamente lejana. La aproximación de MCML es usar, por punto, una distribución sobre los otros puntos (es decir, no se incluye el punto de entrenamiento) 4.12

$$p^A(j|i) = \frac{1}{Z_i} e^{-d_{ij}^A} = \frac{e^{-d_{ij}^A}}{\sum_{k \neq i} e^{-d_{ij}^A}}, \quad i \neq j \tag{4.12}$$

En la proyección ideal, $e^{-d_{ij}^A} = 1$ si los puntos x_i y x_j tienen la misma etiqueta y $e^{-d_{ij}^A} = 0$ si los puntos pertenecen a una clase distinta. Como consecuencia, la distribución ideal se convierte:

$$p_0(j|i) = \begin{cases} 1 & c_i = c_j \\ 0 & c_i \neq c_j \end{cases} \tag{4.13}$$

En la siguiente etapa de MCML el objetivo es encontrar la transformación A tal que $\forall(i,j) p^A(j|i) = p_0(j|i))$. Cuando esta transformación es obtenida, se garantiza que todos los puntos de X que pertenecen a la misma clase son proyectados al mismo punto en Y. El problema de minimización de distribuciones se soluciona minimizando la divergencia Kullback-Leibler [86] en 4.14:

$$\min_A \sum_i KL\left[p_0(j|i)|p^A(j|i)\right], \qquad s.t.\ A \in PSD \tag{4.14}$$

4.3. Metodología

Nuestros vectores de parámetros TPOEM fueron ordenados en una matriz con 192 vectores correspondientes a 3×64 vectores de orientación de gradiente espacial y 64 vectores de orientación de gradiente temporal [2]. Con el fin de disponer de un adecuado número de muestras para realizar la estimación de la dimensión intrínseca de los datos y la reducción de dimensiones, se usaron muestras de las bases de datos CK, CK+, JAFFE [101], Yale [52], KDEF [99] e imágenes recopiladas por el autor.

Los vectores de parámetros fueron almacenados en la matriz X así 4.15:

$$X_i = \begin{bmatrix} x_{1,1_i} & x_{1,2_i} & \cdots & x_{1,k_i} \\ x_{2,1_i} & x_{2,2_i} & & \\ \vdots & & \ddots & \\ x_{j,1_i} & & & x_{j,k_i} \end{bmatrix} \tag{4.15}$$

donde X_i es la matriz de parámetros de la muestra i (cuya clase es c_i), j es el vector TPOEM y k es el elemento k dentro del vector TPOEM.

Para efectos de análisis de datos y clasificación preliminar, asumimos que cada fila en X puede ser tratada como un metaclasificador individual. Esto fue realizado de esta

[2] En el capítulo 6 las pruebas iniciales mostraron que es posible usar 2 orientaciones espaciales sin perjudicar los resultados de manera distinguible, pero el análisis de los datos en este capítulo fue realizado inicialmente con los datos originales y luego, para la reducción final de datos, recalculado con 2 orientaciones espaciales

manera para utilizar las ventajas de los clasificadores débiles en la construcción de un clasificador fuerte.

Por simplicidad, y teniendo en cuenta que en este trabajo no se asume cada bit TPOEM individualmente sino como vector conjunto dentro de la celda espacial correspondiente, de ahora en adelante nuestra nomenclatura usada será así: $X_{i,k}$ corresponde al vector TPOEM completo para la muestra i y la celda espacial k.

Una dificultad fundamental en esta parte del trabajo es que el número de muestras usado de aproximadamente 11.000 imágenes es en todo caso limitado para caracterizar un manifold de alta dimensionalidad, de manera que la maldición de la dimensionalidad o pobre generalización son problemas difícilmente evadidos. Esto es un problema general de cualquier sistema de clasificación que se base en un conjunto numeroso de parámetros y en particular en sistemas de clasificación de expresión facial que dependan de un considerable número de vectores de longitud elevada, tales como los sistemas basados en LBP, por cuanto el número de muestras es limitado y posiblemente insuficiente para caracterizar adecuadamente el espacio n-dimensional.

Para explicar este problema, asumiremos 1000 muestras uniformemente distribuidas sobre un hipercubo de dimensión 1. La distancia mínima entre las muestras es, entonces, 1^{-3}. En nuestro caso tenemos datos TPOEM de dimensión 34, de manera que para alcanzar la misma distancia mínima entre muestras es 10^{102} muestras. En general, cada dimensión añadida incrementa el volumen del nuevo hiperespacio de tal manera que la alta dimensionalidad puede ser un problema no solucionable en términos prácticos.

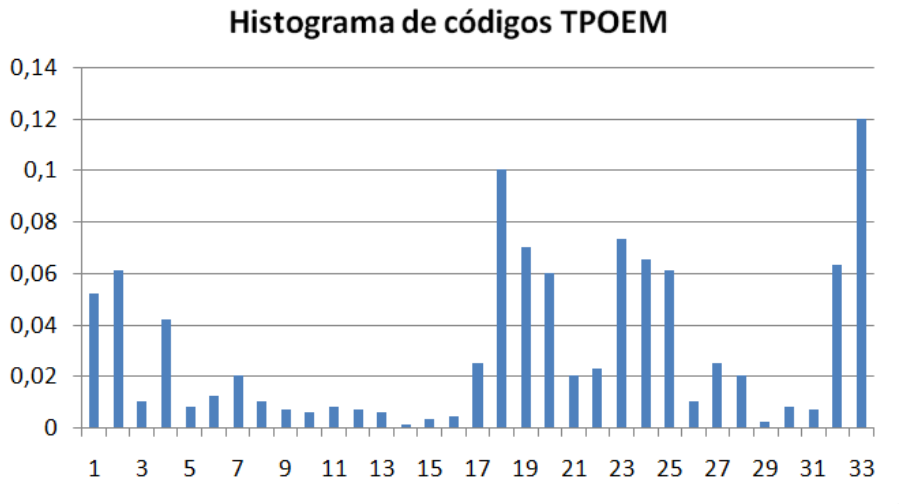
En el caso de problemas de aprendizaje de máquina que involucran la clasificación a partir de un número finito de muestras, si el número de muestras no se incrementa exponencialmente con el número de parámetros, la adición de parámetros decrementa el desempeño del sistema. Esto es conocido como Efecto Hughes [73].

4.3.1. Estimación de la dimensión intrínseca

Un conjunto de datos de dimensión d puede ser transformado a un nuevo conjunto de dimensión D, donde $D > d$ al añadir ruido aleatorio, variables inútiles, variables replicadas, variables codependientes, constantes, etc. Como consecuencia, puede ser establecido que muchos conjuntos de alta dimensión pueden ser transformados a espacios de menor dimensión sin pérdida de información relevante si un asunto similar es resuelto: cómo eliminar esas variables ruidosas, redundantes y, en general, inútiles. Un problema es obtener el número de dimensiones en el cual el conjunto de datos original X puede ser embebido sin pérdida de información importante.

En nuestro conjunto de datos esperamos una alta correlación y redundancia debido a ciertas razones. Tal como se mostró en el capítulo 3, una etapa de la técnica TPOEM es el arreglo de los datos en histogramas por celdas espaciales. Los histogramas TPOEM, tal como sucede en histogramas basados en LBP y muchos histogramas en general, no tienen una distribución homogénea. Hay patrones típicos y elementos cuya contribución es muy pequeña. La naturaleza no homogénea de los histogramas TPOEM puede ser relacionada con teoría de la información. Si tenemos un vector con parámetros típicos

Figura 20. Histograma de códigos TPOEM



y parámetros que nunca o rara vez ocurren, el valor de la entropía no es óptimo, lo cual significa que el vector puede ser embebido en una menor dimensión sin pérdida de información. La entropía de Shannon está definida así 4.16:

$$H(X) = -\sum_{i=1}^{n} p(x_i) log_b p(x_i) \tag{4.16}$$

En nuestro caso, al estimar la distribución de la entropía, los códigos TPOEM más representados hacen que globalmente más del 90 % de la energía de los códigos TPOEM esté concentrado en menos del 60 % de los códigos, lo cual es indicativo de que el tamaño puede ser reducido mediante codificación.

En la figura 20 ilustramos esta situación. La codificación TPOEM tiene un total de 33 diferentes códigos mapeados posibles. No obstante, nótese cómo un buen número de códigos tienen baja representación en los datos, mientras que otros códigos, especialmente los códigos 33, 18, 23, 19, 24, 32, 25 y 2, concentran la mayor parte de la representación. Esto sugiere que la información TPOEM codificada puede ser representada en un espacio de menor dimensión sin pérdida de capacidad de discriminación ni información.

Es importante señalar, sin embargo, que el hecho de que los códigos puedan eventualmente ser reducidos mediante técnicas como codificación no implica que sea recomendable hacerlo. La reducción de dimensiones por codificación, por ejemplo por codificación Huffman [72], típicamente tiene como objetivo reducir el tamaño de los códigos desde un punto de vista de entropía. Sin embargo, en tanto que los códigos iniciales son vectores descriptores de distintas clases, nada garantiza que códigos que eran lejanos en el conjunto inicial usando cierta métrica, por ejemplo distancia Mahalanobis o euclidiana, no sean cercanos en la nueva codificación. Más aún, es posible que códigos que representaban clases distintas sean incluso codificados por el mismo o muy cercano

código en la nueva representación de los datos (el primer caso si la codificación admite cierto grado de pérdida). Por otra parte, el uso de técnicas supervisadas de reducción de dimensiones conduce a otra clase de problemas; el más severo de todos es que para evitar errores metodológicos no se deben usar muestras para la reducción en la validación del sistema[3]. Esto implica que las muestras usadas para la reducción supervisada deben ser excluidas del proceso de validación, lo cual es un importante compromiso: cuantas menos muestrasse usen en el proceso de reducción de dimensiones, menos representación del espacio de alta dimensionalidad, pero cuantas más muestras se usen, más reducido el conjunto de datos disponibles para la validación. Esto es parcialmente solucionado por metodologías de *leave-subjects-out*, pero no deja de ser un inconveniente importante, especialmente porque las muestras de expresión facial no están uniformemente distribuidas por clases, sino que hay clases con mucha menor representación (especialmente miedo y tristeza), lo cual hace que retirar muestras de estas clases sea delicado.

Estimación por análisis de componentes principales: Nuestra primera aproximación para estimar la dimensión intrínseca de los datos fue usar una estimación PCA. Es importante acotar que no esperábamos obtener una solución apropiada con esta estimación, debido a las limitaciones de PCA, pero fue usada para tener una idea inicial. Para ello obtuvimos los eigenvalores λ para X y el valor de dimensión intrínseca fue determinado cuando el valor de λ_i era menor que cierto umbral. Otra aproximación típicamente usada es acumular el valor de λ_i y la dimensión intrínseca es asumida cuando la suma acumulada alcanza un umbral determinado. En la figura 21 se muestran las dimensiones intrínsecas para 3 orientaciones espaciales y 1 orientación temporal de TPOEM [4], usando $\lambda = 0{,}95$.

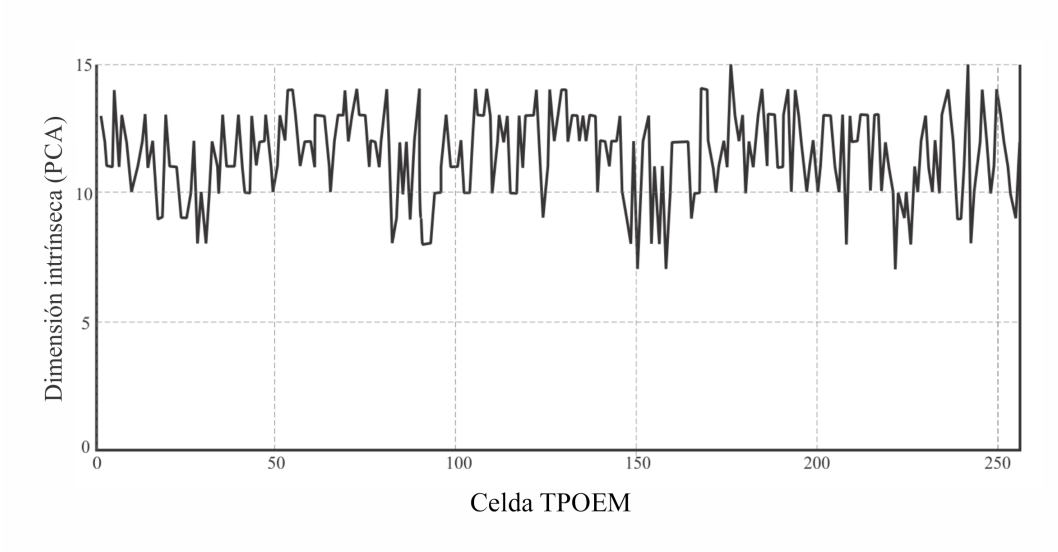
La estimación de dimensiones por PCA es de entre 7 y 15 por celda, pese a que los datos originales estaban en dimensión considerablemente mayor de promedio 34 por celda. Esto indica que hay una alta correlación lineal entre los datos, lo cual es concordante con nuestra hipótesis inicial. Sin embargo, teniendo en cuenta las limitaciones normalmente presentes en la estimación lineal por PCA, decidimos implementar un estimador más adecuado para incluir las posibles relaciones no lineales entre las variables.

Estimación de dimensión por *Nearest-Neighbor Maximum Likelihood Estimator* (NNMLE) NNMLE trabaja de manera similar que los estimadores MLE [90]

[3]Si bien esto es obvio, en la práctica un considerable número de trabajos que requieren de sistemas de clasificación usan una aproximación por etapas modulares: obtención de parámetros, reducción de dimensiones, entrenamiento y validación, sin tener en cuenta que incluso haciendo validación cruzada, si se usaron las mismas muestras en la reducción supervisada de dimensiones y en la validación, la metodología es incorrecta y conduce a tasas de acierto artificialmente elevadas.

[4]Cuando se realizaron estas pruebas aún no habíamos determinado que 2 orientaciones espaciales eran suficientes para describir la expresión facial, reduciendo así el cálculo del algoritmo VPOEM en casi un 30 %. Debido a ello, el cálculo de la dimensión intrínseca en esta sección por PCA fue realizado con 3 orientaciones espaciales. Posteriormente consideramos innecesario repetir estas pruebas para 2 orientaciones espaciales, por cuanto el proceso de reducción de dimensiones final no fue realizado con PCA

Figura 21. Estimación de dimensiones de los parámetros TPOEM usando PCA



[127]. MLE asume que todos los puntos en el conjunto de datos $X_i \in \Re^p$ son la transformación de $Y_i \in \Re^q$ de un espacio de menor dimensión: $X_i = g(Y_i)$. La dimensión q es menor que la dimensión p y g es una transformación continua y homogénea por mapeo [66]. Como tal, el mapeo de los vecinos cercanos de X_i debe ser también cercano en Y_i. LA forma general del estimador MLE está dada por la ecuación 4.17.

$$\hat{m}_R(x) = \left[\frac{1}{N(R,x)} \sum_{j=1}^{N(R,x)} \log \frac{R}{T_j(x)} \right]^{-1} \tag{4.17}$$

donde $N(R,x)$ es el número de muestras a una distancia R del punto x, $T_j(x)$ es la distancia de x a su j-ésimo vecino más cercano y R es el radio de una hiperesfera alrededor de x.

En la práctica, el uso de k vecinos más cercanos puede ser más conveniente. En concordancia, la ecuación NNMLE para obtener la dimensión intrínseca por muestra está dada por la ecuación 4.18.

$$\hat{m}_K(x) = \left[\frac{1}{k-1} \sum_{j=1}^{k-1} \log \frac{T_k(x)}{T_j(x)} \right]^{-1} \tag{4.18}$$

Para obtener la dimensión intrínseca del conjunto de datos hay varias opciones. Inicialmente, $\hat{m}_k$ puede ser calcculado como la media de $\hat{m}_k(x)$ y la dimensión intrínseca es estimada usando un valor constante de k, un rango de valores de k o usando dos valores k_1 y k_2:

$$\hat{m}_k = \frac{1}{n}\sum_{i=1}^{n}\hat{m}_k(x_i) \quad , \quad \hat{m} = \frac{1}{k_2 - k_1 + 1}\sum_{k=k_1}^{k_2}\hat{m}_k \tag{4.19}$$

Estimación de dimensiones por *Local Clustering NNMLE* (LC-NNMLE) El principal inconveniente de estas aproximaciones previas es que los valores de k son constantes, sea usando un valor individual o con dos valores distintos como en la ecuación 4.19. En nuestro problema hay datos agrupados, correspondientes hasta cierto punto a la expresión facial por muestra. Adicionalmente, hay un alto número de *outliners* debido a ruido estadístico y a la presencia de valores que no proporcionan información sobre la expresión facial todo el tiempo. Por ejemplo, algunas celdas pueden ser relativamente buenos clasificadores de ciertas expresiones faciales, pero su salida ruidosa para otras expresiones. Debido a estas consideraciones realizamos una modificación a NNMLE que llamamos LC-NNMLE. Los k vecinos más cercanos son aún usados, pero el valor de k depende de la densidad de puntos alrededor del punto x_i. La densidad relativa de puntos en una hiperesfera de radio R alrededor del punto x_i puede ser calculada de acuerdo con la ecuación

$$D\left(R, x\right) = \frac{N(R, x)}{\left(\frac{\pi^{n/2}}{\Gamma\left(\frac{n}{2}+1\right)}\right)} \tag{4.20}$$

donde

$$\Gamma\left({}^{1}/_{2}\right) = \sqrt{\pi};\;\; \Gamma\left(1\right) = 1;\;\; \Gamma\left(x+1\right) = x\Gamma\left(x\right) \tag{4.21}$$

Lamentablemente, el cálculo de la densidad de datos en espacios de alta dimensión es un problema, por cuanto el valor de $\Gamma(\frac{n}{2}+1)$ con valores de n altos es considerablemente grande. Por ejemplo, para $n = 34$, que es nuestro caso promedio, $\Gamma(\frac{34}{2}+1) \approx 3{,}57 \times 10^{14}$, de manera que la función D forzosamente tiene valores bastante pequeños y muy variables. La variabilidad de D está dada porque al elegir R, un valor muy pequeño puede hacer que no haya ningún punto en la vecindad, especialmente teniendo en cuenta el gigante volumen de un espacio n-dimensional alto. Por otra parte, un valor R grande puede reducir la precisión de la estimación. De esta forma, la estimación de la densidad de datos usando D fue apropiada para valores de n relativamente pequeños y para datos creados artificialmente (hiperesferas o hipercubos embebidos en espacios dimensionales de mayor dimensión, incluyendo ruido Gaussiano), mas no fue adecuado para nuestros datos limitados.

Debido a esta limitación, nuestra siguiente aproximación fue usando un parámetro como métrica del nivel de *clustering* de datos alrededor de los puntos x_i. Para cada punto x_i en el conjuno de datos, se construyó una matriz de distancias Euclidianas entre cada par de puntos. El costo de memoria de este procedimiento es considerable, especialmente si el número de muestras en el conjunto de datos es elevado; no obstante, el procesamiento no es complejo. Una vez obtenida, se calcula un grafo truncado, en

el cual se eliminan las rutas correspondientes a distancias mayores que un umbral (en nuestro caso, experimentalmente se definió este valor de umbral tal que se eliminen aproximadamente 90 % de las rutas del grafo) y se obtiene un coeficiente de *clustering* por nodo tal como se sugiere en [176]. El coeficiente es calculado según la ecuación 4.22.

$$C_i = \frac{2\left|\{e_{j,k} : v_j, v_k \in N_i, e_{j,k} \in E\}\right|}{k_i\left(k_i - 1\right)} \tag{4.22}$$

donde C_i es el coeficiente de *clustering* local, v y e son los vértices y bordes contenidos en el grafo $G(V, E)$, N_i es la vecindad para el vértice v_i y k_i es el número de vecinos del vértice v_i. Con el valor de C_i, el *clustering* de los datos puede ser aproximado y los *outliners* no son incluidos en la estimación local NNMLE.

La siguiente etapa del proceso es definir el número de vecinos a usar por cada punto x_i en función de su valor de C_i correspondiente. Si bien es posible desarrollar una función compleja para determinar el valor, en nuestro caso utilizamos una función simple de mapeo de modo que $f(C_i) \to k_i$ y usamos los valores obtenidos de k_i como vecindades variables en NNMLE convencional. La ecuación de estimación de dimensión intrínseca por muestra es mostrada en 4.23.

$$\hat{m}_k(x) = \left[\frac{1}{k_i - 1}\sum_{j=1}^{k-1}\log\frac{T_{k_i}(x)}{T_j(x)}\right]^{-1} \tag{4.23}$$

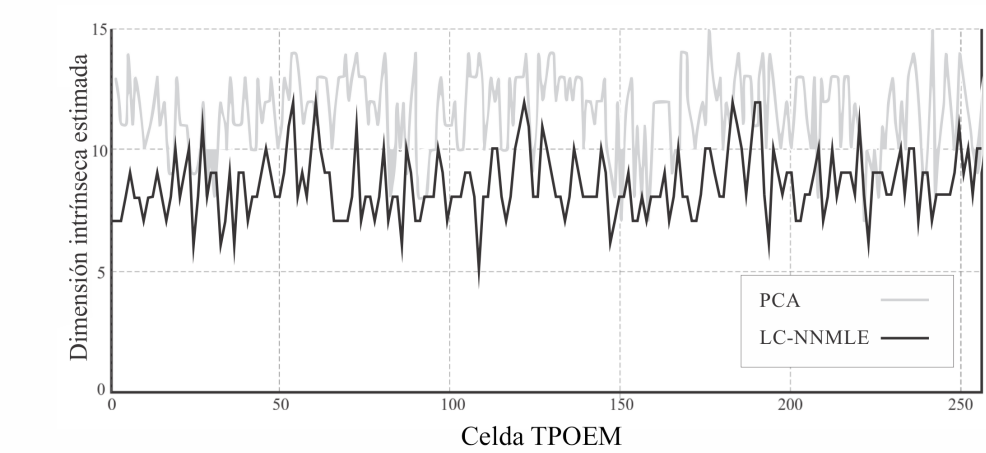
Es importante precisar que los costos completos de este proceso son bastante elevados, especialmente cuando el tamaño del conjunto de datos es grande. La construcción de la matriz de distancias Euclidianas es un procedimiento matricial simple, si bien su costo de memoria puede ser grande. Sin embargo, el cálculo del grafo G sí es considerablemente costoso, requiriendo de hasta 14 horas de cálculo en nuestro equipo (i7, 2.4 GHz, 8 GB RAM), aunque es un procedimiento que se realiza una sola vez para todo el conjunto de datos de prueba y luego, para validación de un nuevo conjunto de datos, sólo requiere la proyección a un nuevo espacio dimensional usando la matriz de transformación.

En la figura 22 se muestra la estimación de dimensión intrínseca usando LC-NNMLE.

Se puede observar que LC-NNMLE calcula una dimensión intrínseca considerablemente menor que la estimada por PCA. Es decir, es razonable esperar que una reducción de dimensión más dramática pueda ser conseguida hasta cierto punto. Un inconveniente, sin embargo, es que los estimadores de dimensión intrínseca tienden a subestimar la dimensión de un conjunto de datos si la dimensión intrínseca real es relativamente grande. Para probar esto, creamos un espacio agrupado de dimensión variable en un espacio de más alta dimensión [5], incluyendo ruido Gaussiano. Posteriormente se realizó estimación de dimensión intrínseca usando un número fijo de vecinos, con estimación NNMLE y LC-NNMLE. La estimación de la dimensión intrínseca es precisa cuando la

[5]Por espacio agrupado nos referimos a un conjunto de datos con muestras agrupadas en distintos *clusters*, con el fin de simular datos de alta dimensión que representan distintas clases, en oposición a hiperesferas o hipercubos que representan todos los datos en un mismo grupo

Figura 22. Estimación de dimensiones de los parámetros TPOEM usando LC-NNMLE



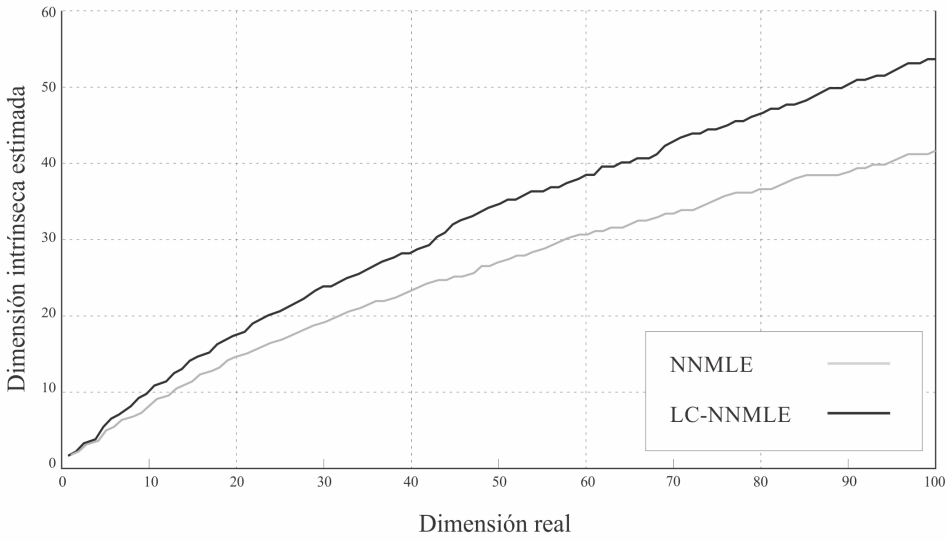
dimensión real es pequeña, pero en la medida en que la dimensión intrínseca se incrementa, la estimación pierde precisión. La principal razón para esta discrepancia entre la dimensión real y la estimada es que estamos usando el mismo número de puntos para representar el conjunto de datos, pero los puntos se tornan más dispersos cuando la dimensión en la cual están embebidos aumenta. Este problema es agravado porque el incremento de la distancia entre los puntos es exponencial, de manera que se requiere de un número mucho más elevado de muestras para representar el *manifold* de alta dimensión, incluso si este *manifold* es de forma relativamente simple. Para demostrar este problema, se realizó una nueva estimación de dimensión intrínseca con datos generados usando el mismo protocolo, pero incrementando linealmente el número de datos en función de la dimensión embebida. La dimensión intrínseca fue calculada usando NNMLE y LC-NNMLE. Los resultados se muestran en la figura 23.

Al incrementar linealmente el número de puntos por conjunto de datos para representar el conjunto n-dimensional, la precisión de la estimación de dimensión intrínseca mejora, pero no tan rápidamente porque la dispersión de los datos se incrementa a una tasa mucho mayor. Más aún, éste es apenas un ejemplo teórico, por cuanto en la práctica el número de muestras es generalmente limitado, de modo que incrementar el conjunto de datos no es realizable para mejorar la estimación.

Debemos precisar que para datos no agrupados en *clusters* nuestras pruebas iniciales no mostraron resultados mejores que usando NNMLE convencional, pero en estas pruebas con datos agrupados obtuvimos resultados considerablemente más precisos. Esta mejora es obtenida debido a la naturaleza quasi-supervisada del algoritmo LC-NNMLE.

Se puede determinar en la figura 23 que si bien el problema de la subestimación de dimensión con alta dimensión de los datos persiste, los resultados tienen mejor precisión que usando NNMLE convencional, por cuanto aunque los valores de dimensión intrín-

Figura 23. Estimación de dimensiones de un cluster de datos n-dimensionales usando NNMLE y LC-NNMLE



seca obtenidos por LC-NNMLE son menores que la dimensión real de los datos, en todo caso son más cercanos que los obtenidos con NNMLE. En consecuencia, determinamos que LC-NNMLE puede ser una buena alternativa para estimar la dimensión de nuestros datos de alta dimensionalidad.

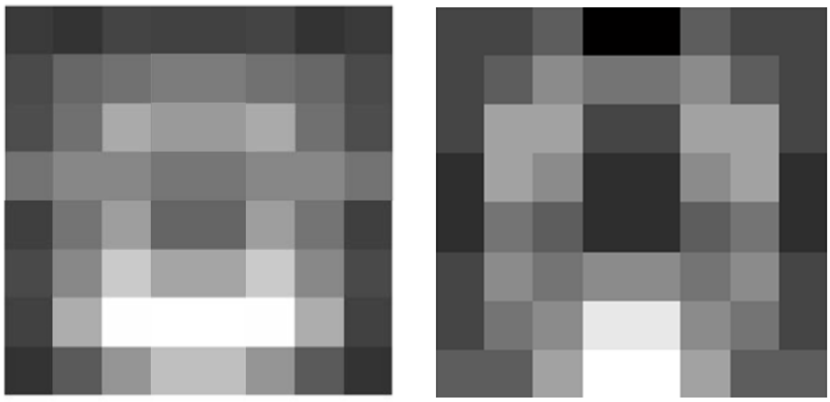
En la figura 24 se muestra los resultados de la estimación de dimensión intrínseca por celda espacial usando NNMLE y LC-NNMLE. Con el fin de facilitar la visualización, en la figura se muestra el promedio reflejado de las dimensiones intrínsecas, con el fin de que la visualización sea simétrica.

Las regiones alrededor de la boca, ojos y cejas tienen los más altos valores de dimensión intrínseca en ambos casos. Esto es razonable, por cuanto esperamos que haya más información en estas regiones que tienen mayor variabilidad tanto interclase como intraclase. Adicionalmente, es un buen resultado, porque esto implica que menos datos son requeridos en las regiones que no proporcionan mucha información, de modo que la clasificación es más rápida en estas regiones y más precisa en regiones que aparentemente proporcionan mayor capacidad de discriminación. El uso de LC-NNMLE determina una dimensión intrínseca por celda espacial menor que MMLE, mientras que preserva altos valores de dimensión en las regiones en las que se espera mayor información.

4.3.2. Reducción no supervisada de dimensiones

Una vez obtenida la dimensión intrínseca por celda espacial, el siguiente paso es la reducción de dimensiones del conjunto de datos usado. El conjunto total corresponde a 128 vectores de orientaciones espaciales y 64 vectores de orientación temporal, para

Figura 24. Estimación de dimensiones de las celdas espaciales TPOEM usando NNMLE y LC-NNMLE



una matriz completa de 192 vectores que conforman TPOEM. En primer lugar usamos metodologías simples de reducción no supervisada de dimensiones, para tratar de establecer su utilidad y sus inconvenientes.

PCA: La primera aproximación fue el uso de reducción de dimensiones usando PCA. Para ello, usamos los n más importantes eigenvectores en la reconstrucción lineal de los datos, siendo n la dimensión intrínseca estimada por vector usando estimación PCA [6]. Si bien esto dificulta la comparación de resultados, ya que con otros métodos de reducción de dimensiones usamos otro objetivo de reducción, puede al menos dar idea general de la utilidad de PCA en la reconstrucción y uso de los datos reducidos.

La reducción de dimensiones usando PCA condujo a ciertos inconvenientes, apoyados por la teoría en la temática. PCA proyecta los datos en el eje en el cual hay máxima varianza. Sin embargo, esto no garantiza que el nuevo conjunto de datos proyectados contenga parámetros adecuados para la discriminación entre clases. Adicionalmente, siendo PCA una proyección lineal, descarta posibles correlaciones no lineales en el conjunto de datos. Como consecuencia de estas limitaciones, el desempeño de la clasificación usando PCA para reducción de dimensiones es pobre, con aún menor precisión que el desempeño usando los datos completos sin ningún procesamiento.

Lamentablemente, la clasificación usando datos reducidos por PCA fue notablemente peor que usando datos no reducidos. Por otra parte, esto era esperado, por cuanto la reducción no supervisada PCA de datos de alta dimensión puede probablemente proyectar los datos en direcciones de alta varianza que no correspondan a alta discriminación entre clases sino a alto ruido en el conjunto de datos inicial. Para determinar esto, hicimos una prueba simple de clasificación de dos clases por *pattern matching*, ira

[6] No usamos la dimensión intrínseca estimada por LC-NNMLE por cuanto ésta corresponde a una estimación no lineal, de modo que PCA tendría problemas de reconstrucción con una restricción más drástica que la determinada por estimación lineal de dimensión intrínseca.

vs. neutral, con datos completos y datos reducidos PCA. Los resultados se muestran en la tabla 11.

Tabla 11. Comparación de clasificación usando datos sin procesar y reducidos por PCA

	No reducidos		Reducidos PCA	
	Ira	Neutral	Ira	Neutral
Ira	**100**	23	**73**	50
Neutral	38	**521**	252	**307**

Es claro que el uso de reducción de dimensiones por PCA fue altamente perjudicial para la precisión de la clasificación, pese a haber usado una reconstrucción con más del 95 % de la energía de los datos preservada. Esto es explicado porque la mayor varianza en el conjunto de datos inicial corresponde a la varianza entre individuos, no entre clases, de manera que la proyección PCA atenúa considerablemente las características representativas de la expresión facial, dificultando y, en algunos casos, imposibilitando por completo el reconocimiento de la expresión facial.

Autoencoding: El autoencoding es una herramienta poderosa usada típicamente para aprender una representación comprimida y distribuida de un conjunto de datos [10], de modo que su uso en reducción de dimensiones es amplio. En este trabajo, implementamos *autoencoding* usando una red neuronal *feedforward* con cuatro capas. Las primeras tres capas tienen funciones de excitación sigmoideas y la última una función de excitación lineal.

Experimentalmente obtuvimos el número de nodos aproximadamente óptimo por capa usando una validación LSO con los datos[7]. Iterativamente, el número de nodos fue cambiado, de manera que el *autoencoder* fue entrenado con un conjunto de entrenamiento y el resto de los datos fue usado para la validación. Posteriormente el número de capas y nodos era cambiado iterativamente y el proceso fue repetido hasta alcanzar el mínimo error.

A partir de la etapa de entrenamiento, se obtuvo que el número óptimo de nodos fue 61 para la primera capa, 32 para la segunda y 25 para la tercera. El número de pesos por calcular era de aproximadamente 5143 por vector TPOEM (el valor exacto depende del valor de dimensión intrínseca al cual se desea reducir los datos, que es variable de acuerdo con la posición espacial del vector).

La principal dificultad en el entrenamiento de los *autoencoders* fue la alta dimensión de los datos. Cada generación de pesos requiere de un considerablemente elevado número de individuos creados por mutación, con el fin de seleccionar el mejor de todos, y debido al elevado número de muestras y dimensiones, la convergencia del algoritmo es bastante lenta. Debido a esto, el tiempo de procesamiento del algoritmo es bastante elevado, lo cual dificultó considerablemente el desarrollo de este procedimiento.

[7]Resaltamos que el número de nodos no se puede en realidad optimizar directamente debido a la naturaleza no heurística del entrenamiento, de modo que los valores obtenidos son una aproximación.

Nuestros resultados mostraron características interesantes de los *autoencoders*. Con una elección adecuada del número de nodos y un alto número de iteraciones para garantizar que los pesos tienen adecuadas mutaciones, el grado de reducción de dimensiones es adecuado, con reducción promedio de 64 %. Hay, sin embargo, algunos inconvenientes para el problema de reconocimiento de patrones. El primer inconveniente es que las nuevas dimensiones no necesariamente proporcionaron buena capacidad de discriminación, de manera que el conjunto transformado es comprimido, pero el problema de clasificación fue en algunos casos más complicado. El segundo inconveniente está relacionado con los problemas de overfitting que tienen los sistemas basados en redes neuronales. Se ajustan muy bien a datos conocidos, pero pueden tener inconvenientes con datos nuevos, tal como se encontró con los datos de validación. Debido a esto, entradas parecidas, que probablemente representan datos de la misma clase en un problema de clasificación, pueden tener salidas muy distintas debido al *overfitting*, así que la clasificación se torna más difícil. Debido a esto, incluso si los *autoencoders* proporcionaron una alta rata de compresión, los nuevos datos no fueron adecuados para el problema de clasificación.

Sin embargo, esta parte del desarrollo sirvió como fundamento de los sistemas de clasificación más avanzados basados en deep learning, que mostramos en el capítulo 6, en donde mostramos cómo el uso de autoencoders apilados como etapas intermedias de un problema de clasificación supervisado tiene excelentes resultados incluso en un entorno de alta dimensionalidad.

4.3.3. Reducción supervisada de dimensiones

Nuestra siguiente aproximación fue usar reducción supervisada de dimensiones. La principal ventaja del uso de técnicas supervisadas de reducción de dimensiones es que el nuevo conjunto de datos es embebido en proyecciones que intentan preservar la separación entre clases, de manera que el problema de clasificación es mejorado. En el caso de clasificacion por variables débiles, como en nuestro caso, existe una restricción importante, sin embargo. Una variable individual no es buena clasificadora por sí misma, de manera que la arquitectura principal es el ensamble de todos los parámetros como metaclasificadores débiles y cuya combinación puede determinar un sistema fuerte de clasificación. Esto, sin embargo, puede ser problemático en la etapa de reducción supervisada de dimensiones, pues se basa en la adecuada separación de las clases en cada variable, lo cual no necesariamente es un hecho. Adicionalmente, es importante siempre tener en cuenta que un procedimiento supervisado implica forzosamente que las muestras usadas en la construcción de la proyección deben quedar descartadas de etapas posteriores de validación. Esto se explica porque una reducción supervisada de dimensiones es, de hecho, una etapa de clasificación, de manera que usar muestras tanto en la determinación de la reducción de dimensiones como en la validación del problema es un error metodológico que, lamentablemente, es recurrente en trabajos de clasificación encontrados en la bibliografía revisada.

LDA supervisado: La primera técnica de reducción supervisada de dimensiones que probamos fue LDA supervisado. LDA convencional separa los datos entre dos clases, de manera que incluimos una etapa de clasificación multiclase para separar entre las seis expresiones y la instancia neutral.

Para nuestro problema de clasificación de 7 clases usamos un método modificado de 1 vs. todos. Las metodologías convencionales de 1 vs. todos implican el diseño de N clasificadores (uno por clase), en el cual cada clase tiene su propio clasificador. El entrenamiento es realizado al separar los datos de acuerdo a su pertenencia o no al conjunto de la clase entrenada, como se muestra en la ecuación 4.24.

$$\begin{aligned} \forall x_i \in \ class\, a, \ \ x_i \ &\in \ class\, I \\ \forall x_i \notin \ class\, a, \ \ x_i \ &\in \ class\, II \end{aligned} \tag{4.24}$$

Una vez los grupos clase I y clase II son obtenidos, se calculan los coeficicientes de transformación C_a, donde a corresponde a la clase entrenada. El proceso se repite para todas las clases y una metodología de clasificación es realizada, por ejemplo seleccionando el clasificador cuya salida es más alta.

En nuestro caso usamos una aproximación similar, pero la arquitectura 1 vs. todos no fue expresión vs. resto de los datos, sino expresión vs. instancia neutral. De esta manera, los datos para cada expresión fueron enfrentados a los datos de instancias neutrales. Como consecuencia, la salida ideal describe qué tan lejos una entrada está de la instancia neutral e, indirectamente, qué tan lejos está de una expresión particular. Preferimos esta aproximación porque en nuestra opinión es mejor una metodología cuyo error principal esté en expresiones erróneamente clasificadas como neutrales y no en expresiones erróneamente clasificadas como otra expresión, y el uso de esta metodología 1 vs. todos adaptada permite esto.

El proceso fue repetido para todos los 192 vectores TPOEM, de manera que el entrenamiento completo incluye el cálculo de 192×6 transformaciones LDA supervisadas. Cada uno de los conjuntos C por celda (vector TPOEM) es asumido como un clasificador débil, de manera que el puntaje final es la suma de las salidas de cada clasificador y la clasificación es hecha por una metodología *winner takes all*.

Si bien la técnica usada no es compleja, de modo que no se esperan resultados de clasificación comparables con los del estado del arte (y tampoco es el objetivo de esta etapa, en todo caso), esperábamos que si los resultados son adecuados podemos determinar que los datos proyectados siguen siendo descriptores adecuados de la expresión facial.

La clasificación fue diseñada por la fusión de los clasificadores completos con *winner takes all*, salvo en los casos en los que no haya un ganador claro. En esos casos, la muestra es declarada como neutral. Los resultados se observan en la tabla 12.

Estos resultados muestran que LDA supervisado proyectó los datos en una nueva dimensión en la cual las clases son aún separables. De otra forma, los resultados de clasificación simple hubiesen sido considerablemente inferiores. De esta forma, se puede

Tabla 12. Clasificación usando reducción supervisada LDA

Expresión	Clasificación
Ira	87.8 %
Disgusto	79.37 %
Miedo	67.5 %
Alegría	89.43 %
Neutral	70.41 %
Tristeza	61.39 %
Sorpresa	79.78 %

determinar que una reducción de dimensiones supervisada, incluso relativamente simple, puede ayudar a simplificar el problema.

***Maximum Collapsing Metric Learning (MCML)*:** Con el fin de implementar MCML con nuestros datos, asumimos que cada vector TPOEM es un clasificador individual y fue entrenado con MCML con una arquitectura similar a la usada para LDA supervisado. El ensamble de clasificadores fue uado para construir un clasificador más fuerte. Para probar la validez de MCML como reductor supervisado de dimensiones, incluimos una etapa de validación.

Es importante resaltar que para los resultados mostrados en esta etapa no fue implementado ningún sistema de clasificación experto. Como tal, la salida de los clasificadores es siempre la adición simple de los puntajes individuales por celda. El principal objetivo del trabajo presentado en este capítulo es mostrar la viabilidad de algunas técnicas de reducción de dimensiones para reducir el tamaño del conjunto de datos, mientras se perserva la información necesaria para separar los datos de acuerdo a las clases. Consecuentemente, con el fin de determinar apropiadamente si el dato fue reducido adecuadamente, asumimos que el uso de un sistema sofisticado de clasificación podría ser equívoco, puesto que algunos resultados buenos de clasificación pueden ser obtenidos debido a complejidad del sistema de clasificación en vez de la validez del nuevo conjunto de datos embebido. Es claro que si los datos no son representativos de las clases, sin importar cuán complejo sea el sistema de clasificación, los resultados de clasificación no pueden ser buenos. Sin embargo, un sistema complejo de clasificación puede atenuar o eliminar complejidades de los manifolds embebidos creados por la reducción de dimensiones, de manera que el uso de clasificadores más simples permite una comparación más fácil y directa entre técnicas de reducción.

La clasificación usada para los datos transformados por MCML fue la distancia Mahalanobis entre cada dato por celda en el conjunto de validación y el centro de masa por celda por expresión en el conjunto de entrenamiento, en la ecuación 4.25.

$$\bar{x}^{tr}_{k,c} = \frac{1}{N} \sum_{i} x^{tr}_{i,k,c} \tag{4.25}$$

donde N es el número de muestras por expresión en el conjunto de entrenamiento, $x^{tr}_{i,k,c}$ es la muestra i, celda k, expresión c en el conjunto de entrenamiento. La distancia Mahalanobis entre una muestra en el conjunto de entrenamiento y una muestra de validación $x^{val}_{i,k,c}$ está dada por la ecuación 4.26.

$$D_M(x_{i,k,c}) = \sqrt{\left(x_{i,k,c} - \overline{x^{tr}_{k,c}}\right) S^{-1} \left(x_{i,k,c} - \overline{x^{tr}_{k,c}}\right)}, \quad i = 1, 2, ..., N \tag{4.26}$$

$$c = 1, 2, ..., 7 \tag{4.27}$$

Para esta parte del trabajo se usó distancia Mahalanobis en vez de distancia Chi-square porque la transformación MCML embebe los datos en una nueva dimensión en los cuales la dimensión de cada variable ya no guarda relación con la dimensión de las otras variables, en oposición a la transformación lineal supervisada LDA o a los datos sin procesar. Debido a esto, Mahalanobis es una mejor aproximación, pues descarta las posibles variaciones producidas por la adimensionalidad o dimensión abstracta de las nuevas variables.

El puntaje total está dado por la suma de distancias por expresión por celda:

$$S_M(i, c) = \sum_k D_M(x_{i,k,c}), \quad c = 1, 2, ..., 7 \tag{4.28}$$

Una vez los puntajes por expresión son obtenidos, la decisión es tomada de acuerdo con el valor de c para el cual el puntaje total sea mínimo. De manera similar al procedimiento usado en la reducción PCA, hicimos una prueba preliminar con ira vs. neutral con datos sin procesar y datos reducidos con MCML. Los resultados se muestran en la tabla 13.

Tabla 13. Comparación de clasificación usando datos sin procesar y reducidos por MCML

	No reducidos		Reducidos MCML	
	Ira	Neutral	Ira	Neutral
Ira	**100**	23	**85**	38
Neutral	38	**521**	221	**338**

Si bien los resultados son superiores a los obtenidos usando reducción por PCA, son inferiores a los obtenidos usando datos sin procesar. Es claro que en este caso la reducción de dimensiones ocasionó una importante pérdida de desempeño de clasificación pese a que MCML intenta maximizar las distancias inter clase, mientras minimiza las distancias intra clase. No obstante el uso de métrica Mahalanobis intenta normalizar el efecto de las unidades o dimensiones abstractas para cada variable, los resultados fueron

peores. En este trabajo ejecutamos diversos entrenamientos MCML con distintas soluciones iniciales aleatorias, pero incluso con el uso de un alto número de iteraciones (que fueron en todo caso limitadas, pues el entrenamiento de una reducción MCML para dos clases en varios casos requirió de más de 72 horas de cálculo), la convergencia a una solución no fue necesariamente alcanzada. Este es un problema común al lidiar con espacios de alta dimensión y un limitado número de muestras, y es particularmente crítico cuando debido a que cada variable es débil, los valores por muestra no necesariamente muestran una tendencia clara y diferenciable entre las clases.

No obstante, una prueba adicional con la reducción MCML más exitosa fue implementada, pero esta vez asignando distinto valor a los parámetros extraídos según su importancia aparente en la discriminación entre clases. Con esto, una nueva métrica fue obtenida así:

$$S_M(i,c) = \sum_k w_k D_M(x_{i,k,c}), \quad c = 1, 2, ..., 7 \tag{4.29}$$

Los valores de w_k dependen de la clasificación individual de cada celda k, y fueron obtenidos con un subconjunto disjunto de entrenamiento. Los resultados obtenidos se muestran en la tabla 14.

Tabla 14. Comparación de clasificación usando datos sin procesar y reducidos por MCML con pesos ponderados

	No reducidos		Reducidos MCML	
	Ira	Neutral	Ira	Neutral
Ira	**105**	18	**108**	15
Neutral	35	**524**	33	**526**

Estos resultados permiten ver que si bien incluir pesos ponderados representa una mejora tanto en datos sin reducir como en datos reducidos, la mejora es mucho más notable con los datos reducidos, superando los resultados globales de clasificación. Esto muestra cómo la reucción supervisada de dimensiones puede mejorar la clasificación, pero es importante ser cuidadoso con la manipulación de los datos en las nuevas dimensiones embebidas. Adicionalmente, esta prueba proporciona una conclusión positiva notable. La expresión de ira tiene usualmente una pobre rata de clasificación comparada con otras expresiones, porque muchos individuos muestran ira con ligeros gestos del ceño, mientras el resto de músculos faciales permanecen relativamente estáticos. Pese a este inconveniente, el sistema de clasificación básico usando distancias Mahalanobis mostró una tasa de clasificación de 87.25 % en el problema de dos clases ira vs. neutral. Naturalmente, se espera que la tasa de clasificación completa en el sistema de 7 clases sea menor, pero el resultado fue satisfactorio en todo caso teniendo en cuenta la simplicidad del sistema. Este resultado prueba que los descriptores TPOEM son adecuados para la caracterización de la expresión facial y que el uso cuidadoso de reducción de dimensiones puede ayudar al desempeño del sistema.

4.4. Conclusiones

La reducción de dimensiones para el problema de clasificación de la expresión facial probó ser una tarea compleja. Mientras que numerosos cálculos para la estimación de la dimensión intrínseca probaron que los datos podían ser transformados a una menor dimensión, existían varios inconvenientes para conseguir este propósito: i. Los estimadores de dimensión intrínseca tienden a subestimar la dimensionalidad de un conjunto de datos de alta dimensión, ii. Transformar los datos a una menor dimensión no necesariamente mejora la clasificación (y en muchos casos la empeora), iii. Usar técnicas supervisadas de reducción de dimensiones implica que los datos usados para la reducción de dimensiones deben ser descartados para su uso en la etapa de validación, lo cual limita aún más la considerable restricción de disponibilidad de datos de expresión facial. Más aún, en numerosas ocasiones estos problemas son de difícil detección, por cuanto por una parte es imposible determinar si la dimensión calculada es de hecho menor que la dimensión intrínseca real y por otra parte el desempeño de los clasificadores puede mejorar artificialmente, pero de forma metodológicamente equivocada, si se cometen algunos errores en el desarrollo del procesamiento de datos y el diseño de clasificadores, que son errores muy comunes encontrados en la bibliografía consultada. Con el fin de detectar y prevenir estos inconvenientes, las metodologías usadas tanto en la reducción de dimensiones como en los sistemas de clasificación fueron cuidadosas, usando siempre conjuntos disjuntos para reducción de dimensiones/entrenamiento y validación y metodologías de LSO en todos los casos.

La manipulación de datos con altas dimensiones es un problema que no es bien comprendido aún. Muchos de los inconvenientes son la difícil abstracción de espacios de muy alta dimensión, la imposibilidad de representar estos espacios con conjuntos de datos reales, típicamente limitados, las formas geométricas complejas que tienen los datos en altas dimensiones, el agrupamiento de datos en las esquinas de los espacios de alta dimensión y la dificultad del uso de métricas convencionales para medir distancias en estos espacios de alta dimensión. Históricamente, los sistemas de clasificación no han tenido un buen desempeño con datos de altas dimensiones, tal como se reseña en [40] y [47], especialmente teniendo en cuenta las expectativas surgidas en esta área, debido a que la metodología de clasificación puede ser propensa a sobre especialización, la caracterización de los espacios de alta dimensión es difícil con datos limitados y el ruido puede ser suficientemente grande para imposibilitar o al menos dificultar la tarea. Por último, la reducción de dimensiones puede transformar los datos a espacios de menor dimensión, pero con manifolds más complejos y en los cuales la abstracción de las nuevas dimensiones impidan el uso de ciertas técnicas de clasificación. Debido a esto, en muchos casos es preferible usar los datos completos sin procesar en vez de manipular los datos para embeberlos en dimensiones más pequeñas. En nuestro caso el problema de la alta dimensión fue parcialmente reducido al dividir el rostro en celdas espaciales, de manera que las muestras son representadas por un vector más limitado en tamaño en vez de usar los miles de bits individuales de todo el descriptor TPOEM, pero en todo caso el problema de la alta dimensión persiste.

En nuestro trabajo probamos varias técnicas de estimación de dimensiones y reducción de dimensiones convencionales, tales como PCA, WPCA, LDA, MCML y NNMLE, y desarrollamos e implementamos una técnica de estimación de dimensión intrínseca llamada LC-NNMLE, que mejora los resultados de NNMLE al tener en cuenta las complejidades locales de cada punto en el conjunto de datos en vez de usar un número fijo de vecinos más cercanos, que son problemáticos con el uso de métricas euclidianas. El principal inconveniente de esta nueva aproximación es que el cálculo de las complejidades locales de los *clusters* es una tarea de alto costo de cálculo y memoria, con costos exponencialmente crecientes en función de la dimensión original del conjunto de datos y el número de muestras. Este problema es atenuado porque el cálculo sólo debe ser realizado una vez para estimar la dimensión intrínseca y no se requiere posteriormente de otros cálculos. Los resultados mostraron que para datos agrupados en *clusters* la estimación de LC-NNMLE fue más cercana a la dimensión real que usando NNMLE, especialemente con valores más grandes de dimensión real.

Lamentablemente, el proceso propio de reducción de dimensiones para un problema de clasificación es complejo con las altas dimensiones de los datos usados. En tanto que el volumen de los espacios de alta dimensión es muy grande, los datos no ocupan este volumen de manera homogénea, sino que las muestras tienden a agruparse en las esquinas de los espacios de alta dimensión. Como tal, las técnicas de reducción no supervisada de dimensiones frecuentemente tienen un pobre desempeño en estos espacios de alta dimensión y la proyección de los datos a una menor dimensión puede hacer perder propiedades de discriminación entre clases. Este problema fue evidente en este trabajo, pues se mostró cómo las técnicas de reducción no supervisada de dimensiones consiguieron embeber los datos en espacios considerablemente más pequeños (en nuestro caso, PCA, WPCA y *Autoencoding*, pero algoritmos de clasificación simple, que tenían resultados apropiados para su limitada complejidad con datos sin procesar, se tornaron insuficientes para realizar la clasificación.

Por otra parte, las técnicas de reducción de dimensiones usadas probaron ser satisfactorias para embeber los datos en menor dimensión, preservando la capacidad de clasificación. Reiteramos en esta sección que es fundamental eliminar los datos usados para la reducción supervisada de dimensiones y no usarlos en etapas posteriores de validación, porque la reducción supervisada es de por sí un aprendizaje intermedio de clases. Preservar los datos para su uso en la etapa de validación, sea inadvertida o deliberadamente, hace crecer de manera artificial los resultados de la validación, especialmente si en la etapa de reducción supervisada se usa buena parte de los datos o incluso todos los datos, y este error es frecuentemente encontrado en la bibliografía.

5. Selección de parámetros

5.1. Introducción

Los parámetros TPOEM extraídos son determinados por 128 vectores de orientaciones por gradiente acumulado espacial y 64 vectores de orientación por gradiente acumulado temporal[1]. Estos vectores fueron procesados y su longitud por vector fue reducida usando técnicas no supervisadas y supervisadas de reducción de dimensiones en el capítulo 4. Sin embargo, en las pruebas preliminares en el capítulo 3 se mostró cómo algunas de las celdas espaciales tenían mayor relevancia en la clasificación que otras y, probablemente, algunas de las celdas son completamente irrelevantes en la clasificación. Debido a esto, en esas pruebas básicas iniciales se usó una ponderación basada en error para determinar el peso de cada celda en la clasificación final. Sin embargo, es posible desarrollar técnicas más avanzadas de selección de parámetros (celdas) del conjunto de datos completos.

Si bien es intuitivo pensar que usar un mayor número de parámetros que el necesario no es perjudicial para el sistema de clasificación final, por ejemplo asumiendo que las variables innecesarias serán tratadas como tal por el clasificador experto, en realidad usar variables adicionales puede deteriorar el desempeño de la clasificación global. Esto ocurre debido a que la presencia de estos parámetros adicionales puede hacer que los sistemas de clasificación experto no tengan convergencia (particularmente habitual con redes neuronales, autoencoders y sistemas minmax) o que el ruido añadido por las variables superfluas imposibilite la clasificación exitosa.

Debido a estas consideraciones, en este capítulo mostraremos las técnicas diseñadas e implementadas para realizar la selección de parámetros a partir del conjunto de datos TPOEM extraídos de las muestras.

Nuestra nomenclatura es la misma encontrada típicamente en la bibliografía. Extracción de parámetros se refiere al procesamiento realizado con el fin de transformar un conjunto de parámetros de un universo a un espacio de menor dimensión, mientras que selección de parámetros se refiere a la obtención de un subconjunto de parámetros a partir del conjunto inicial, sin realizar transformación sobre los mismos. Es decir, siguiendo esta nomenclatura, la etapa previa de reducción de dimensiones es equivalente a la extracción de parámetros.

En la sección 5.2 haremos una breve reseña teórica de selección de parámetros y estado del arte. En la sección 5.3 mostraremos nuestra metodología usada para selección de parámetros en un entorno de clasificación débil. En la sección 5.4 haremos una descripción de los resultados experimentales de este trabajo. Finalmente, en la sección 5.5 mostraremos las conclusiones relevantes del trabajo referido a este capítulo.

[1]En realidad, en este punto del trabajo los parámetros extraídos son 256, siendo 192 parámetros espaciales y 64 parámetros temporales. No obstante, en el proceso de clasificación descrito en el capítulo 6, encontramos que usar 128 vectores espaciales es suficiente para la descripción de clases, de manera que a partir de entonces, salvo en pruebas específicas de comparación, se usó esta aproximación.

Figura 25. Esquema general del proceso de selección de parámetros

5.2. Fundamentación teórica y estado del arte de selección de parámetros

La selección de parámetros está definida como el conjunto de actividades realizadas con el fin de seleccionar parámetros relevantes de un conjunto completo de parámetros tal que el nuevo conjunto de parámetros sea más adecuado para realizar clasificación [58]. En general, hay dos razones principales para excluir un parámetro del conjunto completo inicial. Primero, el parámetro no necesariamente proporciona información sobre las clases, lo cual no sólo no ayuda en el problema de clasificación, sino además puede deteriorar el desempeño de la clasificación debido al ruido añadido. Segundo, el parámetro puede ser redundante. Esto es, la información contenida en el parámetro ya está presente en otro parámetro o subconjunto de parámetros, sea por contener información idéntica o muy parecida o por contener información que pueda ser determinada como la transformación lineal o no lineal a partir de otros parámetros [2]. La selección de parámetros es principalmente usada cuando el número de muestras en el conjunto principal de datos es relativamente pequeño y el número de parámetros es grande. En la figura 25 se muestra el esquema básico del proceso de selección de parámetros, en el cual a partir de un conjunto de datos con un número m de parámetros se realiza un proceso de filtrado y selección para obtener un conjunto de n parámetros donde $n < m$.

En general, la selección de parámetros se define tal que dado un conjunto de parámetros $F = \{F_i | i = 1, ..., N\}$, encontrar un subconjunto $F_M = \{F_{i1}, F_{i2}, ..., F_{iM}\}$ tal que $M < N$ y se optimice una función objetivo, por ejemplo clasificación exitosa.

El principal objetivo de la selección de parámetros es obtener un nuevo conjunto de datos tal que la complejidad de la descripción del problema y del sistema de clasificación sean reducidas, haya mejor generalización del problema y la posibilidad de overfitting sea menor. Hay varias técnicas ampliamente usadas para realizar selección de paráme-

[2]Esta parte, sin embargo, no es tan evidente. Si bien en numerosos trabajos se considera que los parámetros redundantes son perjudiciales y deben ser descartados, tal como en [62] [154] [197] [100], en [58] se mostró cómo la adición de parámetros redundantes puede ser útil en la clasificación debido a que se atenúa el efecto de los parámetros ruidosos y a que ciertos parámetros pueden ser redundantes en teoría, pero su reconstrucción a partir de otros parámetros es en la práctica muy difícil, de modo que son relevantes en todo caso

tros. La aproximación más simple es entrenar y validar diferentes clasificadores usando todos los probables subconjuntos de parámetros y el conjunto completo. Sin embargo, esta aproximación es usada en términos prácticos sólo cuando el conjunto de parámetros es pequeño, porque la combinación de parámetros en diversos subconjuntos se incrementa muy rápidamente cuando el conjunto es grande [3]. Otra posibilidad más ampliamente usada cuando el número de parámetros es alto es iterativamente incrementar o decrementar el tamaño del subconjunto de parámetros usado. Esto es llamado regresión por pasos (*stepwise regression*), y puede ser realizada sea iniciando con el conjunto completo de datos e iterativamente descartando parámetros (*backward elimination*), iniciando sin parámetros e iterativamente añadir parámetros (*forward selection*) o una combinación de las dos técnicas (*bidirectional elimination*) [43].

Una vez la metodología de decisión de subconjuntos de parámetros es definida, es necesario usar un algoritmo para evaluar el desempeño de la clasificación con el nuevo conjunto de datos. Si bien una métrica que se base simplemente en la precisión de clasificacion para cada subconjunto suena razonable, hay tres defectos en esta aproximación. El primero es que dado un clasificador suficientemente poderoso, es posible que las mejores precisiones de clasificación se obtengan cuando el subconjunto de parámetros aún tenga un buen número de parámetros innecesarios. Es decir, los errores del subconjunto son camuflados por el potente clasificador. El segundo inconveniente está dado por el bias de los datos, tal como se muestra en [136] [29]. Esto se refiere a que debido a que los datos usados son los mismos, los valores de regresión obtenidos usando metodología de *forward selection* son mejores que usando datos independientes. Este inconveniente puede ser atenuado si se usan conjuntos independientes en la selección de parámetros y la validación, pero puede ser complicado si el conjunto de datos completo es limitado. Por último, al usar como métrica la clasificación, el subconjunto final es aquél para el cual los errores de clasificación sean menores. Sin embargo, esto introduce un problema metodológico fundamental que rara vez es tenido en cuenta en la implementación de selección de parámetros y no lo encontramos referido en la bibliografía consultada: al usar el conjunto completo de datos para las pruebas de desempeño con diversos subconjuntos de parámetros, los algoritmos van a eliminar los parámetros que ocasionan que la clasificación empeore. Esto suena razonable, pero no hay garantía de que en realidad estos parámetros eliminados fueran perjudiciales en el universo de datos, sino sólo en el conjunto de datos utilizado. Mostraremos esto con un ejemplo simple, en la tabla 15.

Supongamos que tenemos un conjunto de datos con el clima registrado del día anterior y de exactamente un año atrás y se pretende diseñar un clasificador que prediga

[3]Por ejemplo, si el conjunto inicial tiene 20 parámetros, probar las combinaciones con 20, 19, 18, 17 y 16 parámetros requiere de 6196 entrenamientos y validaciones (que en realidad es mucho más grande, pues cada entrenamiento y validación tiene varias etapas en una metodología *leave-subjects-out*. En cambio si el conjunto inicial es de 10 parámetros, la prueba con combinaciones de 10, 9, 8, 7 y 6 parámetros requiere de únicamente 386 entrenamientos y validaciones. Naturalmente, en nuestro caso con 192 parámetros una aproximación como tal es imposible, ya que sólo probar con 191 y 190 parámetros requiere de 18256 entrenamientos y validaciones (nuevamente, en realidad mucho más, debido a la metodología *leave-subjects-out*), y con ello sólo se conseguiría eliminar máximo 2 parámetros del conjunto inicial

Tabla 15. Predicción del clima

	Clima hace un año	Clima ayer	Clima hoy
1	Caliente	Caliente	Caliente
2	Media	Caliente	Caliente
3	Fría	Fría	Fría
4	Caliente	Media	Fría
5	Media	Media	Caliente
6	Media	Fría	Fría

el clima de hoy. El dato 4 es un *outliner* en cuanto a que la temperatura hace un año es opuesta a la temperatura de hoy. Con este conjunto limitado, es probable que un algoritmo de selección de parámetros determine que el parámetro de clima hace un año sea un parámetro perjudicial, dado que este dato *outliner* puede ocasionar errores en la validación. Sin embargo, esto no quiere decir que en realidad este parámetro fuese inútil en la clasificación y probablemente si el conjunto de datos fuese más numeroso la selección de parámetros sería más adecuada usando como métrica el desempeño de clasificación.

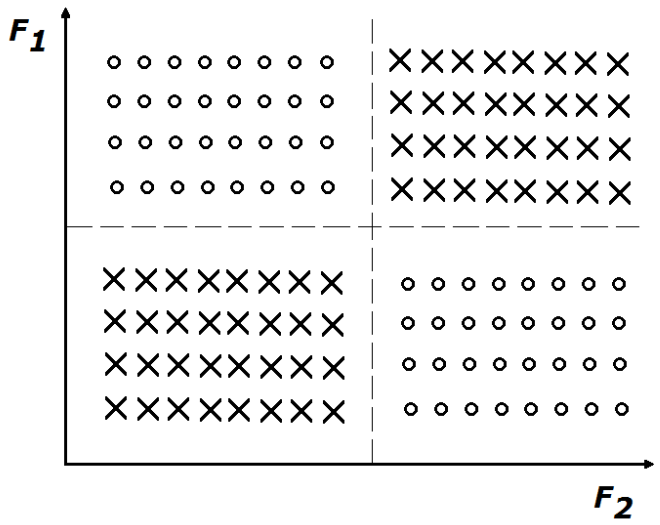
De manera análoga, el uso de los datos completos para la selección de parámetros puede ocasionar un efecto involuntario: el sistema selecciona justamente los parámetros más adecuados para clasificar con este conjunto de datos (*cherrypicking*), lo cual es una manera que eleva artificialmente los resultados de clasificación. De hecho, dado un número de parámetros suficientemente grande, así sean aleatorios, es posible encontrar siempre un subconjunto de parámetros con los cuales la clasificación mejore y, en teoría, incluso pueda ser clasificación perfecta, si se hace el proceso de selección de parámetros con el conjunto completo.

Dadas estas consideraciones, una alternativa es obtener una métrica que optimice una función de costo, por ejemplo una métrica cuyo valor se incremente en cuanto el subconjunto de parámetros sea más adecuado para la clasificación, pero cuyo valor disminuya si el número de parámetros se incrementa.

En [186] se mostró cómo la inclusión de parámeros que no son relevantes en el conjunto de parámetros conduce a una clasificación menos eficiente. La eliminación de parámetros irrelevantes puede parecer una tarea relativamente simple, pero la eliminación de parámetros redundantes es más compleja debido a que una técnica basada en la relevancia de los parámetros puede dar puntuación similar a distintos subconjuntos así tengan distinto número de parámetros redundantes. De hecho, algunos parámetros aparentemente irrelevantes pueden ser usados en conjunto, mientras que su desempeño individual es pobre. Esto es mostrado en un ejemplo simple de clasificación de 2 clases con 2 parámetros, en la figura 26.

En el ejemplo de la figura 26, los parámetros F_1 y F_2 proporcionan individualmente información nula acerca de las clases. Es decir, sin importar qué valor tengan F_1 o F_2 individualmente, la pertenencia a cualquiera de las dos clases es igualmente probable.

Figura 26. Aporte individual y conjunto de parámetros débiles



En consecuencia, si F_1 y F_2 pertenecen a un conjunto completo de parámetros F, es probable que en una metodología *forward selection* o *bidirectional elimination* estos parámetros no sean incluidos nunca en el conjunto de parámetros en ninguna iteración, por cuanto añadirlos individualmente no va a mejorar el resultado de clasificación. Debido a esto, el problema de selección de parámetros puede tornarse bastante complejo.

5.2.1. Tipos de técnicas de selección de parámetros

Las principales técnicas usadas para medir el desempeño de un subconjjnto de parámetros son las técnicas basadas en filtros, *wrapper-based* y técnicas embebidas:

- **Técnicas basadas en filtros:** Las técnicas basadas en filtros son independientes de los clasificadores. Los filtros diseñados miden los parámetros individuales y dependiendo de la relevancia obtenida, la dependencia entre parámetros y clases y otras métricas, algunos parámetros son eliminados y un subconjunto de parámetros es obtenido [94]. La principal desventaja de esta aproximación es que es posible que un subconjunto de parámetros tenga buen desempeño pese a que el desempeño individual de cada parámetro sea limitado (como en el ejemplo mostrado en la figura 26), de modo que la eliminación de parámetros puede conducir a una peor clasificación.

- **Técnicas *wrapper-based*:** Las técnicas *wrapper* se refieren a la evaluación de distintos subconjuntos de parámetros, midiendo su desempeño [83]. Generalmente la métrica incluye la tasa de clasificación y la complejidad del subconjunto de datos. Debido a esta metodología, son computacionalmente costosos y generalmente no se pueden usar con un conjunto numeroso de parámetros. Si bien estas técnicas incluyen el desempeño de la clasificación como métrica, con todos

los inconvenientes reseñados que esto representa, sus resultados generalmente son mejores que usando técnicas basadas en filtros. Sin embargo, las técnicas *wrapper* son muy limitadas en numerosos problemas reales, por ejemplo con un gran número de parámetros, porque en estos casos es difícil y extenso intentar definir un adecuado número de subconjuntos de parámetros que incluyan un adecuado rango de subconjuntos posibles viables.

- **Técnicas embebidas:** Los métodos embebidos también dependen de los clasificadores al igual que los métodos *wrapper*. Sin embargo, la búsqueda de parámetros es incluida en el diseño de la clasificación [87]. Como consecuencia, la adición o eliminación de parámetros del conjunto de datos es una parte integral del proceso.

Estas técnicas requieren de métodos para medir las características de los parámetros (técnicas basadas en filtros) o los clasificadores (técnicas *wrapper* y técnicas embebidas).

5.2.2. Métricas de los parámetros

En el libro *Statistical Pattern Recognition* [178], capítulo *Feature selection and extraction* hay una descripción detallada de estas métricas de evaluación. Presentaremos una breve reseña incluyendo observaciones del autor acerca de la utilidad, ventajas y desventajas de los métodos, con el fin de tener una idea inicial de los métodos usados en nuestro trabajo.

Las métricas de los parámetros se refieren a la evaluación del desempeño individual de cada parámetro, principalmente para eliminar ruido estadístico debido a parámetros irrelevantes y redundancia. Estas métricas son independientes de los clasificadores y los modelos, de manera que pueden tener baja capacidad de discriminación comparadas con otras métricas basadas en clasificadores, pero generalmente son de fácil implementación y a veces son la única opción por cuanto las métricas basadas en clasificadores pueden ser de difícil ejecución debido al tamaño y la complejidad de los datos.

- **Clasificación de los parámetros:** Cada parámetro obtiene un puntaje dependiendo de su desempeño de clasificación, sea individualmente o, si es posible, dentro de un subconjunto de parámetros. El principal inconveniente de esta métrica es que el desempeño individual de un parámetro no necesariamente implica que es útil o no dentro del contexto de un subconjunto de parámetros.

- **Información mutua y correlación:** La idea subyacente es que si un parámetro tiene información relevante, los puntajes de correlación e información mutua entre el parámeto y las clases debería representar esto. Si bien es una idea razonable, tiene dos inconvenientes. El primero es que el cálculo de información mutua y correlación con conjuntos de datos de alta dimensión es un cálculo extenso. El segundo es similar al problema de clasificación de parámetros. Por ejemplo, es posible que dados dos parámetros F_1 y F_2 con baja información mutua y correlación con las clases en un problema de 2-clases, mientras que hay un parámetro F_3 con

mayores puntajes en estas métricas. Sin embargo, es posible que el ensamble de F_1 y F_2 produzca una mejor clasificación global.

- **Distancia intraclase:** Esta métrica se refiere al cálculo de la distancia entre muestras para distintas clases. La idea es que si un parámetro sirve como discriminador entre clases, las distancias entre los miembros de la misma clase deberían ser relativamente pequeñas, en comparación con las distancias interclases. Si hay un parámetro cuya presencia incrementa las distancias intraclase, puede ser un indicador de que este parámetro no es relevante en la discriminación y está añadiendo ruido al sistema. Por otra parte, esto depende de la métrica usada para medir las distancias entre puntos y del tipo de ruido posible en los parámetros. Como vimos previamente, ciertas métricas no son útiles para medir espacios de alta dimensión y cierto tipo de ruido no necesariamente incrementa las distancias entre clases, de modo que no es posible determinar con este método si efectivamente corresponde a un parámetro inútil. Finalmente, es posible construir teóricamente infinitos conjuntos de datos en los cuales hay parámetros que incrementan la distancias intraclase pero paradójicamente son parámetros imprescindibles en la clasificación. Si bien son casos teóricos, en la práctica estos casos son frecuentes también, especialmente en conjuntos de datos no convexos.

- **Distancias probabilísticas:** Dadas las funciones de densidad de probabilidad condicional, esta medida evalúa las distancias probabilísticas. Las desventajas de esta metodología son que no es tan útil para variables continuas, el número de muestras debe ser suficientemente alto para determinar con cierta precisión las funciones de densidad de probabilidad condicional y en muchos casos los datos no se ajustan a funciones convencionales. Teniendo en cuenta que en nuestro caso cada parámetro no es un número sino es un vector de longitud considerable, este método no es aplicable, pues es imposible construir un modelo adecuado de los espacios n-dimensionales grandes producidos por estos vectores con un conjunto tan limitado de datos.

5.2.3. Métricas de los clasificadores

Estas métricas se basan principalmente en la precisión de clasificación dependiendo del subconjunto de parámetros usado. Como tal, son típicamente usadas con métodos *wrapper* y embebidos. Si bien en la literatura se usa comúnmente el desempeño de la clasificación como la mejor métrica, ha otras posibles métricas que pueden evaluar la utilidad de distintos subconjuntos de parámetros.

- **Tasa de clasificación:** Ésta es la métrica de clasificadores más directa. Después de todo, el objetivo de un clasificador es obtener el mejor desempeño de clasificación, de modo que una métrica que evalúe esto probablemente contribuya a obtener un buen subconjunto de parámetros. Sin embargo, tal como se reseñó previamente, medir únicamente la tasa de clasificación puede conducir a que ningún o

pocos parámetros sean eliminados tal que el clasificador usado sea suficientemente complejo y poderoso.

- **Métrica basada en aprendizaje:** En vez de basarse exclusivamente en la tasa de clasificación, esta métrica involucra la evaluación de los procesos de aprendizaje del sistema de clasificación cuando distintos subconjuntos de parámetros son usados. La evaluación de las curvas de aprendizaje con el eje horizontal representando el número de parámetros y el vertical la tasa de clasificación puede servir para determinar si hay un número insuficiente de parámetros (cuando el error de aprendizaje continúa disminuyendo al aumentar el número de parámetros, sin disminuir el error de validación -i.e. sin entrar a zona de *overfitting*-) o si hay más parámetros que los necesarios (el error de aprendizaje permanece relativamente constante al aumentar el número de parámetros). Los principales problemas de esta metodología son que los cálculos requeridos para cada evaluación son más extensos que para métodos basados en desempeño y las curvas de aprendizaje en función del número de parámetros no son muy confiables si corresponden a subconjuntos de parámetros muy distintos.

Cada una de las métricas reseñadas tiene su propia aproximación metodológica, de modo que sugerimos al lector interesado que se refiera a la bibliografía para obtener información más detallada, por cuanto la reseña descrita en este capítulo es sólo una breve introducción.

5.3. Metodología de selección de parámetros en un sistema basado en clasificadores débiles

En nuestro trabajo cada parámetro es tratado como un metaclasificador débil, debido a que la capacidad de clasificación de cada parámetro no es elevada, de modo que su utilidad es basada en la combinación de un elevado número de metaclasificadores débiles para construir un clasificador fuerte. Esto es razonable si se tiene en cuenta que nuestros parámetros TPOEM corresponden a zonas específicas del rostro, de manera que no se espera que la información proporcionada por un parámetro sea suficiente para tener una tasa de éxito razonable en el problema de clasificación de 7 clases. Dado esto, nuestra hipótesis inicial es que la evaluación de los valores dentro de cada uno de los 192 parámetros (teniendo en cuenta que cada parámetro es un vector, no un escalar) es un proceso imposible en términos de costo de cálculo, pero la evaluación del desempeño individual de cada parámetro puede conducir a resultados inadecuados [4]. Otra metodología basada en la evaluación de subconjuntos de parámetros puede ser más precisa,

[4]Esto ocurre porque cada parámetro aislado tiene pobre capacidad de clasificación. Al construir un clasificador por cada parámetro, pocos parámetros tienen tasa de clasificación superior a 20 % y la mayoría apenas entre 15 % y 17 %, que es apenas ligeramente por encima del valor esperado de un clasificador aleatorio de 14.14 %, de modo que la fortaleza de los parámetros no está en la capacidad aislada individual sino en su ensamble conjunto.

pero los requerimientos de cálculo necesarios para una evaluación extensiva hacen imposible implementar esta opción. Dado esto, nuestra aproximación fue usar cada vector TPOEM como un metaclasificador débil y tratar de determinar una metodología viable para seleccionar un subconjunto de vectores de menor tamaño, descartando los vectores inútiles y perjudiciales del sistema.

En las pruebas preliminares de clasificación mostradas en el capítulo 4 inicialmente se asignó una ponderación igual a cada vector en el problema de clasificación. Posteriormente, dado que se espera que la contribución de cada vector sea diferente, se usó una métrica para determinar ponderación individual de cada vector TPOEM. Sin embargo, se espera que incluso algunos vectores sean irrelevantes y posiblemente ruidosos, de modo que eventualmente sea mejor eliminarlos del conjunto de datos (o, mejor todavía, no calcularlos en primera instancia, reduciendo además costos de cálculo y memoria). La eliminación de parámetros aparentemente innecesarios en un sistema de clasificación de 2-clases es trivial, pero en nuestros caso de 6 y 7 clases el problema es un poco más complejo, pues algunos metaclasificadores pueden ser irrelevantes en la discriminación de una o más clases, pero importantes en la discriminación de al menos una de las clases, de modo que el tratamiento de la selección de parámetros debe ser cuidadoso.

Si bien en orden de presentación este capítulo es previo al capítulo dedicado a la implementación y validación de los sistemas de clasificación, el trabajo no sigue el mismo orden cronológico. Inicialmente se hicieron pruebas preliminares de selección de parámetros, los subconjuntos fueron probados usando los primeros sistemas de clasificación implementados y posteriormente los sistemas de clasificación fueron probados y validados. Dado que los sistemas de clasificación fueron modificados y mejorados en numerosas ocasiones, se realizaron etapas de selección de parámetros para cada sistema de clasificación implementado. Con esto se consiguió sucesivamente reducir el número de parámetros finalmente usado. Es decir, en ocasiones ciertos parámetros que eran necesarios de acuerdo con un sistema de validación usando un tipo de clasificador, posteriormente probaron ser innecesarios con sistemas de clasificación más sofisticados. El proceso de selección de parámetros fue realimentado permanentemente con el trabajo paralelo de desarrollo y validación de los clasificadores.

Nuestra aproximación fue una búsqueda secuencial del subconjunto de parámetros que optimizaba la clasificación dadas ciertas restricciones. Primero, se determinó la capacidad individual de cada metaclasificador usando clasificación 1 vs. 1 por cada metaclasificador (es decir, para cada vector TPOEM se construyó el conjunto de clasificadores 1 vs. 1 correspondientes a expresión vs. expresión y expresión vs. neutral [5]). Esta parte de la evaluación es delicada, por cuanto un metaclasificador cuya precisión en todos los concursos 1 vs. 1 salvo uno sea 50 % (i.e. igual que un clasificador aleatorio), pero en el concurso restante sea 100 %, tiene una clasificación global de apenas

[5]En total, 21 clasificadores por vector TPOEM, para un total de 4032 clasificadores por los 192 vectores TPOEM. Teniendo en cuenta que los clasificadores son construidos y validados metodología LSO con 10 % de los individuos por fuera en cada validación, en total se debe hacer 40320 validaciones para probar el conjunto completo y una fracción proporcional para probar un subconjunto de parámetros, lo cual da una idea de la dificultad de costo involucrado de estas pruebas

52.38 %, lo cual es apenas muy levemente superior al resultado de clasificación aleatoria. Sin embargo, este metaclasificador particular es muy bueno en uno de los concursos, de modo que es un discriminador confiable de dos clases. Si bien éste es un ejemplo extremo, en la práctica encontramos un buen número de metaclasificadores con desempeños relativamente bajos en muchos de los concursos entre clase, pero relativamente altos en otros muy específicos, lo cual es razonable si se analiza la naturaleza de los datos: un metaclasificador situado en una región espacial próxima a la boca difícilmente puede discriminar entre expresiones como ira, neutral y, en muchos casos, tristeza, pero es un discriminador eficiente de otras expresiones como sorpresa, alegría, disgusto y miedo. Como consecuencia, como métrica usamos dos puntajes de cada metaclasificador: su puntaje global y su puntaje máximo en los concursos 1 vs. 1.

Por último, teniendo en cuenta que hacer un protocolo completo de búsqueda exhaustiva *top-down* o *bottom-up* es prácticamente imposible debido al costo computacional de cada iteración, decidimos incluir información adicional al proceso de forma manual. Intuitivamente podemos determinar *a priori* cuáles regiones espaciales tienen alta probabilidad de ser relevantes en la caracterización facial. Para esto podemos usar sentido común (es razonable asumir que la boca, el ceño y los ojos son descriptores fundamentales) y los puntajes de celdas espaciales obtenidos en nuestras pruebas de ponderación de parámetros en el capítulo 3. De esta forma, para el subconjunto inicial se seleccionaron los metaclasificadores que corresponden a ojos, cejas, boca y ceño y descartamos areas alrededor del mentón, pómulos y regiones periféricas del rostro cuya contribución no era evidente.

Una vez el subconjunto de parámetros inicial fue determinado, definimos la función de criterio de parámetros como la precisión de clasificación de cada expresión individualmente. Hay diversas maneras de medir el desempeño de un subconjunto de parámetro, por ejemplo las metodologías usadas en [155, 56, 63]. En nuestro caso decidimos usar el criterio de clasificación por cuanto es una medida directa del desempeño del sistema de clasificación, manteniendo costos de cálculo bajos. Adicionalmente, los criterios que se basan en evaluación estadística de los parámetros individuales, por ejemplo con medidas de correlación o modelos Bayesianos, no son adecuados en parámetros que no son relativamente fáciles de modelar, como nuestros parámetros que son vectores de alta dimensión, en comparación a parámetros escalares más convencionales, cuyo modelamiento es más simple. Al hacer esto individualemnte, intentamos garantizar que un parámetro cuyo poder de clasificación es bajo para algunas expresiones pero alto para una o dos es de todas formas incluido. En la ecuación 5.1 se muestra el puntaje por parámetro.

$$J(X_i) = \max(S_c(X_i)), \qquad c = 1, ..., C \tag{5.1}$$

donde c corresponde a las clases del problema multiclase y $S_c(\cdot)$ es el puntaje de la clase c.

Posteriormente se hizo proceso completo de entrenamiento y validación usando LSO, obteniendo el puntaje promedio por cada validación (en total se hacen 10 validaciones

por iteración). El siguiente paso en la mayor parte de algoritmos de búsqueda secuencial es iterativamente incluir o descartar uno o más parámetros tal que la función objetivo sea optimizada (máximo incremento en métodos *bottom-up* y mínimo decremento en métodos *top-down*). Sin embargo, no procedimos de esta manera debido a varias consideraciones. En primer lugar, el número de parámetros es considerablemente alto, de modo que cada iteración implicaría el entrenamiento y validación de un elevado número de distintos subconjuntos de parámetros [6]. En segundo lugar, necesitamos garantizar la validez de clasificación para cada una de las 7 clases, así que el tamaño del problema es más elevado que en una aproximación de 2 clases. En cambio, usamos los puntajes individuales por parámetro obtenidos previamente para el conjunto completo. Para la primera iteración con un parámetro adicional, sólo hicimos 4 pruebas, añadiendo individualmente cada uno de los 4 parámetros no pertenecientes al subconjunto actual cuyos puntajes individuales sean más altos. Entendemos que esta aproximación novedosa tiene una importante desventaja, pues el hecho de que un parámetro tenga un puntaje individual relativamente alto no significa que tiene una contribución de clasificación importante. Esto sucede porque el parámetro puede tener una alta correlación con uno o varios de los parámetros ya pertenecientes al subconjunto, de modo que su inclusión no necesariamente aporta información. El efecto contrario también es posible en una iteración de descarte de parámetros: un puntaje individual bajo no significa forzosamente que el parámetro sea irrelevante, debido a que puede interactuar con otros parámetros para producir una clasificación más fuerte. Sin embargo, desarrollar una etapa de validación y entrenamiento completa con todos los posibles subconjuntos de parámetros es irrealizable, de modo que es un compromiso necesario y las posibles desventajas son atenuadas por la metodología usada.

El puntaje por iteración está dado por la estimación de error de clasificación en cada etapa, como en la ecuación 5.2.

$$J(X) = \frac{1}{n} \sum_{nf=1}^{n} (1 - CE(X_{nf})) \tag{5.2}$$

donde n es el número de pliegues de la validación LSO, nf es el paso de validación, $CE(X_{nf})$ es el error de clasificación en la validación nf.

A continuación se realiza una etapa de descarte de parámetros, usando una aproximación equivalente: los 4 parámetros con desempeño individual más bajo son seleccionados y se realiza etapa de entrenamiento y validación con el subconjunto actual excluyendo cada uno de estos parámetros. El nuevo subconjunto que obtiene mejor puntaje es seleccionado y eventualmente un parámetro es descartado (salvo cuando

[6] Por ejemplo, un problema con 40 parámetros usando *bottom-up* para *Sequential Feature Selection* (SFS) [134] con un parámetro incluido en cada iteración. La primera iteración tiene 40 etapas de entrenamiento y validación; la segunda iteración tiene 39 y así sucesivamente. Naturalmente, si el número de parámetros añadidos por cada iteración es mayor, el número de entrenamientos y validaciones se incrementa notablemente, por cuanto los posibles subconjuntos con distintas combinaciones de parámetros son mucho más numerosos (para 2 parámetros por iteración, sería 780 en la primera iteración, 741 en la segunda y así sucesivamente).

descartar un parámetro implica disminuir el desempeño del sistema, que es frecuente en las primeras iteraciones). El protocolo es parecido al convencional de búsqueda secuencial y sus implementaciones y mejoras más recientes [117].

El proceso es repetido iterativamente hasta que alguna condición se cumpla. Cada vez que un parámetro es añadido, obtiene una señal (*flag*), de modo que no es removido en esa misma iteración sin importar los resultados, para evitar bucles infinitos. Sin embargo, si parámetros que fueron añadidos en iteraciones previas empiezan a ser descartados y la función objetivo no se incrementa, es indicación de que probablemente no hay parámetros elegibles en el conjunto remanente cuya inclusión pueda mejorar el desempeño del sistema. Si éste es el caso, el proceso se detiene y se elige el último subconjunto como subconjunto final de parámetros. De esta forma el proceso tiene similitud con los métodos de búsqueda flotante [135, 153], pero sin muchas de sus desventajas: i. No requiere de búsqueda con todos los posibles subconjuntos de parámetros por cada iteración, con lo cual el costo de cálculo es reducido ii. No evalúa únicamente 1 subconjunto nuevo por etapa de inclusión o descarte de parámetros, de modo que no entra en bucles irrecuperables, iii. Cada parámetro tiene una ponderación previa, de modo que el algoritmo tiene información adicional acerca de la posible relevancia de un parámetro candidato. Adicionalmente, los métodos de búsqueda flotante convencionales son suficientes en problemas de 2 clases, por cuanto descartar un parámetro que no sea útil en la discriminación de las 2 clases garantiza su inutilidad. En nuestro problema multiclase esto no es necesariamente cierto, por cuanto un parámetro puede tener importancia muy específica (i.e. discriminando entre dos clases, así su desempeño en la discriminación de las demás sea bajo).

El proceso, que denominamos SFA-WM (*Sequential Feature Analysis for Weak Metaclassifiers extraction* es más fácilmente explicado en el pseudoalgoritmo 3:

En el pseudoalgoritmo $X_{k,i}$ representa los K parámetros para la muestra i del conjunto de datos, X_k^{tr} y X_k^{val} son los conjuntos disjuntos de entrenamiento y validación respectivamente, usando LSO, Y_k es el subconjunto de parámetros elegibles, $S(\cdot)$ es la función de evaluación de desempeño individual de los parámetros y $J(\cdot)$ es el desempeño de clasificación de un subconjunto de parámetros.

5.4. Resultados

El conjunto inicial de parámetros fue obtenido manualmente, incluyendo áreas faciales que consideramos importantes en el reconocimiento de la expresión facial. Para esto usamos datos obtenidos de ponderación de parámetros en el capítulo 3 y zonas faciales que intuitivamente aportan información de expresión facial. Posteriormente se inició el algoritmo mostrado previamente. Tal como se esperaba, en las primeras iteraciones del algoritmo de búsqueda de parámetros no se eliminó ningún parámetro, por cuanto el subconjunto inicial es razonablemente relevante para la clasificación de la expresión facial. Esto ahorra un considerable costo de cálculo comparado con el tiempo de procesamiento si el algoritmo hubiese iniciado con un conjunto vacío de parámetros.

Algorithm 3 Pseudoalgoritmo para extracción de parámetros por Sequential Feature Analysis for Weak Metaclassifiers

procedure ENTRADA
  $X_k = X_0$ ▷ X_0 es inicializado manualmente
  $Y_k = U - X_k$
  $X_{k,i} = \{x_1, ..., x_N\}$
while *No hay condiciones de parada* **do**
  procedure GENERACIÓN DE DATOS LSO
    $X_{k,i}$ es separado en conjuntos disjuntos de entrenamiento y validación LSO
  procedure ENTRENAMIENTO
    Entrenar el sistema de clasificación usando $X_{k,i}^{tr}$
  procedure VALIDACIÓN
    La tasa de clasificación es obtenida usando los subconjuntos de validación
    $W_k = Y_k$; $b = \Phi$; $w = \Phi$; $V_k = X_k^{val}$
    for i=1 **to** 4 **do** ▷ El valor 4 puede ser cambiado de acuerdo al problema
      $t = \arg\max_{c \in W_k} S(c)$ ▷ $S(\cdot)$ es el puntaje individual por parámetro
      $b = b + \{t\}$; $W_k = W_k - \{t\}$
      $t = \arg\min_{c \in V_k} S(c)$
      $w = w + \{t\}$; $V_k = V_k - \{t\}$
      b and w contiene los 4 parámetros más y menos relevantes respectivamente
  procedure ADICIÓN Y ELIMINACIÓN DE PARÁMETROS
    $y = \arg\max_{b} J(X_k^{val} \cup \{b\})$ ▷ $J(\cdot)$ es la tasa de clasificación de acuerdo al conjunto de parámetros
    $X_k^{val} = X_k^{val} \cup \{y\}$
    $x = \arg\max_{w} J(X_k^{val} - \{w\})$
    $X_k^{val} = X_k^{val} - \{x\}$
  procedure CONDICIONES DE PARADA
    La tasa global de clasificación no mejora luego de varias iteraciones
    Los mismos parámetros son recurrentemente elegidos
end

Posteriormente, tras algunas iteraciones, el algoritmo de búsqueda empezó a incluir y eliminar parámetros en cada iteración, lo que corresponde al principal objetivo de esta metodología de búsqueda combinada. En este punto la relativa simplicidad del sistema de clasificación probó ser una fortaleza. Dado un sistema muy complejo de clasificación, es posible que la introducción de nuevos parámetros no necesariamente reduzca el desempeño del sistema si estos nuevos parámetros son redundantes pero no forzosamente ruidosos, de modo que no hay descarte de parámetros en estas iteraciones. Éste es el motivo por el cual es común que las funciones de puntuación de subconjuntos de parámetros incluyan penalización cuando el número de parámetros se incremente, de modo que un subconjunto de parámetros que tenga desempeño marginalmente mejor que otro pero a costa de una cantidad elevada de parámetros pueda tener un menor puntaje.

En nuestro caso esta última estimación de función de penalización probó no ser necesaria, por cuanto la métrica de clasificación simple por distancia chi-cuadrado hace que sea imposible que el sistema de clasificación aprenda muestras *outliners*, de manera que los parámetros innecesarios son invariablemente excluidos [7]. Dada la simplicidad de clasificación, al incluir un nuevo parámetro que no tenga información relevante para la clasificación global o al menos la discriminación de una o más clases, es improbable que ese parámetro sea incluido permanentemente en el subconjunto de parámetros final. Si esta situación se repite recurrentemente, el algoritmo se detiene, pues considera que no hay más parámetros relevantes en el subconjunto remanente.

Por otra parte, esta metodología sufre de un importante inconveniente. En un problema de clasificación de 2-clases, la arquitectura propuesta es suficientemente fuerte para medir indirectamente la información mutua entre algunos parámetros. Esto es, si en el subconjunto de parámetros elegidos hay uno o más parámetros que proporcionan la misma información que un parámetro candidato, las pruebas con el subconjunto que incluya el nuevo parámetro probablemente no muestran mejoría de desempeño [8]. Sin embargo, en un problema de clasificación de múltiples clases no ocurre lo mismo. Por ejemplo, consideremos un subconjunto de parámetros con 10 parámetros en un problema de clasificación de 3 clases. 5 de estos parámetros (parámetros 1 a 5) proporcionan información importante de discriminación de la clase III, mientras que los restantes parámetros (6 a 10) proporcionan información importante de discriminación de las clases I y II, asumiendo que la información de estos últimos 5 parámetros es irrelevante (ruido estadístico) para la discriminación de la clase III. Si ahora hay un

[7] Esto, sin embargo, no garantiza que haya parámetros potencialmente útiles que no sean eliminados, pero obtener esta garantía sin un sistema de clasificación poderoso que incluya información mutua entre parámetros y modelos estadísticos (que eventualmente son insuficientes, por la incapacidad de describir adecuadamente los espacios n-dimensionales altos con un conjunto limitado de datos), lo que implica, a su vez, un alto costo de cálculo por iteración.

[8] Esto no necesariamente es cierto si la información del nuevo parámetro se puede obtener a partir de un subconjunto de parámetros en el subconjunto actual pero de manera compleja, por ejemplo a partir de transformaciones no lineales. Sin embargo, esto implica que la adición de este nuevo parámetro es en realidad importante salvo que el sistema de clasificación sea suficientemente complejo para calcular estas relaciones no lineales, lo cual es probablemente imposible dada la alta dimensión de los datos.

nuevo parámetro candidato con información relevante de clasificación de la clase III, el puntaje del subconjunto que incluya este parámetro probablemente es superior sin importar que la información sea redundante. De esta manera, esta metodología puede eventualmente incluir parámetros redundantes en el subconjunto final, especialmente cuando estos parámetros son fuertemente especializados. Sin embargo, la inclusión de parámetros redundantes no es un problema grande mientras que el sistema no incluya parámetros que no aporten información útil sino sólo ruido estadístico, de manera que es un compromiso aceptable.

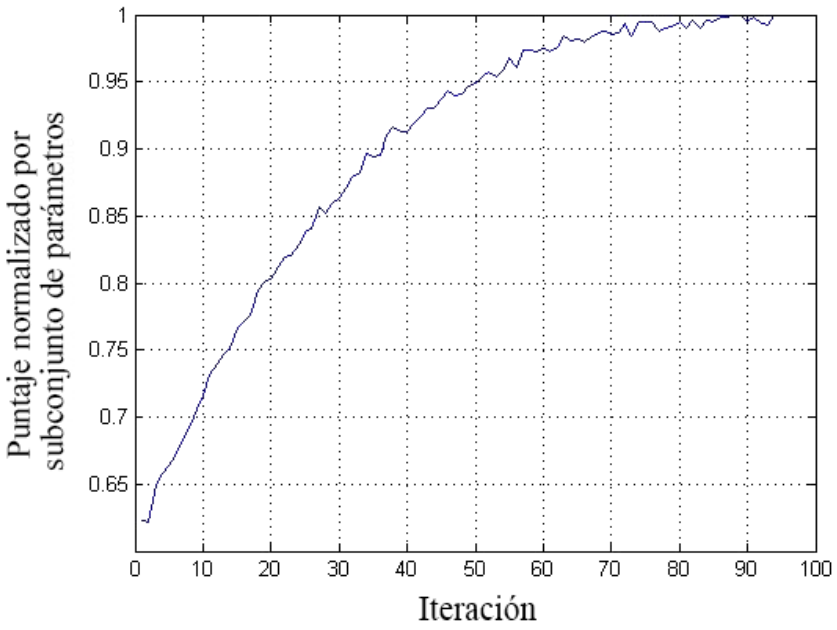
Este inconveniente es atenuado al incluir técnicas más poderosas de clasificación en el desarrollo de la búsqueda de parámetros, por ejemplo usando el sistema de clasificación ponderada *a posteriori* APCC que veremos en el capítulo 6, pero no era realizable en esta etapa del trabajo por cuanto el costo computacional de entrenamiento y validación de múltiples *folds* por subconjunto de parámetros elegibles en cada iteración es un proceso altamente demandante. Adicionalmente, las últimas pruebas realizadas para este trabajo, ya con algoritmos sofisticados de clasificación, mostraron alto desempeño con los parámetros seleccionados en esta etapa del trabajo, de manera que consideramos que el objetivo de reducir el costo de extracción de parámetros fue conseguido satisfactoriamente.

En la figura 27 se muestra el puntaje normalizado contra el número de iteraciones. Si bien uno de los criterios de detención del algoritmo de búsqueda de parámetros es cuando recurrentemente no se consigan nuevos subconjuntos de parámetros que mejoren consistentemente el puntaje en iteraciones sucesivas, decidimos extender manualmente la prueba por unas cuantas iteraciones más, con el fin de determinar el comportamiento de la curva luego de la hipotética detención.

La curva no es muy suave en ciertos puntos, especialmente a partir de un determinado número de iteraciones. Esto es razonable, por cuanto la idea del puntaje es obtener una estimación aproximada de la validez de un subconjunto de parámetros, pero debido a la naturaleza del algoritmo de validación por LSO, es posible que un subconjunto de parámetros que en realidad sea ligeramente superior que otro subconjunto, obtenga un puntaje levemente inferior. Esto muestra, sin embargo, que un criterio de detención menos riguroso, tal como detención inmediata cuando en una iteración el puntaje disminuya, puede ser perjudicial para la búsqueda de parámetros, pues haría que el algoritmo se detenga prematuramente. En la búsqueda cuyo puntaje por iteración es mostrado en la figura, el criterio de detención se alcanza en la iteración 85 o en la iteración 89 dependiendo de los valores de los parámetros de detención. En todo caso, se observa que a partir del rango 85-89 añadir iteraciones no mejora de manera distinguible el puntaje, de modo que aparentemente los criterios de detención son adecuados.

Según la curva, gradualmente el algoritmo hace las primeras iteraciones que añaden parámetros al subconjunto sin descartar ninguno -debido a que en las primeras iteraciones todos los parámetros en el subconjunto son fuertemente importantes en la representación de la expresión facial-. Posteriormente, cuando el subconjunto de parámetros elegible va perdiendo los más fuertes candidatos y el subconjunto de parámetros elegidos se va llenando de parámetros, el algoritmo empieza tanto a incluir como a

Figura 27. Puntaje normalizado contra número de iteraciones del algoritmo SFA-WC



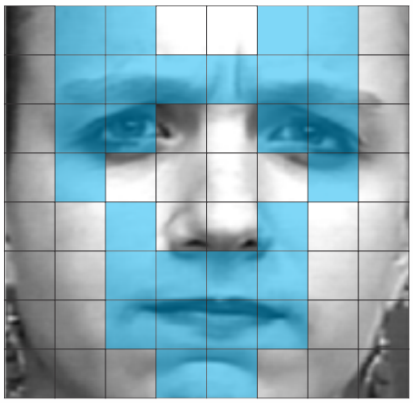
descartar parámetros en cada iteración. En algunas iteraciones el puntaje normalizado disminuye, sin que esto quiera decir forzosamente que el nuevo subconjunto sea peor que el subconjunto precedente. Esto se puede explicar porque el rango de confianza de este puntaje teniendo en cuenta el número de observaciones en el experimento (número de muestras validadas) es de aproximadamente 2.5 % en el rango de puntajes normalizados de 0.8 a 1. Debido a esto ocurren las fluctuaciones del puntaje normalizado. Esto es un problema inevitable por cuanto reducir el intervalo de confianza del puntaje requiere de un mayor número de observaciones, lo cual es imposible debido al tamaño limitado de la base de datos.

Una vez obtenido el subconjunto final de parámetros, hicimos una representación visual de la localización espacial de los parámetros elegidos. En la figura 28 se muestran en azul los parámetros elegidos para el conjunto con codificación TPOEM.

La figura muestra cómo se logró eliminar un considerable número de parámetros. Además nótese la similitud de la localización de los parámetros elegidos con las celdas espaciales de mayor relevancia encontradas en el capítulo 3. Esto hace que la etapa previa de obtención de parámetros TPOEM pueda ser reducida hasta aproximadamente un 45 % de su costo de cálculo inicial. De esta forma, es posible dedicar más tiempo de cálculo al proceso de clasificación sin implicar que el objetivo de cálculo de tiempo real sea incumplido.

Dada la localización espacial de los parámetros elegidos en el conjunto final, se puede argumentar que era posible obtener un subconjunto final muy parecido por selección

Figura 28. Parámetros espaciales TPOEM seleccionados por el protocolo SFW-WM



manual, debido a que los parámetros elegidos corresponden principalmente a zonas que por sentido común son relevantes en la caracterización de la expresión facial. Sin embargo, este análisis no considera que: i. Algunos parámetros corresponden a zonas que no son tan claras descriptoras de la expresión facial según una apreciación visual. ii. Nada garantiza *a priori* que los parámetros eliminados por el algoritmo realmente fueran irrelevantes para la caracterización de la expresión facial. Adicionalmente, el desarrollo de este algoritmo de búsqueda de parámetros no tiene como utilidad exclusiva la búsqueda en conjuntos iniciales de los cuales se pueda determinar por sentido común o visualización simple un subconjunto candidato que aparentemente sea eficiente descriptor. Esta implementación también es útil en la búsqueda general de parámetros, cuando el conjunto de datos no pueda ser inspeccionado manualmente ni haya manera lógica de determinar *a priori* la confiabilidad de cada parámetro.

Otra prueba adicional realizada fue medir la estabilidad y convergencia del algoritmo de búsqueda. Para ello obtuvimos el coeficiente Jaccard [75], tal como se define en la ecuación 5.3.

$$S(X) = \frac{1}{n} \sum_{nf=1}^{n} (1 - CE(X_{nf})) \tag{5.3}$$

La idea de esta métrica es medir el radio entre los parámetros comunes y los parámetros totales entre dos distintos subconjuntos de parámetros. Para ello el algoritmo fue ejecutado dos veces y obtuvimos el coeficiente Jaccard en cada iteración. Los resultados están mostrados en la tabla 16.

Tabla 16. Coeficiente Jaccard vs. número de iteracionens. Idénticos subconjuntos iniciales

Iteración	0	20	40	60	80	94
$S(f_i, f_j)$	1	0.897	0.946	0.962	0.981	0.991

El coeficiente Jaccard al iniciar el proceso es 1, puesto que los dos subconjuntos de parámetros son iguales. Después de algunas iteraciones el coeficiente Jaccard disminuye, hasta un mínimo de aproximadamente 0.89, que en todo caso es un valor notablemente alto, teniendo en cuenta que si, por ejemplo, luego de 20 iteraciones la primera búsqueda añade 16 parámetros en común con la segunda búsqueda, el coeficiente Jaccard es de 0.9. Adicionalmente, el coeficiente Jaccard continúa creciendo con más iteraciones, hasta un valor cercano a 1 luego de 94 iteraciones, lo cual implica que pese a las posibles discrepancias en las etapas iniciales de búsqueda, el algoritmo converge en dos búsquedas distintas a subconjuntos finales casi idénticos.

Adicionalmente, realizamos una prueba simplificada de dos búsquedas distintas con subconjuntos disjuntos, con semillas iniciales de 54 parámetros por subconjunto. Los resultados se muestran en la tabla 17.

Tabla 17. Coeficiente Jaccard vs. número de iteracionens. Subconjuntos iniciales disjuntos

Iteración	0	5	10	15	20	25
$S(f_i, f_j)$	0	0.056	0.101	0.158	0.209	0.248

La tabla muestra cómo el coeficiente Jaccard incrementa progresivamente con el número de iteraciones. Después de 25 iteraciones el coeficiente Jaccard es 0.248. Este número puede parecer bajo comparado con los valores de la tabla precedente. Sin embargo, nótese que el mejor escenario posible, considerando los dos subconjuntos disjuntos de 54 parámetros cada uno, es que las dos búsquedas incluyan los mismos 25 parámetros y eliminen 25 parámetros de los iniciales (disjuntos). En este caso el número de parámetros comunes luego de la iteración 25 es 25 y el número total de parámetros es de 83, para un coeficiente Jacccard final de 0.301 [9]. Adicionalmente, para que se cumpla este escenario se requiere que en todas las iteraciones se eliminen parámetros, que es poco probable teniendo en cuenta que esto requiere que en cada conjunto inicial casi la mitad de los parámetros sea considerablemente peor que el resto. Por último, se requiere que las dos búsquedas incluyan los mismos parámetros en 25 iteraciones y, tal como se observó en el experimento previo, en las primeras iteraciones no es razonable esperar este comportamiento.

En comparación con una aproximación convencional de búsqueda flotante con los mismos subconjuntos iniciales de 54 parámetros y asumiendo que el tamaño promedio de la búsqueda flotante en todo el proceso es de 90, que es una estimación muy generosa (por cuanto incluso si en las primeras 36 iteraciones de la búsqueda flotante se añaden parámetros sin descartar ninguno, el tamaño promedio en realidad sería 73, con lo cual

[9] En realidad hay otro escenario mejor, correspondiente a que una de las búsquedas incluya parámetros ya existentes en el otro subconjunto y viceversa, pero esto requiere que en los dos conjuntos aleatorios se encuentren parámetros muy poderosos, de modo que sean añadidos en las iteraciones. Esto es estadísticamente muy improbable y, de hecho, el valor esperado de este tipo de parámetros por subconjunto es de apenas 5.27.

el costo de cálculo es mayor), el número de validaciones requeridas en las 94 iteraciones es de 31.208, comparado con 752 en nuestra propuesta [10], de modo que el costo de cálculo se reduce notablemente.

A continuación realizamos algunas pruebas para comparar el desempeño de nuestra propuesta con otras alternativas sugeridas en la bibliografía. La primera comparación es con el trabajo en [185], usando *Fast Correlation-Based Filter* (FC-BF). En esta metodología el concepto denominado correlación predominante es introducido, con un método basado en filtros para intentar identificar parámetros relevantes y redundantes. La idea principal es usar la entropía condicional de las variables para obtener la ganancia de información. Las métricas basadas en entropía requieren valores discretos, de modo que para implementarlas discretizamos nuestros vectores TPOEM. Esta implementación es un reto complicado, porque en realidad no es posible discretizar fácilmente vectores de alta dimensión (longitud promedio 34) sin probablemente perder información considerable [11]. La discretización no puede ser realizada manualmente y cualquier técnica no supervisada puede conducir a la proyección de datos a un nuevo espacio n-dimensional en el cual la discriminación interclase es posiblemente perdida si los pliegues de alta dimensión son complejos y no convexos, como en nuestro caso. Consecuentemente, la discretización fue diseñada como un problema de reducción supervisada de dimensiones usando MCML.

Para evitar el error metodológico común de usar datos para reducción supervisada de dimensiones tanto en reducción como en validación, que es un error de prueba de hipótesis basada en datos debido a que la reducción supervisada es cierta clase de clasificación entrenada, decidimos separar los datos en pliegues. La metodología es LSO, con 10 pliegues con individuos disjuntos; 4 de ellos usados en la discretización, 5 en el entrenamiento y el restante en la validación, repitiendo el proceso 10 veces.

Una vez los subconjuntos discretizados fueron obtenidos, el protocolo para obtener los parámetros predominantes fue realizado. Nótese que el algoritmo es altamente demandante porque requiere del cálculo de la incertidumbre asimétrica [133] para todos los parámetros y clases. Más aún, el hecho de que un parámetro sea no predominante no implica que no sea relevante para el problema de clasificación, tal como mostramos previamente en el capítulo 3. Finalmente, los costos de cálculo de esta metodología son bastante altos, por cuanto en vez de evaluar el desempeño de algunos subconjuntos

[10] Para búsqueda flotante es $((256-90)\times 2)\times 94 = 31{,}208$. Los cálculos asumen en la metodología de búsqueda flotante que sólo se tiene en cuenta un parámetro para descartar/añadir por iteración; de otra forma el costo se incrementa exponencialmente. Para nuestra aproximación el costo es $4\times 2\times 94 = 752$

[11] Esta problema de discretización puede ser visto como un problema de reducción de dimensiones de una alta dimensión a una dimensión 1 discreta, que es sujeta a varios errores y, en un problema de múltiples clases tiene inconvenientes. Por ejemplo, en un problema de 2 clases una reducción a 1 dimensión puede hacer que los datos de la clase I queden embebidos tal que en general los datos de la muestra i pertenecen a la clase I si $y_i < 0$. Pero en un problema de múltiples clases embebido en una dimensión; incluso si los datos de la clase I pertenecen a una región, los de la clase II a la siguiente región del medio y los datos de la clase III quedan embebidos en la derecha de la clase II, tácitamente, para un sistema de clasificación, la clase III es más lejana de la clase I que la clase II, lo cual no es necesariamente correcto.

de parámetros por iteración, requiere del cálculo de la incertidumbre simétrica entre el parámetro candidato y el subconjunto actual de parámetros, lo cual es altamente demandante. Usamos el mismo subconjunto inicial que el de nuestra prueba inicial con nuestra propuesta, con 94 iteraciones. En realidad esta comparación no es justa, por cuanto el costo promedio por iteración con la metodología FC-BF es aproximadamente 7 veces más grande que el costo con nuestra aproximación, sin incluir el problema de reducción de dimensiones y discretización, que fue mucho más demandante (de hecho, el algoritmo requiere de más tiempo en esta etapa de reducción de dimensiones y discretización que en la búsqueda de parámetros).

Finalmente, implementamos la propuesta en [35], que es una búsqueda de parámetros basada en SVM multiclase, denominada SVM-MFFS (*SVM-Multiclass Forward Feature Selection*. Con el fin de usar una aproximación equivalente, construimos 21 clasificadores SVM ($\frac{k(k-1)}{2}$, donde k es el número de clases) usando los protocolos SVM-MMFS y SVM-BFFS descritos en el trabajo citado. Esto es una metodología delicada debido a varias razones. Primero, debido a que los parámetros son obtenidos basados en su desempeño de clasificación, el protocolo usado nuevamente fue LSO. Segundo, tal como encontramos en pruebas preliminares usando SVM 1 vs. 1 por parámetro para hacer clasificación multiclase (estas pruebas corresponden al capítulo 6), esto no significa que los parámetros con mejor desempeño en realidad proporcionen mejor representación de las clases, debido a que algunos parámetros obtienen altos puntajes fáciles para la clasificación de clases fáciles (en nuestro caso, alegría y sorpresa principalmente), así que los parámetros que tengan que lidiar con clasificación entre 2 clases típicamente problemáticas obtienen puntajes injustamente bajos.

A diferencia de las pruebas con la primera comparación, en este caso no ejecutamos un número fijo de iteraciones, sino que usamos las condiciones de parada de ejecución descritas en el artículo citado. Los costos de cálculo por iteración son más elevados debido a que se requiere calcular 21 clasificadores SVM-MFFS por iteración, el cálculo de los pesos y estabilización por parámetro (elevado, por cuanto debe realizarse en promedio 200 veces por iteración debido al tamaño del número de parámetros en el subconjunto elegible) y el cálculo del desempeño de clasificación por iteración. El uso de las condiciones de detención definidas por defecto en el trabajo representó otro problema, por cuanto en la mayoría de nuestras pruebas esto significó que el algoritmo se detuvo después de un número muy elevado de iteraciones. Es decir, el algoritmo no redujo sustancialmente el número de parámetros del conjunto total. Probablemente esto obedece a que en un entorno de parámetros débiles, definir los parámetros candidatos para ser incluidos en cada iteración según el protocolo de estabilidad definido en el documento no necesariamente sea un criterio adecuado. Esta técnica puede ser muy útil cuando hay algunos parámetros muy poderosos y algunos parámetros débiles en el conjunto de datos, pues los parámetros débiles son raramente incluidos en el conjunto final (de hecho, el ejemplo mostrado en el artículo citado muestra un número máximo de 10 iteraciones, y con 2 parámetros el ejemplo ya muestra 100 % de clasificación). En nuestro caso, sin embargo, con un conjunto diverso en el cual la mayor parte de parámetros son débiles y algunos irrelevantes, el algoritmo probó no ser suficientemente

buen discriminador entre algoritmos útiles e irrelevantes.

En la tabla 18 mostramos los resultados de tiempo de procesamiento promedio por iteración usando nuestra técnica propuesta,

Tabla 18. Costo de cálculo por iteración usando distintas téctnicas

Técnica	**Costo de cálculo por iteración**
SFA-WM	11.397s
FC-BF	3.104s
SVM-MFFS	87.822s

Si bien es posible que optimizando el código se consiga reducir el costo de cálculo de las pruebas, los resultados son concordantes con los señalados en la bibliografía, con costo de procesamiento por iteración de aproximadamente 450ms por iteración para FC-BF (en nuestro caso cada iteración en realidad es 10 iteraciones, debido a la realización de 10 validaciones LSO por iteración) y 41.92s por iteración para SVM-MFFS, pero con un conjunto de parámetros mucho más pequeño que el usado en nuestro trabajo.

Los resultados muestran que la reducción de dimensiones por FC-BF tiene considerable menor costo de cálculo por iteración que con nuestra técnica propuesta y la reducción por SVM-MFFS es la más costosa. Sin embargo, debe añadirse que en el caso de FC-BF los resultados mostrados son los de costo por iteración, sin incluir el costo de la reducción de los datos a 1 dimensión por MCML, cuyo costo total de procesamiento fue mucho mayor que el costo total del algoritmo de de extracción de parámetros.

Naturalmente, el desempeño de los algoritmos de búsqueda de parámetros no se debe medir en el tiempo de cálculo simplemente, sino en su capacidad de discriminación de extracción de parámetros sin afectar la tasa de clasificación. En la tabla 19 mostramos la clasificación por clase con los datos originales y reducidos por SFA-WM, FC-BF y SVM-MFFS, así como el número de parámetros por técnica.

Tabla 19. Clasificación y número de parámetros con los datos completos y reducidos por SFA-WM, FC-BF y SVM-MMFS

	Ira	**Dis.**	**Mie.**	**Ale.**	**Tri.**	**Sor.**	**Neu.**	**Num. parám.**
Total	93.52	91.58	89.33	100.00	87.00	94.75	90.64	256
SFA-WM	94.02	91.95	89.06	100.00	88.06	95.47	91.30	112
FC-BF	92.94	90.40	86.88	99.50	88.41	95.06	90.28	150
SVM-MMFS	94.87	90.23	85.94	100.00	84.06	96.30	90.64	163

Estos resultados muestran que en general se obtienen mejores resultados de clasificación con los métodos SFA-MC y SVM-MMFS, ligeramente superiores a los obtenidos con los datos completos. La mejoría respecto de los datos completos no es enorme, pero esto era esperado, por cuanto en este conjunto de parámetros el aporte individual de cada uno es muy reducido y no hay parámetros poderosos en los cuales se concentre

buena parte de la capacidad de discriminación. Adicionalmente, los resultados de clasificación con los datos completos ya son suficientemente buenos, de manera que esperar aumentar la tasa de clasificación significativamente no es realista. Sin embargo, se puede observar que nuestra técnica pudo reducir el número de parámetros de 256 a 112, mientras que la reducción con las otras técnicas fue menos notoria, y sin embargo la clasificación es similar/superior. Es decir, nuestra técnica fue apropiada para reducir sustancialmente el número de parámetros (y con ello, proporcionalmente el costo de extracción de los parámetros TPOEM) sin afectar el desempeño del sistema.

Estas pruebas muestran que nuestro algoritmo propuesto cumple la función de seleccionar parámetros adecuados del conjunto total de datos, preservando la capacidad de clasificación del sistema.

5.5. Conclusiones

En este trabajo mostramos cómo la búsqueda flotante de parámetros en un conjunto de datos en el cual cada parámetro es débil y el número de parámetros es elevado para un problema de clasificación de múltiples clases es un problema complejo. Sin embargo, nuestra metodología propuesta probó que el costo de cálculo puede ser reducido sin afectar el desempeño de clasificación y probablemente mejorándolo [12]. Nuestro algoritmo es aplicable tanto a problemas de dos clases como a problemas multiclase, por cuanto la ponderación individual de cada parámetro en la búsqueda iterativa depende principalmente de su capacidad de discriminación entre al menos dos de las clases, de manera que un parámetro muy especializado es incluido en el subconjunto de parámetros sin importar si su capacidad de discriminación global no es muy elevada. Esto es crítico en un problema multiclase, pues un algoritmo de búsqueda basado, por ejemplo, en correlación entre parámetros y clases, puede descartar parámetros por su relativa pobre correlación entre el parámetro y varias de las clases, siendo sin embargo un parámetro valioso en la discriminación de un subconjunto de las clases.

En la sección de resultados 5.4 mostramos que el algoritmo propuesto tiene buen desempeño y estabilidad. Los coeficientes Jaccard son altos y crecientes en dos búsquedas distintas que empiecen con la misma semilla de parámetros, lo que muestra que pese a la relativa alta variabilidad de resultados de clasificación y métricas de evaluación incluso con subconjuntos idénticos de parámetros [13], lo cual indica la alta convergencia y estabilidad de nuestra propuesta.

Nuestras pruebas incluyeron la búsqueda de parámetros con nuestra propuesta y con dos trabajos altamente relevantes en el estado del arte, por cuanto están especializados

[12] Los resultados mostraron ligeramente mejores resultados con datos reducidos que con datos completos, pero no se puede afirmar con certeza que la efectivamente hay mejoría de clasificación, por cuanto aún hay traslape de los rangos estadísticos de confianza 95 %, pero sin duda no hay reducción del desempeño y el costo de cálculo de extracción de parámetros TPOEM es notablemente reducido.

[13] Esto es ocasionado por la relativa alta variabilidad estadística de resultados al usar validación con un subconjunto pequeño de datos, pero es un problema inevitable teniendo en cuenta el conjunto limitado de los datos

en un caso en el problema de altas dimensiones (FC-BF) y en otro caso en el problema multiclase (SVM-MFFS). El algoritmo SFA-WM propuesto fue considerablemente superior en costo de cálculo a los dos algoritmos probados (mayor costo de cálculo por iteración que el FC-BF, pero el algoritmo FC-BF requiere de la proyección de cada parámetro a una dimensión y posterior discretización, que sumado al costo de la búsqueda de parámetros hace que el costo completo sea mayor que SFA-WC). Adicionalmente, las pruebas de desempeño mostraron que el conjunto reducido por SFA-WC tiene un desempeño aparentemente superior que usando los datos completos (estadísticamente no necesariamente superior, pero en todo caso no inferior), pese a que el conjunto de parámetros fue reducido en casi un 60 %.

La reducción de los parámetros TPOEM implica que la etapa de extracción de los códigos TPOEM puede ser considerablemente simplificada. El costo de extracción no es en realidad 60 % inferior, debido a que la codificación TPOEM está optimizada para usar datos de manera eficiente por reciclaje, pero sí se alcanza una reducción de alrededor del 55 % de costo de cálculo de esta etapa e incluso se puede reducir más drásticamente si se acepta un pequeño compromiso de reducción de clasificación.

Es posible mejorar las prestaciones de nuestra propuesta si se incluye una especialización en el problema multiclase. En vez de realizar una búsqueda global de parámetros, es posible obtener diferentes subconjuntos de parámetros especializados en la clasificación de cada clase. Para ello el proceso es muy parecido al protocolo mostrado, con la variación de que habría una búsqueda de parámetros independiente por cada clase y el criterio de evaluación de cada subconjunto por iteración no es el desempeño global sino el desempeño del subconjunto en la discriminación de la clase evaluada en un sistema 1 vs. 1. Esta mejora no fue implementada en nuestro problema porque no hay ninguna clase problemática, en el sentido de difícil clasificación comparada con los resultados del estado del arte y de la clasificación realizada por evaluadores humanos, entonces consideramos que la posible mejoría no sería importante. Sin embargo, en un problema multiclase con clases de difícil clasificación, esta implementación puede ayudar a mejorar la clasificación de estas clases complicadas, gracias a la especialización de subconjuntos de parámetros.

6. Sistema de clasificación para reconocimiento de expresión facial

6.1. Introducción

En este capítulo se mostrará el trabajo realizado con el fin de obtener la clasificación de la expresión facial a partir de los parámetros seleccionados y extraídos en las etapas precedentes. En general, una de las mayores dificultades encontradas es que si bien los parámetros TPOEM probaron ser descriptores eficientes para el reconocimiento de la expresión facial usando clasificadores muy simples con distancias Chi-cuadrado y distancias Euclidianas, especialmente al comparar directamente los resultados usando estas técnicas con parámetros convencionales LBP y derivados modernos, en el estado del arte se encuentra un considerable número de trabajos que, al usar sistemas de clasificación más sofisticados, dan un salto dramático de desempeño, alcanzando cotas de clasificación más elevadas. En este capítulo probamos diversas metodologías con complejidad y sofisticación crecientes, con el objetivo de determinar qué técnicas pueden proporcionar estos excelentes niveles de clasificación.

Uno de los principales tropiezos de intentar obtener resultados comparables o superiores a los de algunos trabajos del estado del arte fue relacionado con la metodología del proceso de validación. En este capítulo mostramos cómo el uso de metodología aleatoria cruzada basada en muestras, especialmente con múltiples muestras por individuo por expresión (o incluso todas las muestras de cada individuo por expresión en algunos trabajos) puede conducir a resultados de clasificación muy elevados, pero con problemas de generalización. Normalmente los inconvenientes de generalización son fácilmente encontrados en una metodología convencional aleatoria cruzada, por cuanto la clasificación se deteriora notablemente con las muestras de validación. No obstante, al usar múltiples muestras por individuo, es poco probable que haya muestras de validación sin muestras del mismo individuo y expresión en el conjunto de entrenamiento, de modo que el propósito fundamental de la validación cruzada se incumple. Para ello, hicimos un entrenamiento y validación con SVM-RBF, metodología aleatoria cruzada basada en muestras y el resultado de clasificación de 6 expresiones más neutral fue de 99.13 %. Si bien este resultado es casi perfecto, en realidad es producto de esta metodología, por cuanto con una metodología LSO (entendiéndose como *leave-subjects-out* y no como *leave-samples-out*), que garantiza que todas las muestras de los individuos pertenecientes a la validación son descartadas del entrenamiento, incluyendo las muestras de otras clases, los resultados son de alrededor de 94 %.

No obstante, en el trabajo de este capítulo conseguimos obtener clasificadores sofisticados bastante apropiados para este trabajo de reconocimiento de expresión facial, con resultados de clasificación y gran capacidad de generalización comparables con los del estado del arte incluso en condiciones muy desventajosas de metodología de validación. La capacidad de generalización es mostrada en el capítulo 7, donde hacemos pruebas de generalización de desempeño de clasificación con validación con otras bases de datos, con resultados promisorios.

En la sección 6.2 hacemos una descripción de algunas metodologías de clasificación típicamente usadas en este tipo de problemas, incluyendo funciones discriminantes lineales, máquinas de soporte vectorial y deep learning, que fueron usadas en algunas de nuestras pruebas.

En la sección 6.3 hacemos la descripción de la metodología, desarrollo y validación de distintas etapas de este trabajo. Inicialmente, la clasificación es realizada por sistemas simples basados en discriminantes lineales. Posteriormente implementamos una metodología de ponderación que denominamos APCC (*A Posteriori Confidence Classification*), cuyo objetivo, a diferencia de buena parte de las técnicas basadas en ponderación, no es establecer una importancia intrínseca a cada metaclasificador, sino otorgarle una confiabilidad dependiendo de su respuesta. Los resultados de clasificación fueron mucho mejores que con técnicas convencionales de ponderación. En esta sección también incluimos un análisis de los inconvenientes de los clasificadores en espacios de alta dimensión, que dificultan la obtención de un clasificador óptimo. Adicionalmente incluimos clasificación basada en SVM, con análisis de las dificultades y tropiezos de uso de esta metodología. Finalmente, añadimos información mutua entre clasificadores, que permite mejorar la clasificación global, pues el problema ya no es una suma de metaclasificadores individuales, sino un ensamble estadístico del aporte de todos ellos.

En esta misma sección añadimos la prueba previamente reseñada de entrenamiento y validación aleatoria cruzada con 10 pliegues, que nos permite fundamentar nuestra hipótesis de que los resultados de clasificación nuestros y los publicados en otros trabajos son altamente dependientes de la metodología de validación y que el uso de cierto tipo de metodologías puede ser un error de prueba de hipótesis que conduce a resultados muy elevados de clasificación, con pobre capacidad de generalización que no es evidente dado el mecanismo de validación.

Por último, dados los recientes desarrollos en la técnica *deep learning* de clasificación, decidimos implementar un clasificador basado en esta técnica. A diferencia del *deep learning* convencional usado para reconocimiento de patrones en imágenes, en el cual las entradas de los clasificadores son los valores directos de cada pixel, en nuestra metodología las entradas a las máquinas de *deep learning* son los resultados de estimación de metaclasificación SVM y lineal. En nuestra propuesta las máquinas *deep learning* tienen que trabajar con datos de dimensión reducida, producida por la estimación de metaclasificación previa, y su función es aprender jerarquías de profundidad creciente en cada capa, hasta describir la jerarquía superior, de expresión facial. Los resultados son levemente superiores que los obtenidos con nuestras metodologías previas, aunque la diferencia no es notable y es insignificante dado el rango de incertidumbre. No obstante, consideramos que esta metodología puede ser una alternativa viable para abordar el problema de reconocimiento de la expresión facial.

En la sección 6.4 hacemos comparación de nuestros resultados con los resultados obtenidos en trabajos similares con las bases de datos CK y CK+ para 6 y 7 expresiones. Esta comparación muestra el grado de éxito de los parámetros TPOEM y los sistemas de clasificación usados para el reconocimiento de la expresión facial.

Uno de los mayores retos del sistema es el reconocimiento de la expresión facial en

imágenes de baja resolución o usando códigos de longitud limitada para reducir costo de cálculo. En la sección 6.5 mostramos cómo los códigos TPOEM permiten trabajar con imágenes de resolución hasta 128×128 pixeles sin perjudicar la capacidad de clasificación.

En la sección 6.6 hacemos una reseña de las conclusiones relevantes extraídas del trabajo desarrollado en este capítulo, algunas observaciones generales y recomendaciones para trabajo futuro.

El aporte del trabajo de este capítulo al desarrollo de sistemas de clasificación para reconocimiento de expresión facial es la prueba de sistemas de clasificación con complejidad creciente, nuestro análisis de las dificultades de ciertos tipos de clasificadores en espacios de alta dimensión, el sistema de ponderación APCC para clasificación basada en respuesta y no en importancia predeterminada de cada clasificador, las pruebas que permiten entender por qué ciertas metodologías de validación folded pueden producir resultados cercanos a la clasificación perfecta y el desarrollo de fusión de clasificadores SVM con máquinas *deep learning* que permiten combinar los mejores atributos de cada técnica: simplicidad de reducción de datos de las SVM y capacidad de construcción jerárquica de alto nivel con las máquinas *deep learning*.

6.2. Fundamentación teórica y estado del arte

Un sistema de clasificación se refiere al procesamiento de datos con el fin de separar un conjunto de datos en distintas clases o *clusters*. El primer caso se refiere a la clasificación supervisada, en el cual el conjunto de datos es etiquetado *a priori* y el objetivo es discriminar la información del conjunto de datos. El segundo caso es la clasificación no supervisada, cuyo objetivo es organizar los datos de acuerdo a similitud.

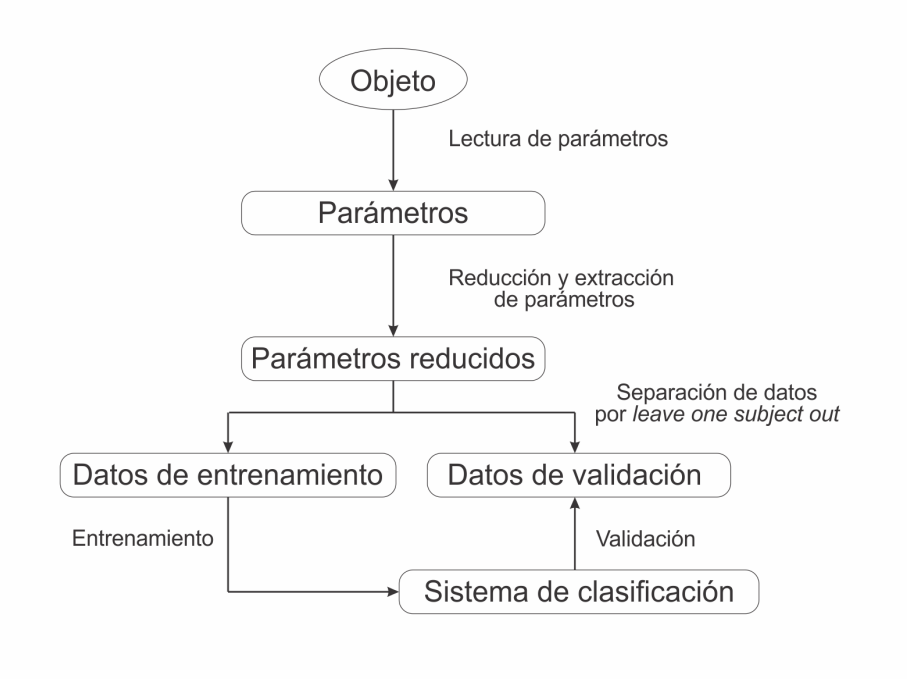
Con el fin de definir la descripción teórica y la metodología usada en este capítulo, usaremos la nomenclatura dada en [178]. Un parámetro o instancia es el vector p-dimensional $x = (x_1, x_2, ..., x_p)^T$ y cada x_i es una medida u observación de las características del objeto. Para clasificación supervisada, como en nuestro caso, hay C clases (6 expresiones faciales y una instancia neutra, para 7 clases) $c_1, ..., c_C$ asociadas con los parámetros en el conjunto de datos por la variable z tal que si $z = i$, la muestra pertenece a c_i.

El aprendizaje supervisado debe tener un buen desempeño con los datos del conjunto de entrenamiento, pero a su vez debe ofrecer una adecuada generalización con datos de validación o datos nuevos. Generalmente, un compromiso entre los dos objetivos debe ser alcanzado, por cuanto es una tarea relativamente fácil entrenar un sistema de clasificación que se ajuste a los datos del conjunto de entrenamiento con desempeño perfecto/casi perfecto, pero como resultado con pobre generalización.

6.2.1. Arquitectura de un sistema de clasificación

La arquitectura general de un sistema de clasificación es mostrada en la figura 29. La información es extraída del objeto, luego esta información es procesada para reducir

Figura 29. Esquema general del entrenamiento y validación de un sistema de clasificación



la dimensión o proyectar el conjunto de datos a una espacio n-dimensional con mejor representación y los parámetros del objeto son seleccionados. Se obtienen dos conjuntos disjuntos de entrenamiento y validación, en nuestro caso usando en general arquitectura LSO, para entrenar y validar el sistema. Una vez el sistema es entrenado, se puede usar con muestras nuevas desconocidas.

La etapa de clasificación debe proporcionar una salida basada en la probabilidad de emparejar los datos con el conjunto de vectores de parámetros. En el sistema básico template matching esto es logrado al buscar coincidencia exacta o al menos cercana similitud entre los vectores de parámetros y un conjunto de parámetros previamente establecido. Los sistemas de clasificación más generales incluyen variaciones estadísticasa debido a errores de medición, ruido estadístico, ruido de calibración y otras fuentes de variabilidad.

Las etapas de procesamiento pueden incluir reducción de dimensiones sea manualmente, cuando los vectores de parámetros son fácilmente analizables, o automáticamente, al usar una reducción de dimensión que proyecte los datos en una menor dimensión con mejor representación.

Hay numerosos métodos que pueden ser usados en la etapa de clasificación, tales como *template matching*, redes neuronales, lógica difusa, árboles Bayesianos, distancias métricas, LDA, estimadores de kernels, SVM y, más recientemente, deep learning. En la mayoría de los casos el clasificador óptimo es definido como aquél que proporcione menor rata de error cuando es probado con un conjunto de validación, pero ocasionalmente

se incluyen requerimientos adicionales, tales como la introducción de una función de castigo que penalice fuertemente ciertos errores de clasificación. En nuestro caso se usaron funciones de castigo con la idea de que es mejor clasificar incorrectamente una expresión como neutral que como perteneciente a otra expresión. Como tal, la tasa global de clasificación disminuye ligeramente, pero las prestaciones prácticas del sistema son mejores.

6.2.2. Retos en el diseño de un sistema de clasificación

Un sistema ideal de clasificación proporciona una adecuada representación con los parámetros del conjunto de datos tal que puede ser usado para predecir y extrapolar salidas dadas nuevas entradas. En la práctica hay diversos retos que hacen que el problema sea frecuentemente complejo.

- Problema de inferencia: Dado un vector de entrada x y su objetivo correspondiente t, la distribución de probabilidad conjunta $p(x, t)$ es una representación de la incertidumbre de las variables. Un sistema de clasificación debería ser capaz de producir un predictor que minimice el error de clasificación de nuevas entradas. Esto es, sin embargo, un problema complejo, pues en la práctica es probable que no exista una frontera de clasificación que produzca cero error de clasificación, así como posibles condiciones adicionales, tales como funciones de catigo, añaden restricciones al clasificador.

- Generalización del clasificador: El conjunto de entrenamiento en un sistema de clasificación es finito. Data tal limitación, es posible generar un clasificador cuya salida es 100 % correcta cuando es probado con los datos de entrenamiento siempre que no existan muestras idénticas pertenecientes a distinta clase. Esto es posible al construir un clasificador suficientemente complejo que transforme los datos a una menor dimensión tal que cada muestra es projectada a variables separables. Esto es, sin embargo, una aproximación metodológicamente errada, puesto que generalmente conduce a overfitting: el sistema se ajusta bien a datos conocidos, pero su desempeño es pobre con datos nuevos. Hay diversas técnicas para escoger y probar los modelos y métodos para validar los clasificadores, pero en la mayor parte de los casos el diseño de los clasificadores requiere de intervención humana cuidadosa para prevenir el problema. El principal inconveniente es que la solución óptima para un sistema de clasificación se basa en datos conocidos, correspondientes a los datos del conjunto de entrenamiento, de modo que el clasificador no sabe que un modelo de clasificación subóptimo puede producir mejores resultados globales. Algunas técnicas de entrenamiento incluyen realimentación de un subconjunto de validación, así que el clasificador es entrenado con un conjunto de entrenamiento, validado con el conjunto de validación y de acuerdo con los resultados hay realimentación que permite modificar el clasificador.

- Inconvenientes de los parámetros: Hay dos inconvenientes principales referentes a los parámetros usados para la descripción del problema. Por definición, los

parámetros en el conjunto de datos son intuitivamente válidos para discriminar entre clases, pero los valores y modelos de discriminación no son directamente conocidos. Debido a esto, los parámetros que se usan son elegidos previamente sin certeza acerca de su utilidad. Como consecuencia, es posible que algunos de los parámetros seleccionados no sean útiles para la discriminación entre clases y añadan ruido estadístico al sistema. La etapa de selección de parámetros ayuda a atenuar este inconveniente, pero en un problema de alta dimensión esto no necesariamente es suficiente y, por otra parte, el débil aporte de cada parámetro individual puede ocasionar que la etapa de selección de parámetros descarte parámetros que en realidad aportan información útil de clasificación.

- Definición de optimalidad: Hay diversos métodos para medir el desempeño de un sistema de clasificación. El más convencional es la tasa de error, que es básicamente una métrica para evaluar qué tan bien el sistema de clasificación se ajusta a un subconjunto de datos. La tasa de error a menudo conduce a overfitting, tal que el clasificador se ajusta al ruido del conjunto de datos en vez de a su estructura general inherente, lo cual es especialmente inconveniente si los datos son producidos usando condiciones específicas no necesariamente repetibles en escenarios reales. Debido a ello, hay otras definiciones de optimalidad, tales como capacidad de discriminación, tasa de error de Bayes y tasa de error real, así como técnicas de validación como validación cruzada, *bootstrap* y estimación *holdout*. Otro inconveniente de las tasas de error es que la métrica generalmente no considera si una muestra fue completamente clasificada erróneamente (i.e. el sistema tiene completa certeza de clasificación) o si fue una decisión disputada (i.e. el sistema tiene cierta idea de clasificación, pero no contundente), de modo que probabilidades posteriores pueden ser usadas para mejorar la definición de optimalidad. El principal inconveniente es que la etapa de validación en este tipo de trabajos generalmente es menos compleja que la etapa de clasificación, especialmente con datos no convexos de alta dimensión, mientras que la etapa de validación es típicamente vista como una evaluación simple de la clasificación con una matriz de confusión, pero el diseño de clasificación puede tener una evaluación de desempeño más apropiada para entender mejor la estructura de los datos, las causas de error y, consecuentemente, mejorar la clasificación.

Adicionalmente, en nuestro caso particular de clasificación de múltiples clases con un elevado número de parámetros de alta dimensión, hay otra suerte de inconvenientes adicionales. En principio, disponer de un buen número de parámetros es una característica deseada, y suena razonable que el uso de todos los parámetros, incluso si algunos de ellos son ruidosos o redundantes, debería conducir a una mejor tasa de clasificación, siendo la memoria y los costos de cálculo el único inconveniente. Sin embargo, éste no es el caso, tal como se mostró en [77]. Los parámetros ruidosos afectan el desempeño del sistema si son incluidos como variables de clasificación. Además, la elevada dimensión de los parámetros hace imposible generar un modelo preciso para representar los manifolds de alta dimensión con un conjunto de datos limitados, tales como los usados

típicamente en bases de datos de expresión facial. Como consecuencia de esto, los sistemas de clasificación que se basan en una representación probabilística o alguna otra clase de representación de *manifold* tienden a fallar o, cuando la metodología de validación es defectuosa, producir resultados de clasificación irrealista, con valores cercanos a la perfección pero metodología errada.

6.2.3. Métodos de sistemas de clasificación

A continuación haremos una reseña de los métodos de clasificación aplicados posteriormente en el desarrollo de nuestra propuesta de clasificación para este trabajo.

Funciones discriminantes lineales: Las funciones discriminantes lineales consisten en la transformación lineal de los datos en el vector x tal que $y(x, w)$ proporcione separación entre las clases C_k, siendo w los pesos de transformación y C_k las etiquetas de los datos en x [107]. La transformación $y(x, w)$ más básica es mostrada en la ecuación 6.1.

$$y(x) = f(w^T x + w_0) \tag{6.1}$$

donde $f(\cdot)$ es una función no lineal, pero dado $y(x)$ constante en las fronteras de decisión, $f(\cdot)$ es también constante, de modo que la trasformación en las fronteras de decisión es lineal. Así, la transformación es definida como un modelo lineal generalizado [106].

La función se puede modificar al usar una transformación tal que $\widetilde{w} = (w_0, w)$ y $\widetilde{x} = (1, x)$, de manera que la frontera de decisión pase por el origen del nuevo espacio dimensional, así que la clasificación es realizada según la equación 6.2.

$$\widetilde{w}^T \widetilde{x} \begin{cases} > 0 \\ < 0 \end{cases} \Rightarrow x \in \begin{cases} C_1 \\ C_2 \end{cases} \tag{6.2}$$

La tarea es obtener la discriminación lineal que minimice una función de error. Algunos criterios convencionalmente usados son: i. criterio de perceptrón, cuyo objectivo es encontrar la frontera que garantice la menor distancia de error entre las muestras y el hiperplano de separación [80], ii. criterio de mínimo cuadrado, en el cual la transformación $\widetilde{w}$ minimiza la suma de los errores cuadrados en la clasificación [145] y iii. criterio de discriminante Fisher lineal, que es una mejora sustancial sobre los anteriores criterios, y su objetivo es encontrar la frontera que maximice la separación entre las clases al obtener el radio entre las varianzas interclase y las varianzas intraclase [50].

Máquinas de soporte vectorial (SVM): En la ecuación 6.3 se muestra un método convencional de separación entre dos clases mediante transformación lineal.

$$y(w^T x + w_0) > 0 \tag{6.3}$$

El principal problema de esta aproximación es que la frontera puede estar en cualquier posición si bien la condición sea cumplida, lo cual puede conducir a fronteras muy próximas a puntos pertenecientes a alguna de las clases y esto puede ocasionar problemas de generalización. Al incorporar una nueva frontera mayor que 0, la ecuación se convierte en 6.4.

$$y(w^T x + w_0) > b \tag{6.4}$$

Si $b = 1$, los hiperplanos de separación son definidos por:

$$\begin{aligned} H_1 &: w^T x_i + w_0 \geq 1 \\ H_2 &: w^T x_i + w_0 \leq -1 \end{aligned} \tag{6.5}$$

La distancia entre los dos hiperplanos es $2/|w|$, así que al minimizar w la separación entre los datos por la frontera $g(x) = 0$ es optimizada, que es un problema con restricciones dado por 6.6.

$$y(w_i^T x_i + w_0) \geq 1 \tag{6.6}$$

La restricción es impuesta por la necesidad de preservar perfecta separación entre las clases C_1 y C_2. La forma primal de la función objetivo es dada por 6.7.

$$L_p = \frac{1}{2} w^T w - \sum_{i=1}^{n} \alpha_i \left(y_i \left(w^T x_i + w_0 \right) - 1 \right) \tag{6.7}$$

La solución es obtenida al encontrar los puntos de silla de L_p al diferenciar L_p respecto de w y w_0 e igualando a cero. Estas condiciones son añadidas a las condiciones de Kuhn-Tucker tal como se muestra en 6.8.

$$\frac{\delta L_p}{\delta w_0} = -\sum_{i=1}^{n} \alpha_i y_i = 0 \tag{6.8}$$

$$\frac{\delta L_p}{\delta w} = w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0 \tag{6.9}$$

$$y_i(x_i^T w + w_0) - 1 \geq 0 \tag{6.10}$$

$$\alpha_i \geq 0 \tag{6.11}$$

$$\alpha_i(y_i(x_i^T w + w_0) - 1) = 0 \tag{6.12}$$

La última condición implica que los puntos x_i que estén en los hiperplanos H_1 y H_2 tienen un valor $\alpha_i \geq 0$, mientras que los puntos x_i por fuera de los hiperplanos deben tener $\alpha_i = 0$[1].

[1]Si x_i está en los hiperplanos, $y_i(x_i^T w + w_0) - 1 = 0$. Como tal, $\alpha_i = 0$ si el punto está por fuera de los hiperplanos para satisfacer $\alpha_i(y_i(x_i^T w + w_0) - 1) = 0$.

Los valores de α son obtenidos por resolución cuadrática, el valor de w_o es derivado de una condición complementaria y la transformación w es obtenida por la ecuación 6.13.

$$w = \sum_{i \in SV} \alpha_i y_i x_i \tag{6.13}$$

donde SV son los vectores de soporte. La clasificación es dada por la ecuación 6.14.

$$\sum_{i \in SV} \alpha_i y_i x_i^T x - \frac{1}{n_{SV}} \sum_{i \in SV} \sum_{j \in SV} \alpha_i y_i x_i^T x_j + \frac{1}{n_{SV}} \sum_{i \in SV} y_i = \begin{cases} > 0 \\ < 0 \end{cases} \Rightarrow x \in \begin{cases} C_1 \\ C_2 \end{cases} \tag{6.14}$$

En la ecuación 6.14 se asume que los datos en C_1 y C_2 son perfectamente separables. Sin embargo, esto no es necesariamente cierto en aplicaciones prácticas, debido a *outliners* que traspasan los hiperplanos de separación o debido a muestras erróneamente etiquetadas. De cualquier manera, no es posible encontrar una SVM para separar los datos en este caso. Una solución es usar una transformación por kernel para proyectar los datos a un espacio de mayor dimensionalidad donde puedan ser separados. Sin embargo, esto puede conducir a *overfitting* del sistema de clasificación, así que no es necesariamente una aproximación adecuada. Como consecuencia, una mejor alternativa es diseñar una SVM que permita la clasificación errada de algunos datos usando funciones de castigo. De esta forma las ecuaciones en 6.5 se transforman a las ecuaciones en 6.15.

$$\begin{aligned} &H_1 : w^T x_i + w_0 \geq 1 - \xi_i \\ &H_2 : w^T x_i + w_0 \leq -1 + \xi_i \\ &\xi_i \geq 0, \;\; i = 1, ..., n \end{aligned} \tag{6.15}$$

Usando una función de castigo para penalizar las clasificaciones erradas, la nueva forma primal del Lagrangiano es dado por la ecuación 6.16.

$$L_p = \frac{1}{2} w^T w + C \sum_i \xi_i - \sum_{i=1}^{n} \alpha_i \left(y_i \left(w^T x_i + w_0 \right) - 1 + \xi_i \right) - \sum_{i=1}^{n} r_i \xi_i \tag{6.16}$$

El nuevo multiplicador r_i es introducido para prevenir valores negativos de ξ_i. La diferenciación respecto de w_0, w y ξ_i produce las ecuaciones mostradas en 6.17.

$$\begin{aligned}
\frac{\delta L_p}{\delta w_o} &= -\sum_{i=1}^{n} \alpha_i y_i = 0 \\
\frac{\delta L_p}{\delta w} &= w - \sum_{i=1}^{n} \alpha_i y_i x_i = 0 \\
\frac{\delta L_p}{\delta \xi_i} &= C - \alpha_i - r_i
\end{aligned} \tag{6.17}$$

***Deep learning*:** Deep learning es un desarrollo reciente en aprendizaje de máquina, cuyo objetivo fundamental es modelar abstracciones de alto nivel usando arquitecturas en cascada de distintas transformaciones lineales y no lineales [10]. En general, una de las limitaciones más importantes de las algoritmos de aprendizaje de máquina es su dificultad de aprender abstracciones complejas equivalentes a las usadas típicamente en el lenguaje y la visión humanos. En [41] se mostró que una de las características aún faltantes es la profundidad del aprendizaje. Con *deep learning* se intenta abordar esta falencia mediante el aprendizaje de jerarquías de alto nivel usando la combinación de jerarquías de más bajo nivel, de manera que estas jerarquías bajas sean aprendidas automáticamente por el sistema sin inferencia humana.

Otro inconveniente típicamente encontrado en los sistemas de aprendizaje es la maldición de la dimensionalidad (*curse of dimensionality*). Esto sucede porque los sistemas de aprendizaje generalmente intentan aprender usando una noción muy básica de similitud entre parámetros. Esto puede funcionar en bajas dimensiones, pero en altas dimensiones el número de muestras requerido para aprender las variaciones locales se incrementa exponencialmente, salvo en casos afortunados en los que *manifolds* complejos puedan ser separados relativamente bien por fronteras simples [163].

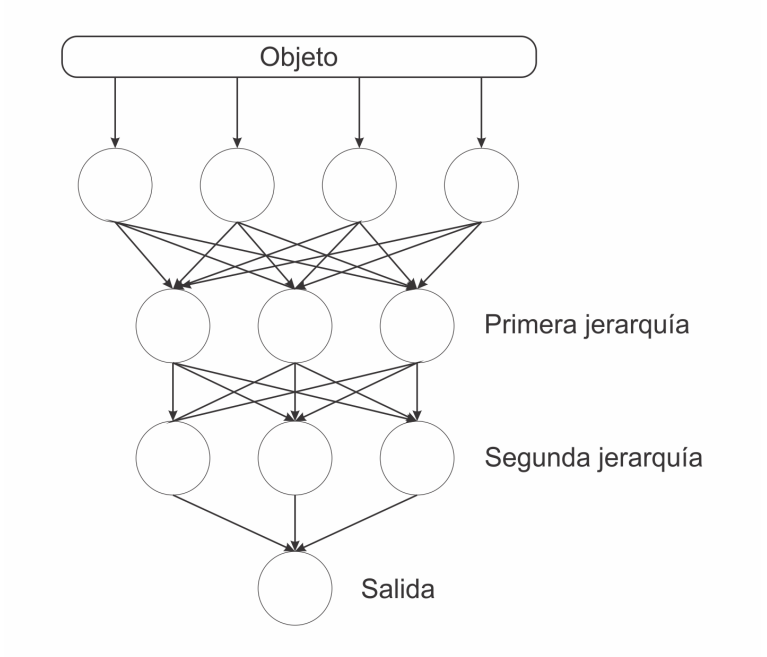
Por mucho tiempo hubo intentos infructuosos de entrenar arquitecturas de múltiples capas con adecuado nivel de generalización. *Deep belief* [70] representó el desarrollo que permitió la implementación de modelos estructurados con múltiples capas, mediante el uso de producto de expertos (en oposición a combinación de expertos), máquinas de Boltzmann y arquitectura especializada que permitieron resolver los problemas de pobre generalización de las máquinas de múltiples capas.

De esta manera se han resuelto los problemas de limitaciones teóricas de arquitecturas con número reducido de capas, los problemas de generalización no local de diversas técnicas de aprendizaje de máquina y las limitaciones de sistemas de aprendizaje basados en kernels o en árboles de decisión.

La arquitectura básica de una máquina *deep learning* es mostrada en la figura 30.

En la figura, la idea es que cada máquina de bajo nivel aprenda características de baja jerarquía. De esta forma, ciertos parámetros, tales como texturas locales en el caso de imágenes, son aprendidos por algunas de las máquinas de bajo nivel. Las máquinas intermedias aprenden a partir de la decisión de las máquinas de baja jerarquía. Así sucesivamente, cada capa debe interpretar patrones a partir de las salidas de las capas

Figura 30. Arquitectura de una máguina *deep learning*



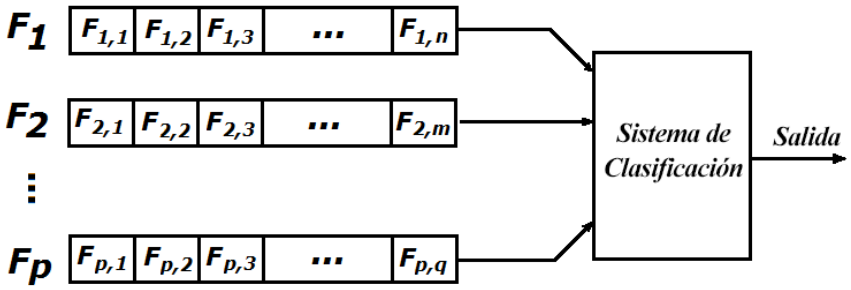
precedentes y de esta forma construir lenguaje de alto nivel. Por ejemplo, una imagen con una casa puede ser interpretada por características locales específicas desde el bajo nivel de los pixeles y progresivamente se construye el concepto abstracto de más alto nivel referente a casa.

6.3. Nuestra propuesta

En el capítulo 3 usamos una metodología muy simple de clasificación por template matching. El objetivo era probar la validez de los parámetros POEM, VPOEM y TPOEM como descriptores de las expresiones faciales. Si bien la comparación de los resultados obtenidos con nuestros parámetros TPOEM y otros métodos exitosos recientes que usan codificación basada en LBP fue exitosa, el uso de sistemas de clasificación más adecuados puede proporcionar mejoras notables en la precisión de clasificación. Si bien algunos clasificadores no obtuvieron mejoras sustanciales, en todos los casos las pruebas usando t-test mostraron una diferenciación estadística entre la metodología usada y los sistemas iniciales de clasificación.

Un elemento importante al realizar clasificación con un elevado número de parámetros es evaluar las implicaciones de acuerdo con la metodología de clasificación usada. Si bien en nuestro caso el número de parámetros no es mayor que el número de muestras, cada parámetro es un vector, de modo que la cantidad de posible información comparada con el número de muestras es muy elevada. En los sistemas de clasificación si $p >> N$, donde p es el número de parámetros y N es el número de observaciones, diversas técnicas de clasificación pueden conducir a overfitting y pobre generalización. En

Figura 31. Sistema de clasificación propuesto



nuestro caso parte del problema fue eliminado en la etapa de selección de parámetros, pero en todo caso la cantidad de posible información en nuestro conjunto de parámetros extraídos es suficientemente grande como para requrir de desarrollo cuidadoso de los sistemas de clasificación.

En esta sección mostraremos el desarrollo de nuestros sistemas de clasificación, incluyendo el diseño, las pruebas y la validación. Adicionalmente mostraremos un ejemplo de cómo el uso de una metodología errónea de clasificación y validación puede conducir a aparente tasa de clasificación elevada, pero en realidad los resultados son productos de una aproximación defectuosa al sistema de clasificación y un error metodológico de prueba de hipótesis típicamente encontrado en trabajos de esta naturaleza.

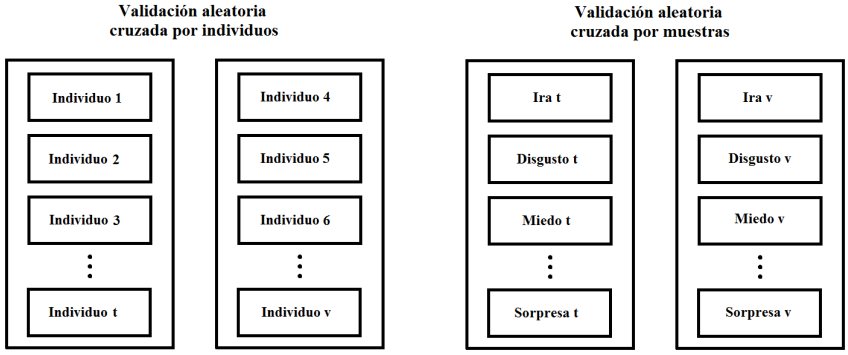
En la figura 31 mostramos la arquitectura de nuestro sistema de clasificación. Los parámetros F_p corresponden a cada una de las celdas TPOEM. En tanto que cada parámetro es un vector, para cada parámetro p corresponde un componente $F_{p,q}$, donde q es la posición en el vector TPOEM. Estos parámetros son la entrada del sistema de clasificación que determina la expresión facial.

6.3.1. Metodología de validación

La metodología usada para todo el proceso es la siguiente: el conjunto de datos completo se divide en 10 partes aproximadamente iguales, usando LSO. La separación se hace basada en individuo, no en muestra, de manera que no hay muestras del mismo individuo tanto en el conjunto de entrenamiento como en el conjunto de validación. Con 9 de las partes se realiza el proceso de reducción supervisada de dimensiones, selección de parámetros y entrenamiento, y con el conjunto restante se realiza la validación. El proceso se repite 10 veces con todos los conjuntos disjuntos posibles, de modo que cada individuo pertenece en alguna iteración al conjunto disjunto de validación y en las iteraciones restantes al conjunto de entrenamiento.

Si bien los requerimientos de cada iteración son considerables, tomar atajos, tales como hacer reducción supervisada de dimensiones y selección de parámetros una sola vez al comienzo con el conjunto de datos y luego entrenar y validar únicamente con los datos reducidos y parámetros seleccionados puede ocasionar un error de prueba de hipótesis. Esto sucede debido a que la reducción supervisada de dimensiones y la selección

Figura 32. Metodología cruzada por individuos (o LSO) contra metodología cruzada por muestras



de parámetros son etapas de pseudoclasificación, así que el uso de esta metodología disminuye el costo de procesamiento de las pruebas, pero produce valores de clasificación superiores a los obtenidos cuando el sistema de entrenamiento es completamente ciego a las muestras de validación. De hecho, dado un número finito de muestras sin repetición idéntica de muestras para dos clases distintas, es posible diseñar una etapa de reducción supervisada de dimensiones que posteriormente permita clasificación perfecta de las muestras usando cualquier tipo de validación por pliegues si la reducción de dimensiones se realiza con todas las muestras, incluyendo las muestras que posteriormente hacen parte del procedimiento de validación.

Salvo señalado explícitamente, todos los resultados en este capítulo son obtenidos usando la metodología aleatoria cruzada por individuos (o LSO). En la figura 32 se muestra la diferencia entre las dos metodologías. En la metodología cruzada por individuos los conjuntos de entrenamiento y de validación son separados basado en individuos, de manera que no es posible contar con muestras del mismo individuo en los dos conjuntos. En la metodología aleatoria cruzada las muestras se eligen aleatoriamente del conjunto completo. Esto implica que en general hay muestras de los mismos individuos en los dos conjuntos y, cuando se usan varias muestras por individuo por expresión, muestras de los mismos individuos y expresiones en los dos conjuntos.

Posteriormente en el capítulo mostraremos el uso de metodología aleatoria cruzada basada en muestras y sus implicaciones en los resultados [2].

[2]El inconveniente de la validación aleatoria cruzada basado en muestras ya fue encontrado en el trabajo en [156]. En este trabajo, usando la base de datos CK, la tasa global de clasificación para el problema de 7 clases con SVM y validacion aleatoria cruzada fue de 94.85 %. Esta clasificación es de muy difícil obtención, particularmente en la base de datos CK, porque en la base de datos revisada CK+ un conjunto numeroso de secuencias fue eliminado porque no era representativo de la expresión facial, de manera que es difícil entender cómo se puede obtener una tasa de acierto tan alta. Para el mismo problema usando LSO, con clasificadores basados en AdaBoost (lamentablemente no hay ejemplo de este tipo de validación con clasificadores SVM, para hacer una comparación directa) la tasa global de clasificación obtenida fue de 89.14 % que es un valor considerablemente menor que con validación aleatoria cruzada, pero mucho más realista. La diferencia no es debida a los distintos clasificadores,

6.3.2. Aproximación por discriminantes lineales convencionales

El uso de 2 orientaciones espaciales más una orientación temporal para la codificación TPOEM representa un conjunto de 192 vectores (256 si se usan 3 orientaciones espaciales), que corresponden a 8×8 celdas por 3 orientaciones. Nuestra primera aproximación fue usar cada celda (vector) como un metaclasificador y posteriormente fusionar la salida de cada uno, sea como una suma simple de votos de clasificación o con un clasificador fuerte más complejo.

En primera instancia diseñamos 7 clasificadores por FDA para cada celda TPOEM, con el fin de usar una metodología *1 vs. all*. El criterio de clasificación es la suma simple de los resultados de cada clasificador. El entrenamiento es mostrado en el pseudoalgoritmo 4.

Algorithm 4 Entrenamiento por discriminante Fisher

procedure ENTRADA
  $X_k = \{x_1, x_2, ..., x_N\}$, $C = \{C_1, ..., C_N\}$, $C_i \in \{1, 7\}$
procedure GENERACIÓN DE DATOS
  X_k es separado en 10 subconjuntos usando LSO
procedure ENTRENAMIENTO
  for t=1 **to** 10 **do**
    X_k^{tr} : subconjunto de entrenamiento
    X_k^{v} : subconjunto de validación
    for k=1 **to** 7 **do**
      for c=1 **to** 7, $c \neq k$ **do**
        $class_1 = X_k^{tr}, , \ C_i = cl$
        $class_2 = X_k^{tr}, , \ C_i \neq cl$
        Entrenar $FD_{k,cl,t}$
      end
    end
  end

En este procedimiento se entrenan los clasificadores $FD_{k,cl,t}$, donde k es cada celda, cl es cada clase por celda y t es la iteración. Una vez obtenidos los clasificadores, la etapa de validación es descrita en el pseudoalgoritmo 5 .

Este procedimiento también puede ser ejecutado usando diferentes pesos por celda, tal como se hizo en el capítulo 3. Para ello, simplemente el cálculo de los valores de $Scl_{i,k,c}$ se hace según la ecuación 6.18.

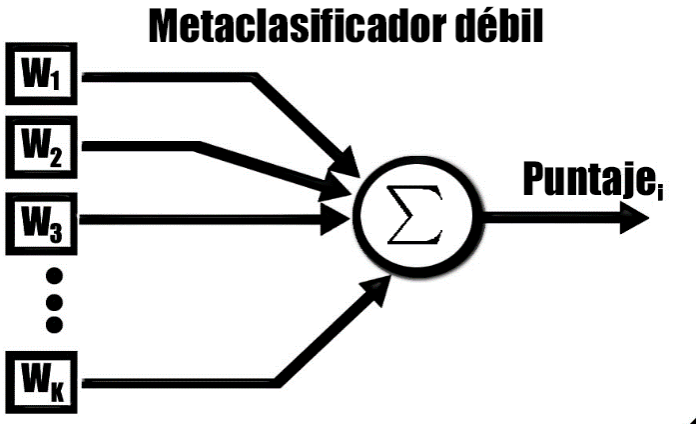
$$Scl_{i,k,c} = \sum_k w_k z^v_{i,k,c}, \quad c = 1, ..., 7 \tag{6.18}$$

puesto que en este mismo trabajo se hace clasificación usando SVM y AdaBoost, validación *leave-subjects-out* para 8 clases (incluyendo la expresión desprecio) para la base de datos CK+ y los resultados son muy similares: 76.85 % y 76.14 % respectivamente. Esto muestra el gran efecto del protocolo de validación en los resultados de clasificación global.

Algorithm 5 Validación por discriminante Fisher

procedure ENTRADA
    $X_k = \{x_1, x_2, ..., x_N\}$, $C = \{C_1, ..., C_N\}$, $C_i \in \{1, 7\}$
procedure VALIDACIÓN
    for t=1 **to** 10 **do**
        X_k^v : subconjuntos de validación ▷ Los mismos subconjuntos de validación obtenidos en la etapa de entrenamiento
        for i=1 **to** N **do**
            for k=1 **to** K **do**
                Obtener la salida de los metaclasificadores FD por celda:
                $FD_{k,cl,t}\{X_{i,k}^v\} \rightarrow z_{i,k,c}^v, \;\; c = C_i$
            end
            La muestra $x_{i,k}^v$ es clasificada usando *winner takes all*:
            $Scl_{i,k,c} = \sum_k z_{i,k,c}^v, \;\; c = 1, ..., 7$
            $y_i = \arg\max_c \sum_k Scl_{i,k,c}$
            if $y_i = C_i$ **then**
                Incrementar la clasificación correcta de la clase C_i
            else
                Incrementar la clasificación incorrecta de la clase C_i
            end
            Ajustar la matriz de confusión:
            $M_{C_i,y_i} \leftarrow M_{C_i,y_i} + 1$
        end
    end

Figura 33. Fusión de clasificadores débiles ponderados



El proceso de fusión simple de clasificadores es mostrado en la figura 33. Cada uno de los puntajes obtenidos es multiplicado por la ponderación w_k y se obtiene el puntaje por cada muestra i.

Los resultados iniciales de desarrollo de este capítulo son mostrados usando la base de datos CK en vez de la base de datos CK+. La motivación para esta decisión es que la mayor parte de trabajos publicados en reconocimiento automático de expresión facial usan la base de datos CK, que ha estado disponible por mucho más tiempo, de manera que la comparación entre nuestros resultados y otros resultados del estado del arte es más directa. No obstante, en los desarrollos finales hacemos entrenamiento y validación usando tanto la base de datos CK como la base de datos CK+, con el fin de posibilitar la comparación de resultados usando cualquiera de estas dos bases de datos como referencia.

Los resultados de clasificación de 7 expresiones y 6 expresiones usando la composición de metaclasificadores por FDA son mostrados en las tablas 20 y 21 respectivamente.

Las tasas globales de clasificación para 7 y 6 expresiones son 85.34 % y 94.31 %. Nuestras pruebas previas usando la base de datos CK y clasificación por template matching dieron resultados de 83.3 % y 93.3 % respectivamente, de modo que la inclusión de metaclasificación ponderada por FDA representó mejoría sustancial [3].

Los resultados fueron obtenidos haciendo 10 repeticiones LSO y promediando los resultados de 50 entrenamientos y clasificaciones [4].

[3]Con el fin de determinar que efectivamente el resultado con FDA fue una mejora estadística y no producto de variaciones normales de resultados, hicimos una prueba t-test por pares. El t-valor obtenido fue 3.37, para $p = 0{,}0023$ y $1 - p = 0{,}9977$. Esto garantiza incluso con $\alpha = 1\,\%$ que FDA produce una mejora de resultados de clasificación. Para el resto de pruebas de clasificación mostradas en este capítulo usamos prueba t-test por pares y, en todas las pruebas los resultados tienen validez estadística con $\alpha < 5\,\%$ comparados con la prueba precedente, salvo en las pruebas con deep learning, que no mostraron mejoría estadística t-test comparado con clasificación SVM+FDA.

[4]Normalmente el protocolo con 10 pliegues hace únicamente 10 entrenamientos y validaciones. Sin embargo, al hacer un mayor número de entrenamientos y validaciones que el número de pliegues, se

Tabla 20. Reconocimiento de 7 expresiones faciales usando TPOEM y estimación por metaclasificadores FDA, base de datos CK

	Ira	Dis.	Mie.	Ale.	Tri.	Sor.	Neu.
Ira	**83.4**	4.4	0.0	0.0	0.0	0.0	12.2
Disgusto	1.5	**92.8**	0.0	0.0	1.2	0.0	4.5
Miedo	1.5	1.8	**70.3**	8.1	2.8	0.0	15.5
Alegría	0.0	0.0	0.8	**96.0**	0.0	0.0	3.2
Tristeza	3.0	0.0	0.0	0.0	**74.9**	2.4	19.7
Sorpresa	1.5	0.0	0.8	0.0	0.0	**92.9**	4.8
Neutral	0.8	0.0	0.8	0.0	11.3	0.0	**87.1**

Tabla 21. Reconocimiento de 6 expresiones faciales usando TPOEM y estimación por metaclasificadores FDA, base de datos CK

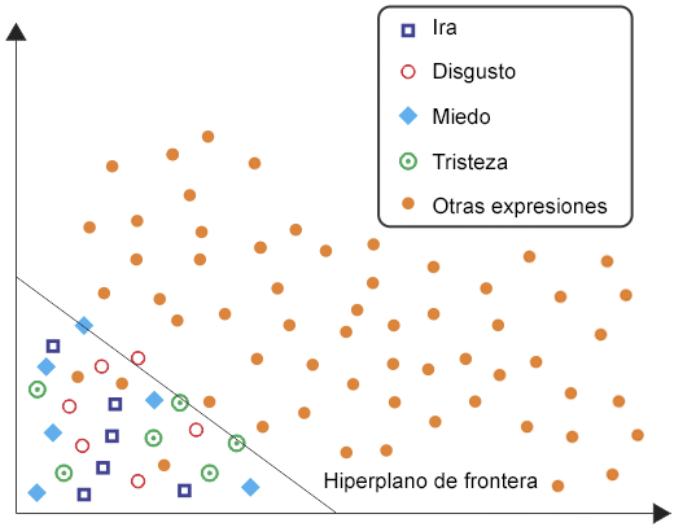
	Ira	Dis.	Mie.	Ale.	Tri.	Sor.
Ira	**91.4**	7.9	0.7	0.0	0.0	0.0
Disgusto	1.9	**97.0**	0.6	0.0	0.5	0.0
Miedo	1.3	1.3	**86.2**	8.5	2.7	0.0
Alegría	0.0	0.0	1.8	**97.8**	0.0	0.4
Tristeza	2.6	0.0	0.0	1.8	**93.7**	1.9
Sorpresa	0.0	0.0	0.0	0.2	0.0	**99.8**

6.3.3. Análisis por discriminante Fisher con valor de confianza individual *a posteriori* por celda

El uso de los metaclasificadores 1 vs. todos con discriminante Fisher tiene una importante deficiencia. En el capítulo 3 se mostró cómo algunas celdas proporcionan mayor información de discriminación y debido a ello el uso de pesos ponderados mejoró las tasas de clasificación. Sin embargo, los discriminantes Fisher por celda tienen distintas capacidades de discriminación de expresiones individuales en celdas particulares. Por ejemplo, algunas celdas espaciales son mejores en la discriminación de ciertas expresiones, mientras que su desempeño es pobre en la discriminación de otras expresiones. Un ejemplo es simulado en la figura 34.

aumenta el grado de confianza de la respuesta, por cuanto en distintas iteraciones se están usando diferentes pliegues de entrenamiento y validación en todo caso. El cálculo de confianza estadística del resultado es, no obstante, más complejo, ya que el uso de mayor número de iteraciones que pliegues implica reducir independencia entre iteración e iteración, pero en todo caso constituye una opción para mejorar el grado de confianza estadística de la prueba con un conjunto de datos limitado.

Figura 34. Simulación del traslape de parámetros en la región superior del rostro en expresiones de ira, disgusto, miedo y tristeza



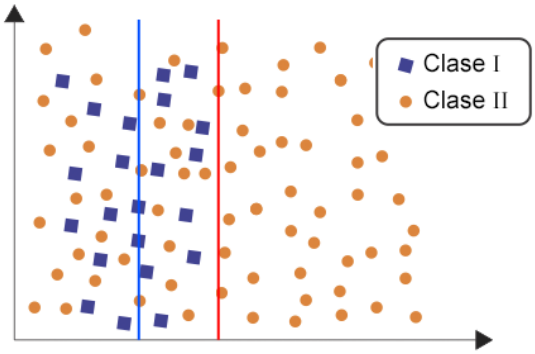
Dado el traslape entre distintas expresiones en las celdas en el ceño, estas celdas individualmente son pobres discriminantes de la expresión facial, así que su confiabilidad no es buena si se usan individualmente y el uso de pesos estáticos no proporciona mejora de discriminación. Por otra parte, estas celdas proporcionan alta tasa de discriminación entre grupos de expresiones que tienen gestos de ceño similares (i.e. una expresión del ceño descarta que la expresión sea alegría, sorpresa o neutral, así no dé infomación suficiente de discriminación entre las demás). Esta información es importante en la definición de la fusión de metaclasificadores para el sistema completo.

De manera similar, las regiones inferiores del rostro tienen capacidades de discriminación entre expresiones y entre grupos de expresiones. Los códigos TPOEM alrededor de la boca tienen alta discriminación de sorpresa y alegría, y ciertas confusiones entre miedo y alegría (especialmente en etapas iniciales de la generación de la expresión de miedo) son solucionadas por las celdas superiores del rostro.

Dadas estas consideraciones, se incluyó una aproximación Bayesiana para añadir la confiabilidad de cada metaclasificador por celda dependiendo de su salida. La diferencia entre el uso de ponderación previa y esta aproximación *a posteriori* está en que la primera idea se basa en la confiabilidad de la celda *per se*, mientras que la confiabilidad *a posteriori* se basa en la salida de la celda.

El uso de confiabilidad *a posteriori* tiene otra ventaja importante cuando hay datos con traslape. Los datos con traslape son un inconveniente considerable en el diseño del sistema de clasificación, especialmente con alta dimensionalidad, pues hay que intentar evitar el traslape sin recurrir a soluciones que se ajusten muy bien a los manifolds

Figura 35. Traslape en un problema de dos clases



particulares, ocasionando overfitting. En expresión facial hay típicamente gran traslape entre datos debido a que hay similitud de gestos locales entre varias expresiones. Desde un punto de vista económico, está bien que las expresiones faciales no involucren la modificación del rostro completo para cada expresión, pero esto genera patrones similares para regiones que no definen expresiones particulares. Hay maneras activas de lidiar con este traslape, tal como incluir una clase adicional etiquetada como desconocida y diseñar un clasificador sin estas muestras o usando una clasificación doble para regiones con y sin traslape, como en [132]. Estas aproximaciones son convenientes cuando la dimensión de las muestras es pequeña, hay datos adecuados para definir apropiadamente las regiones de traslape y el traslape no es severo. Sin embargo, los datos de expresión facial son limitados, lo cual limita el uso de esta alternativa.

En la figura 35 ilustramos el problema usando un problema simple de 2 clases.

En este problema 1 vs. todo, los datos de la clase I están traslapados con los datos del resto de las clases [5] (Clase II). Dado el bajo número de muestras pertenecientes a la clase I comparado al resto de las clases, un clasificador que etiquete la mayoría las muestras como pertenecientes a la clase II produce el más bajo error de clasificación, pero no es útil para problemas de clasificación. Esto es un problema al usar métodos sofisticados de clasificación, pues en muchos casos los hiperplanos de clasificación generados ocasionan que muchas de las muestras fuesen clasificadas con la misma clase (hiperplano azul en la figura 35), e incluso era usual obtener metaclasificadores que clasificaban todas las muestras como pertenecientes a la misma clase salvo ajuste muy fino individual de cada metaclasificador. Sin embargo, otro hiperplano de clasificación (hiperplano rojo en la figura 35) produce mayor error de clasificación global, pero ninguna muestra clasificada

[5] Esto ocurre con expresiones faciales frecuentemente. Por ejemplo, algunos parámetros en la boca y regiones cercanas tienen patrones similares entre las expresiones de alegría y miedo, mientras que el resto de las expresiones tienen muy distintos patrones en estas regiones. Como tal, una arquitectura 1 vs. todos produce un traslape de alta dimensión similar cuando la expresión de la muestra es alegría o miedo.

como clase II pertenece a la clase I. Es decir, la confianza *a posteriori* en este caso da como resultado que para este clasificador específico una salida de clasificación de clase II automáticamente descarta la muestra como perteneciente a la clase I, sin importar que la tasa global de clasificación sea inferior que en el primer caso. En consecuencia, en un problema con parámetros débiles no necesariamente los mejores clasificadores son los clasificadores que tengan menor tasa de error global, sino los clasificadores que permitan aumentar la confiabilidad de clasificación.

La metodología usada para esta etapa fue incluir el cálculo de la confianza *a posteriori* con un subconjunto de los datos de entrenamiento. El conjunto de entrenamiento fue separado en dos conjuntos disjuntos. Con el primero se hizo reducción de dimensiones, selección de parámetros y entrenamiento. Con el segundo se hizo el cálculo de la confianza *a posteriori*. La etapa de cálculo de confianza *a posteriori* es realizada una vez los metaclasificadores $FD_{k,cl,t}$ han sido obtenidos con el subconjunto de entrenamiento 1, y se muestra en el pseudoalgoritmo 6. Denominamos a este procedimiento novedoso APCC (*A Posteriori Confidence Classification*).

La inclusión de las confiabilidades *a posteriori* por celda produjo mejoras en la tasa de clasificación. Las matrices de confusión obtenidas están mostradas en las tablas 22 y 23.

Tabla 22. Reconocimiento de 7 expresiones faciales usando TPOEM y estimación APCC, base de datos CK

	Ira	Dis.	Mie.	Ale.	Tri.	Sor.	Neu.
Ira	**86.3**	3.6	0.0	0.0	0.0	0.0	10.1
Disgusto	1.4	**93.5**	0.0	0.0	1.0	0.0	4.1
Miedo	1.7	1.5	**74.6**	6.8	2.3	0.0	13.1
Alegría	0.0	0.0	1.1	**96.2**	0.0	0.0	2.7
Tristeza	2.7	0.1	0.0	0.0	**78.9**	2.5	15.8
Sorpresa	1.3	0.0	1.2	0.0	0.0	**93.1**	4.4
Neutral	0.0	0.0	0.4	0.0	10.6	0.0	**89.0**

La tasa global de reconocimiento de 7 clases es 87.4 % y de 6 clases es 95.5 %. Estos resultados muestran que el uso de la confiabilidad *a posteriori* produce una mejoría notable en los resultados de clasificación, incluso con metaclasificadores FDA simples [6].

Si bien los resultados son razonablemente buenos, los clasificadores FDA no producen un hiperplano de separación que se pueda ajustar un poco mejor a las posibles complejidades de los *manifolds* de alta dimensión, los datos no convexos de la codificación usada y el frecuente traslape de los datos. Adicionalmente, en los espacios de

[6] En comparación con la prueba sin confiabilidad a posteriori, los resultados de t-test produjeron $p = 0{,}0016$

Algorithm 6 Entrenamiento y validación *A posteriori*

procedure ENTRADA
  $X_k = \{x_1, x_2, ..., x_N\}$, $C = \{C_1, ..., C_N\}$, $C_i \in \{1, 7\}$
procedure CONFIABILIDAD POR CELDA
  for t=1 **to** 10 **do**
    X_k^{tr2} : subconjunto de entrenamiento ▷ Subconjunto disjunto de validación no usado para reducción de dimensiones, extracción de parámetros o diseño del sistema de clasificación
    for i=1 **to** N **do**
      for k=1 **to** K **do**
        Obtener la salida de los metaclasificadores por celda:
        $FD_{k,cl,t}\{X_{i,k}^{tr2}\} \rightarrow z_{i,k,c}^{tr}, \;\; c = C_i$
      end
    end
    $P_{k,c,t} = P(x_{i,k}^{tr2} \in C_i | z_{i,k,c}^{tr} > 0)$
    $P'_{k,c,t} = P(x_{i,k}^{tr2} \notin C_i | z_{i,k,c}^{tr} < 0)$
  end
procedure VALIDACIÓN
  for t=1 **to** 10 **do**
    X_k^v : subconjunto de validación
    for i=1 **to** N **do**
      for k=1 **to** K **do**
        Obtener la salida de los metaclasificadores FD por muestra por celda:
        $FD_{k,cl,t}\{x_{i,k}^v\} \rightarrow z_{i,k,c}^{tr}, \;\; c = C_i$
        for c=1 **to** 7 **do**
          if $z_{i,k,c}^{tr} > 0$ **then**
            $Scl_{i,k,c} \leftarrow Scl_{i,k,c} + P_{k,c,t}$
          else
            $Scl'_{i,k,c} \leftarrow Scl'_{i,k,c} + P'_{k,c,t}$
          end
        end
      end
      La muestra $x_{i,k}^v$ es clasificada usando *winner takes all*
      $y_i = \arg\max_c \sum_k (Scl_{i,k,c} - Scl'_{i,k,c}), \;\; c = \{1, 7\}$
      Ajustar la matriz de confusión:
      $M_{C_i,y_i} \leftarrow M_{C_i,y_i} + 1$
    end
  end

Tabla 23. Reconocimiento de 6 expresiones faciales usando TPOEM y estimación APCC, base de datos CK

	Ira	Dis.	Mie.	Ale.	Tri.	Sor.
Ira	**93.2**	6.3	0.5	0.0	0.0	0.0
Disgusto	2.3	**96.5**	1.2	0.0	0.0	0.0
Miedo	0.8	1.0	**91.5**	4.2	2.5	0.0
Alegría	0.0	0.0	1.7	**97.5**	0.0	0.8
Tristeza	2.2	0.0	0.0	1.5	**94.7**	1.6
Sorpresa	0.0	0.0	0.0	0.4	0.0	**99.6**

alta dimensión los datos tienden a agruparse en las esquinas de alta dimensión [7]. Muchas de las muestras TPOEM de alta dimensión tienden a ocupar espacios en regiones en las cuales los volúmenes son grandes, pero métricas convencionales como distancias Euclidianas tienden a ser relativamente pequeñas. Adicionalmente, los espacios de alta dimensión no son adecuadamente caracterizados por un conjunto limitado de muestras, de modo que la discriminación por una proyección a un espacio de una dimensión puede ser problemática dado que las dispersiones y los promedios de los conjuntos de datos pueden ser inadecuadamente obtenidos debido al reducido número de muestras disponibles.

En consecuencia, pese a la mejoría obtenida con el uso de las confiabilidades *a posteriori*, consideramos que el uso de una arquitectura 1 vs. 1 puede ser mejor para ayudar con el inconveniente del traslape, especialmente si el traslape ocurre entre subconjuntos que no interfieren en una competencia 1 vs. 1 particular. Se puede observar esto en un ejemplo simple mostrado en la figura 36.

Nótese cómo pese a que hay fuerte traslape entre los datos en las clases I, II y III, que produciría problemas en una arquitectura 1 vs. todos, el uso de una arquitectura 1 vs. 1 produce adecuada capacidad de discriminación de los datos para las clases sin traslape y una mejor capacidad de discriminación para las clases con traslape. La metodología es descrita en el pseudoalgoritmo 7.

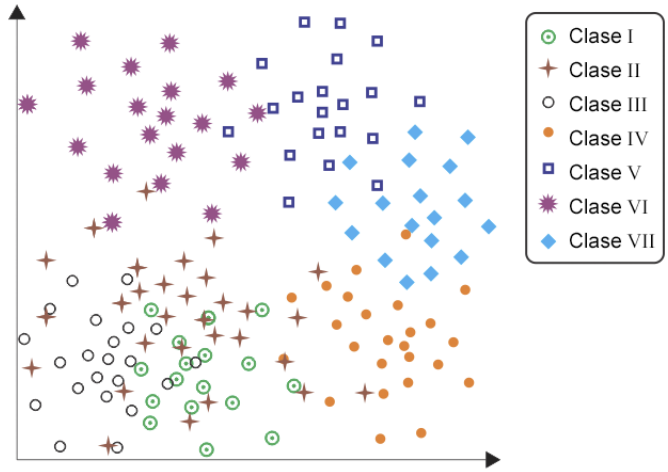
La salida del protocolo descrito es un conjunto de clasificadores 1 vs. 1 en $FD_{k,c_1,c_2,t}$ donde k es el índice de los vectores TPOEM, c_1 y c_2 son las dos clases del enfrentamiento 1 vs. 1 y t es el número de plieque LSO.

La estimación de la confiabilidad *a posteriori* por celda y la validación se hacen según el pseudoalgoritmo 8.

El uso de metaclasificadores 1 vs. 1 por celda mejora las tasas de clasificación,

[7] En http://blogs.msdn.com/b/ericlippert/archive/2005/05/13/high-dimensional-spaces-are-counterintuitive-part-two.aspx se encuentra una explicación intuitiva y simple de cómo los datos de alta dimensión tienden a agruparse en las esquinas del espacio y ocupar el volumen en la superficie del espacio [91].

Figura 36. Simulación de problema de clasificación de 7 clases



Algorithm 7 Discriminante Fisher con arquitectura 1 vs. 1

procedure ENTRADA
    $X_k = \{x_1, x_2, ..., x_N\}$, $C = \{C_1, ..., C_N\}$, $C_i \in \{1, 7\}$
procedure GENERACIÓN DE DATOS
    X_k es separado en 10 subconjuntos usando LSO
procedure ENTRENAMIENTO
    for t=1 **to** 10 **do**
        $X_k^{tr_1}$: subconjunto de entrenamiento ▷ Subconjunto disjunto de entrenamiento usado para entrenar los FDAs
        $X_k^{tr_2}$: training subset 2 ▷ Subconjunto disjunto de entrenamiento usado para obtener los valores de confianza *a posteriori*
        X_k^{v} : subconjunto de validación ▷ Subconjunto disjunto de validación usado para probar la clasificación
        for k=1 **to** K **do**
            for c_1=1 **to** 7 **do**
                for c_2=1 **to** 7, $c_2 \neq c_1$ **do**
                    $class_1 = X_k^{tr_1}$, $C_i = c_1$
                    $class_2 = X_k^{tr_1}$, $C_i = c_2$
                    Entrenar $FD_{k,c_1,c_2,t}$ con $class_1$ y $class_2$
            end
        end
    end
end

Algorithm 8 Arquitectura 1 vs. 1 FDA, estimación APCC y validación

procedure ENTRADA
  $X_k = \{x_1, x_2, ..., x_N\}$, $C = \{C_1, ..., C_N\}$, $C_i \in \{1, 7\}$
procedure CONFIABILIDAD POR CELDA POR SALIDA
  for t=1 **to** 10 **do**
    $X_k^{tr_2}$: subconjunto de entrenamiento 2
    for i=1 **to** N **do**
      for k=1 **to** K **do**
        Obtener la salida de los metaclasificadores FD por celda:
        $FD_{k,c_1,c_2,t}\{x_{i,k}^{tr_2}\} \rightarrow z_{i,k,c_1,c_2}^{tr}$
      end
    end
    $P_{k,c_1,c_2,t} = P(x_{i,k}^{tr_2} \in C_{c_1} | z_{i,k,c_1,c_2}^{tr} > 0)$
    $P'_{k,c_1,c_2,t} = P(x_{i,k}^{tr_2} \notin C_{c_1} | z_{i,k,c_1,c_2}^{tr} < 0)$
  end
procedure VALIDACIÓN
  for t=1 **to** 10 **do**
    X_k^v : subconjunto de validación
    for i=1 **to** N **do**
      for k=1 **to** K **do**
        Obtener la salida de los metaclasificadores FD por celda:
        $FD_{k,c_1,c_2,t}\{x_{i,v}^{tr_2}\} \rightarrow z_{i,k,c_1,c_2}^{v}, c_1 = 1, ..., 7, \;\; c_2 = 1, ..., 7$
        for c_1=1 **to** 7 **do**
          for c_2=1 **to** 7 **do**
            if $z_{i,k,c_1,c_2}^{v} > 0$ **then**
              $Scl_{i,k,c_1} \leftarrow Scl_{i,k,c_1} + P_{k,c_1,c_2,t}$
            else
              $Scl_{i,k,c_1} \leftarrow Scl_{i,k,c_1} - P'_{k,c_1,c_2,t}$
          end
        end
      end
      La muestra $x_{i,k}^v$ es clasificada por *winner takes all*:
      $y_i = \arg\max_c \sum_k (Scl_{i,k,c} - Scl'_{i,k,c}), \;\; c = \{1, 7\}$
      Ajustar la matriz de confusión:
      $M_{C_i,y_i} \leftarrow M_{C_i,y_i} + 1$
    end
  end
end

principalmente en situaciones en las cuales hay traslape local de clases. Sin embargo, hay dos desventajas notables de esta aproximación que en ocasiones hacen preferible el uso de técnicas más sencillas. La primera se refiere al costo de procesamiento de calcular 21 clasificadores por cada vector TPOEM. En total, para un conjunto de aproximadamente 100 vectores TPOEM, se requiere de aproximadamente 2100 clasificadores 1 vs. 1 y el proceso de entrenamiento y validación total requiere de 10 pliegues LSO, con cálculo de aproximadamente 21000 metaclasificadores 1 vs. 1 con sus respectivos valores de confidencia *a posteriori*. Si bien el cálculo de los metaclasificadores FDA es relativamente rápido, el cálculo de los valores de confianza *a posteriori* (definidos por las ecuaciones $P_{k,c_1,c_2,t} = P(x_{i,k}^{tr_2} \in C_{c_1} | z_{i,k,c_1,c_2}^{tr} > 0)$ y similares), es un proceso exhaustivo costoso, pues la estimación probabilística Bayesiana requiere estimar los posibles valores de z_{i,k,c_1,c_2}^{tr} para poder determinar con fiabilidad la probabilidad condicional de que una muestra pertenezca a una clase dado su valor de salida z. El procedimiento de validación es un proceso rápido, ya que la salida de un banco de clasificadores FDA, incluyendo su ponderación *a posteriori* es cálculo que no interfiere en los objetivos de tiempo real, pero el extenso proceso previo dificulta la modificación de los algoritmos luego de un entrenamiento, cuyo costo promedio en nuestro sistema de cómputo fue de promedio 9 horas por entrenamiento y validación con todo el conjunto de datos.

El segundo problema se refiere al mecanismo de decisión binario de un metaclasificador FDA. Ilustraremos el problema con un ejemplo simple basado en resultados reales obtenidos. Las expresiones ira y disgusto tienen patrones similares en la región superior del rostro. Esto implica que en los concursos 1 vs. 1 en esta región, las muestras de estas dos expresiones tienen puntajes divididos entre ira y disgusto. Por otra parte, en la región inferior del rostro hay gran similitud entre las expresiones ira y neutral [8]. Es decir, en una muestra de ira, la expresión disgusto le "roba" puntos en la región superior del rostro a la muestra y la expresión neutral le "roba" puntos en la región inferior del rostro. En consecuencia, este protocolo de clasificación, que no incluye información mutua entre distintas regiones espaciales, penaliza expresiones que comparten patrones locales con otras expresiones. El uso de los valores de confidencia *a posteriori* atenúa fuertemente este problema, por cuanto la decisión deja de ser binaria y una muestra obtiene puntos incluso si no gana el concurso 1 vs. 1, pero sigue siendo un inconveniente que puede ser solucionado con alternativas novedosas que veremos posteriormente en el capítulo.

Los resultados de clasificación obtenidos con la metodología de FDA 1 vs. 1 con confidencia *a posteriori* son mostrados en las tablas 24 y 25.

La tasa global de reconocimiento de 7 clases es 88.2 % y la tasa global de reconocimiento de 6 clases es 95.9 % [9].

[8] Entendiéndose ira real, como la mostrada en las bases de datos CK y CK+. La ira impostada caricaturesca, en la que el individuo muestra los dientes pero no corresponde a una expresión natural de ira, no tiene esta neutralidad en la región inferior del rostro. Incidentalmente, algunas bases de datos no validadas incluyen muestras de ira de esta naturaleza impostada, lo que dificulta su validación con una base de datos estándar, tal como mostraremos en el capítulo 7.

[9] En comparación con la prueba con confiabilidad a posteriori 1 vs. todos, los resultados de t-test

Tabla 24. Reconocimiento de 7 expresiones faciales usando TPOEM y estimación por FDA 1 vs. 1, APCC, base de datos CK

	Ira	Dis.	Mie.	Ale.	Tri.	Sor.	Neu.
Ira	**86.7**	3.5	0.0	0.0	0.0	0.0	9.8
Disgusto	1.5	**93.7**	0.0	0.0	0.5	0.0	4.3
Miedo	1.4	1.5	**79.3**	3.8	2.3	0.0	11.6
Alegría	0.0	0.0	1.1	**96.0**	0.0	0.0	2.9
Tristeza	2.7	0.0	0.0	0.0	**79.4**	2.0	15.9
Sorpresa	0.8	0.0	0.0	0.0	0.0	**92.9**	6.3
Neutral	0.0	0.0	0.6	0.0	9.4	0.0	**90.0**

Tabla 25. Reconocimiento de 6 expresiones faciales usando TPOEM y estimación por FDA 1 vs. 1, APCC, base de datos CK

	Ira	Dis.	Mie.	Ale.	Tri.	Sor.
Ira	**94.0**	5.2	0.8	0.0	0.0	0.0
Disgusto	2.3	**96.2**	1.3	0.0	0.2	0.0
Miedo	0.7	1.1	**92.9**	3.1	2.2	0.0
Alegría	0.0	0.0	1.6	**97.9**	0.0	0.5
Tristeza	1.7	0.0	0.0	1.5	**94.5**	2.3
Sorpresa	0.0	0.0	0.0	0.0	0.0	**100.0**

Antes de continuar con los desarrollos más avanzados de los sistemas de clasificación usados en este trabajo, haremos una breve revisión del uso de discriminantes lineales en este tipo de problemas y sus limitaciones.

6.3.4. Discriminantes lineales en espacios de alta dimensión

Los discriminantes lineales tienen importantes ventajas, siendo las más notables la simplicidad y la facilidad de separación entre clases cuando los subconjuntos de datos son adecuados. Sin embargo, al usar conjuntos reales puede haber inconvenientes que deterioran o, incluso, destruyen su desempeño. En el desarrollo de este trabajo encontramos problemas al trabajar con discriminantes lineales en espacios de alta dimensión.

Traslape de los *manifolds*: El mejor escenario posible para el uso de análisis discriminante lineal es cuando los datos son completamente separables por un hiperplano

con 1 vs. 1 produjeron $p = 0{,}0017$.

lineal, preferiblemente si además hay una separación considerable entre las clases. En esta condición hay al menos una solución (en general, infinitas soluciones) que separa completamente las regiones pertenecientes a distintas clases. En nuestro trabajo, éste no es el caso. La mayor parte de las celdas espaciales corresponden a regiones faciales donde los parámetros extraídos pueden ser similares entre dos o más expresiones. Debido a esto, en una arquitectura 1 vs. todos estos parámetros particulares se traslapan, así que el clasificador es forzado a determinar una frontera de decisión que naturalmente tiene errores de clasificación.

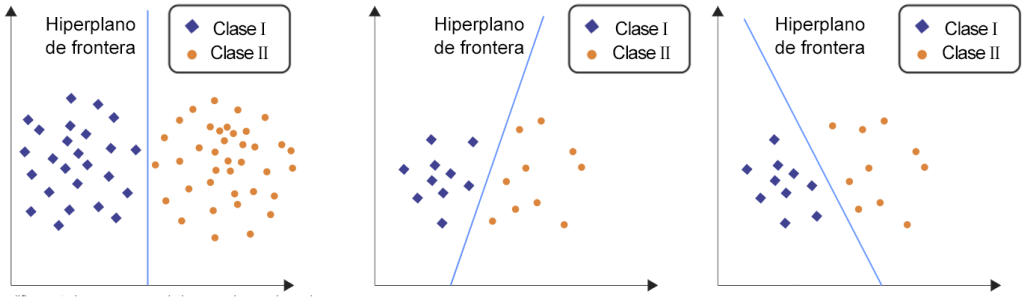
Incluso el uso de una arquitectura 1 vs. 1, si bien reduce el problema, no impide el fuerte traslape entre 2 ó más clases. El uso de nuestra propuesta de confiabilidad APCC redujo notablemente este inconveniente, puesto que no penaliza por completo las expresiones que tengan similitudes con otras expresiones y, por otra parte, celdas cuya relevancia no es notable en la clasificación de algunas expresiones específicas no asignan un puntaje máximo a ninguna expresión en los concursos 1 vs. 1 [10]. Sin embargo, si bien esta notable mejora, el cálculo de probabilidades *a posteriori* es un proceso de moderada precisión con el número limitado de muestras.

Representación de los datos: Idealmente, un conjunto de datos debe proporcionar representación adecuada y suficiente de las clases a ser discriminadas. En nuestro caso, nuestras pruebas previas mostraron que los parámetros TPOEM representan adecuadamente la expresión facial, por cuanto incluso con el uso de clasificadores por métricas chi-cuadrado los resultados fueron buenos. Sin embargo, cualquier clasificación que requiera del modelamiento probabilístico de los datos en alta dimensión tiene inconvenientes debido al número limitado de datos, y el problema se incrementa exponencialmente con la dimensión. Lamentablemente, en las aplicaciones reales la disponibilidad de datos es generalmente limitada y, en nuestro caso particular de expresión facial, bastante reducida. Ilustraremos este problema con un ejemplo sencillo de dimensión 2.

En ambos casos mostrados en la figura 37 las clases I y II son perfectamente separables. En la figura de la izquierda hay un adecuado número de observaciones, de modo que proporcionan buena representación de las clases, incluso cuando sólo un subconjunto de los datos es usado para obtener los clasificadores. Como consecuencia, cualquier hiperplano de separación obtenido incluso por técnicas simples como LDA permite clasificación perfecta. En cambio, en las figuras del centro y la derecha se muestra un problema con baja cantidad de muestras para representación. Usando una metodolo-

[10] Para entender un poco mejor esto en nuestro sistema particular, nótese que las regiones en los pómulos del rostro no aportan información notable más que para la clasificación de un número limitado de expresiones, quizás únicamente alegría. Con un protocolo 1 vs. todos o 1 vs. 1 convencional, las celdas de estas regiones forzosamente tienen que atribuir puntos completos a algunas expresiones. Por ejemplo, en la región de los pómulos un concurso 1 vs. 1 entre disgusto y miedo, con una entrada de alegría, forzosamente da 1 punto fácil a disgusto o a miedo. La confianza *a posteriori* hace que la clase que gane en este concurso específico, sea disgusto o miedo, no gana más que una fracción pequeña de puntos, puesto que el cálculo previo de confianza *a posteriori* determinó que los parámetros obtenidos en esta región de cualquier manera pertenecen a disgusto o a miedo con muy baja (incluso nula) probabilidad.

Figura 37. Representación de datos en un problema de clasificación



gía convencional de entrenamiento y validación por *folding*, la frontera de separación depende fuertemente de cuáles fueron las muestras aleatoriamente seleccionadas para el entrenamiento, de manera que en muchos casos la frontera de separación es defectuosa. Esto hace que dependiendo de los pliegues usados para entrenamiento, se puede obtener fronteras muy distintas. Si se considera que en nuestro caso la dimensión no es una baja dimensión 2, sino dimensión promedio 34, naturalmente los hiperplanos de separación entre clases son frecuentemente inadecuados y, tal como determinamos en nuestras pruebas, altamente variables en inclinación y *offset*.

Complejidad de los *manifolds*: La discriminación lineal se basa en la presunción de que los datos pueden ser aproximadamente discriminados usando una frontera lineal de separación. Sin embargo, éste no es necesariamente el caso. En nuestro trabajo encontramos que los hipervolúmenes de alta dimensión por clase para nuestro conjunto de datos son frecuentemente complejos y no convexos, lo cual, sumado a la cercanía entre los *manifolds* hace imposible definir un buen hiperplano de separación.

Las formas complejas y no convexidades en los subconjuntos de datos introducen errores de clasificación porque no hay una frontera lineal de separación que pueda ser ajustada para discriminar adecuadamente entre las clases. Una alternativa es el uso de transformaciones con *kernels* o proyecciones a espacios de mayor dimensión en los cuales se pueda hacer discriminación perfecta. Sin embargo, la proyección a espacios de alta dimensión requiere que las formas de los *manifolds* de menor dimensión estén adecuadamente representadas, lo cual no es el caso debido al limitado número de observaciones, y frecuentemente la proyección produce *overfitting* en el diseño de los clasificadores [11].

Nuestra siguiente modificación a la metodología de clasificación fue incluir el uso de máquinas de soporte vectorial adaptadas teniendo en cuenta estas consideraciones, en la siguiente subsección.

[11] Para que un kernel produzca una transformación de datos aceptable, es indispensable que los datos representen apropiadamente los *manifolds* complejos. De otra forma la transformación seguramente adaptará la forma de los datos proyectados, de modo que los datos de entrenamiento son fácilmente separables, pero con baja capacidad de generalización.

6.3.5. Metaclasificación basada en máquinas de soporte vectorial

Las máquinas de soporte vectorial generalmente proporcionan adecuada separación en un problema binario en comparación a la separación obtenida por discriminación lineal [55] [12]. La principal motivación para el uso de SVM como metaclasificadores es la complejidad y traslape de los *manifolds*, puesto que las SVM pueden obtener fronteras de decisión con reglas de frontera más sofisticadas que permiten mejorar la separabilidad entre las clases.

Nuestra primera implementación fue usar SVM con funciones Gaussianas de base radial, con el objetivo de desarrollar clasificación no lineal que se pueda addaptar mejor a la forma de los manifolds complejos por clase. El modelo de las SVM-RBF(máquinas de soporte vectorial con funciones de base radial) está dado en la ecuación 6.19.

$$y_i(x) = \sum_{j=1}^{m} w_{ij}\phi\left(\frac{|x-\mu_j|}{h}\right) + w_i, \quad i = 1, ..., N \tag{6.19}$$

donde w es la transformación, ϕ es la función de base radial, μ es el valor de centro, que puede ser calculado de diversas formas, y h es el factor de suavizado de la función radial.

En el primer entrenamiento usamos 3 muestras por individuo por expresión, usando una expresión pico, 2 expresiones menos marcadas y 3 cuadros correspondientes a instancia neutral por individuo[13]. Las funciones de base radial elegidas son Gaussianas dadas por $\phi(z) = e^{-z^2}$. El criterio usado para definir los centros μ_i fue una metodología supervisada basada en el centro de masa de los datos por clase. Técnicas superiores tales como *clustering* supervisado por *k-means* no proporcionaron distinguibles mejores resultados haciendo pruebas limitadas y el cálculo usando *k-means* es computacionalmente costoso para ejecutar todo el proceso de entrenamiento y validación. Para el cálculo del suavizado h la literatura sugiere el uso de s_{max} definido en la ecuación 6.20.

$$s_{max} = \max_{ij} |x_i - \mu_j| \tag{6.20}$$

El cálculo de s_{max} usando este criterio puede conducir a una sobreestimación de la dispersión de los datos, teniendo en cuenta que muchos de los datos TPOEM corresponden a *outliners*[14]. Cuando los datos están conformados por muestras bien definidas y agrupadas en racimos espaciales, el criterio es válido. Sin embargo, nuestras pruebas

[12] Sin embargo, posteriormente veremos cómo en algunos casos las SVM proporcionan más alta tasa global de clasificación, pero su uso como metaclasificadores débiles es más deficiente desde el punto de vista de teoría de información.

[13] Nótese, sin embargo, que la validación se hace por *leave-subjects-out*, de modo que no es posible que el sistema incluya una muestra de expresión de un individuo y la validación incluya otra muestra del individuo de la misma u otra expresión.

[14] Adicionalmente, en el caso de la base de datos CK hay un numeroso conjunto de muestras no representativas o con etiquetas incorrectas, de modo que calcular s_{max} con la mayor dispersión probablemente implica que el valor obtenido no se refiere a una muestra representativa de la clase, sino a una muestra que en realidad probablemente pertenece a otra clase.

mostraron valores altos de s_{max} usando este criterio. Consecuentemente, definimos s_{max} según la ecuación 6.21.

$$s_{max} = 3\sqrt{E(x_i - \mu_j)} \tag{6.21}$$

El valor de la función de suavizado aún es afectado por *outliners* y por muestras erróneamente etiquetadas o no representativas, pero el efecto negativo es atenuado. En la práctica, el valor de la función de suavizado no es muy crítico al usar una función Gaussiana, pero es más importante si la función es una placa delgada (*thin plate*) debido a la divergencia y los posibles valores negativos.

El protocolo de entrenamiento es similar al entrenamiento de los metaclasificadores FDA descrito en el pseudoalgoritmo 7, usando metaclasificadores SVM-RBF.

Algunos de los inconvenientes de las SVM eventualmente pueden ser solucionados [15] usando parámetros variables por cada metaclasificador SVM. Es decir, en tanto que los datos por cada parámetro son bastante distintos, es posible determinar parámetros de las SVM específicos para cada clasificador. Por ejemplo, el parámetro C (el factor *soft margin*) puede ser incrementado hasta cierto punto, pero valores demasiado grandes generan SVM muy especializadas, con alto overfitting [143]. Adicionalmente, usar un valor de C dependiente de los datos de entrada es susceptible a error por los *outliners* (típicamente presentes en expresión facial) [24]. El parámetro ϵ también puede ser modificado para usar una SVM especializada. En general, la bibliografía sugiere un valor de ϵ tal que se use aproximadamente el 50 % de las muestras como vectores de soporte [105] y no debe ser tan pequeño tal que produzca *overfitting* [81].

Ajuste fino de los parámetros de las SVM: Con el fin de determinar la posibilidad del ajuste fino de parámetros de las SVM individuales, simulamos conjuntos de datos de dimensión 50 para un problema de 2 clases con distintas características: en algunos casos el traslape entre las dos clases es bastante grande, en otros casos el modelo de dispersión no es Gaussiano y por último con representación asimétrica de los datos, en tanto que la representación de una de las clases es bastante limitada comparada con la otra, así como combinaciones de posibilidades. Posteriormente, usando metodología *random 10-folded* (en un problema de 2 clases en el que no hay muestras del mismo "individuo" en ambas clases, no hay inconveniente en usar esta metodología de validación), iterativamente se entrenaron SVM lineales y SVM-rbf con bancos de parámetros C, ϵ y γ distintos por cada iteración. Tal como esperábamos, dependiendo del tipo de datos, separabilidad, forma de los *manifolds* y traslape, los bancos de parámetros obtenidos en los sistemas de mejor clasificación eran distintos. Por ejemplo, con alta separabilidad de los datos, el valor de C fue típicamente grande. Con baja dispersión de los datos, los valores de ϵ y γ normalmente fueron pequeños. Distintas complejidades de los *manifolds* afectaron

[15] La bibliografía acerca del desempeño de las SVM en espacios de muy alta dimensión es bastante escasa y los expertos consultados no tenían solución clara sobre el asunto, de modo que nuestras soluciones son hipótesis sustentadas con nuestras pruebas con datos simulados de alta dimensión.

variablemente los parámetros, pero no obtuvimos conclusiones definitivas acerca de una tendencia clara.

Obtener ajustes finos de los parámetros de las SVM usadas como metaclasificadores en nuestro problema, usando un subconjunto de datos para entrenamiento, es un proceso notablemente más complicado. A diferencia de nuestros datos simulados de alta dimensión, en el problema de expresión facial hay datos muy limitados, incluso clases de las cuales no hay más de 25 muestras (tristeza y miedo). Incluso usando un protocolo de *leave-subjects-out* con 90 % de los individuos por subconjunto de entrenamiento, hay menos de 22 muestras por clase para 2 de las clases y no más de 90 muestras por clase para ninguna clase (86 muestras máximo, para neutral). Debido a esto, usar SVM bastante especializadas (por ejemplo con valores pequeños de ϵ) con transformación no lineal produce clasificadores que se ajustan muy bien a los datos de entrenamiento, pero no muy bien en general. Adicionalmente, hay la opción de usar todos los datos de la base de datos para obtener los valores óptimos de las SVM por cada metaclasificador, con el fin de atenuar un poco el problema de la disponibilidad de datos. Sobre esto no hubo consenso acerca de la viabilidad de esta aproximación: por un lado, usar datos en el entrenamiento que posteriormente serán usados en la validación es un error metodológico de prueba de hipótesis. Por otra parte, en este caso no se trata de un entrenamiento supervisado completamente, como en una reducción de dimensiones supervisada, extracción de parámetros o entrenamiento de sistemas de clasificación, sino únicamente la obtención de parámetros de las SVM. Los expertos consultados tuvieron opiniones diversas acerca de la viabilidad metodológica de esta idea. No obstante, consideramos que extraer parámetros específicos de un metaclasificador forzosamente incurre en error metodológico. Adicionalmente, nuestras pruebas condujeron a resultados indeseados: en numerosas ocasiones los parámetros obtenidos permitían diseñar metaclasificadores aceptables, pero luego en el procedimiento real de *leave-subjects-out* los metaclasificadores entrenados no eran adecuados. En consecuencia, nuestra solución fue establecer los parámetros de las SVM *a priori* y entrenarlas y validarlas sin modificarlos posteriormente. Es posible que haya alternativas más adecuadas, pero en nuestras pruebas no conseguimos ninguna opción satisfactoria tanto en resultados como en metodología viable.

Problemas de sobreajuste: En algunas pruebas usamos la metodología descrita en algunos trabajos [151, 84, 156, 6, 159], tomando varias muestras por individuo por expresión y haciendo validación por *random n-folded*. Un ejemplo es mostrado posteriormente en la tabla 30. Los resultados obtenidos fueron sobresalientes, con tasas de acierto superiores a 98 %. Sin embargo, en tanto que la base de datos usada para esta prueba es la base de datos CK, consideramos que los resultados son producto principal de la metodología de validación usada, incluyendo sobreajuste, y no de un sistema de clasificación aceptable.

Normalmente los problemas de sobreajuste son fácilmente detectados porque los resultados muestran una baja capacidad de generalización. Es decir, el sistema de clasificación se ajusta muy bien a los datos de entrenamiento, pero la prueba con datos

nuevos de validación muestra tasas pobres de clasificación. Sin embargo, al usar 3 muestras por individuo por expresión y validación aleatoria cruzada se ofusca el sobreajuste. Esto ocurre porque al usar validación aleatoria cruzada la probabilidad de que ninguna muestra perteneciente a una tripleta (denominaremos tripletas a muestras de la misma clase, mismo individuo) esté en el conjunto de entrenamiento es ligeramente inferior a 0.1 % (exactamente 0.1 % en conjuntos no finitos). Para una muestra que pertenezca al conjunto de validación, la probabilidad de que alguna de las otras dos muestras pertenezca al conjunto de entrenamiento es ligeramente superior a 99 %. Teniendo en cuenta que las muestras de la misma tripleta tienen parámetros TPOEM (o LBP o cualquier otro tipo de parámetro, para este efecto), si bien la validación aleatoria cruzada no permite entrenar con una muestra que posteriormente haya de ser validada en la misma iteración, para efectos prácticos se está entrenando con una o dos muestras muy parecidas a otras pertenecientes a la validación.

Otro inconveniente típicamente ignorado en la clasificación multiclase es el entrenamiento negativo. Supóngase que en el problema de 7 clases, 6 de las muestras de un individuo A pertenecen al conjunto de entrenamiento (clases I, II, III, IV, V y VI) y la muestra restante (clase VII) pertenece al conjunto de validación. Si los parámetros extraídos permiten una adecuada separación interindividuo (que, por demás, es una característica indeseable en un sistema de reconocimiento de expresión facial), el sistema puede reconocer en el entrenamiento cuáles son las características del individuo A al realizar las expresiones I, II, III, IV, V y VI, de modo que puede fácilmente clasificar la muestra de validación como perteneciente a la clase VII. En una validación aleatoria cruzada con 3 muestras por individuo por expresión generalmente hay un numeroso conjunto de muestras de un individuo en el conjunto de entrenamiento por cada muestra del mismo individuo en el conjunto de validación. En consecuencia el entrenamiento negativo permite aumentar los resultados de clasificación de manera metodológicamente incorrecta.

Dadas las altas dimensiones y el relativamente bajo número de muestras en el conjunto de datos, es posible crear clasificadores super especializados que rompan los datos y los ajusten muy bien, de manera que la clasificación con el conjunto de entrenamiento es muy buena. El peor escenario posible es que el clasificador sea tan especializado que se ajuste bien sin importar la etiqueta de las muestras. Para verificar esto, tomamos un subconjunto de datos y pusimos etiquetas de dos clases de manera aleatoria y posteriormente realizamos entrenamiento de los metaclasificadores SVM 1 vs. 1. El proceso fue repetido varias veces con el fin de descartar los eventos de selección "afortunada", incluso pese a su baja probabilidad. Sorprendentemente, dado un sistema de fusión de clasificadores suficientemente poderoso, se pudieron obtener tasas de clasificación considerablemente buenas (lo que, por otra parte, demuestra cómo usar varias muestras por individuo por expresión puede permitir tasas de clasificación casi perfectas), pero el uso de validación *leave-subjects-out* pone en evidencia la ausencia de *overfitting*, pues los resultados de clasificación son bajos, similares a clasificación aleatoria (lo cual es esperado, pues las muestras fueron etiquetadas aleatoriamente). Esto es soportado por la dimensión VC finita (dimensión Vapnik-Chervonenkis)[166], por cuanto si bien

la dimensión VC se incrementa linealmente con el número de parámetros usados para definir el hiperplano de separación, el número de muestras usadas en los conjuntos de entrenamiento es mayor, de manera que teóricamente no es posible definir un hiperplano sobreajustado para un conjunto de puntos aleatorios usando metodología correcta de validación.

Estas pruebas ilustran que el uso de validación aleatoria cruzada basada en muestras en este tipo de problemas multiclase con muestras del mismo individuo de distintas clases puede tener inconvenientes metodológicos de validación. Nuestra validación LSO elimina estos problemas de validación. Nótese que no se debe confundir con validación *leave-one-subject-out* (LOSO), por cuanto la metodología LOSO toma un individuo del conjunto de datos, entrena con todo el conjunto restante y valida con el individuo excluido. En nuestro caso decidimos no usar LOSO debido a diversas consideraciones: i. LOSO es más adecuado para medir el riesgo del modelo de clasificación y no tanto para medir el desempeño de clasificación [5]. ii. LOSO requiere de gran costo computacional para validar con todos los individuos, pues para ello se debe entrenar y validar un número de sistemas completos de clasificación igual al número de individuos. También nótese que usamos la notación LSO en vez de la común *leave-p-out cross-validation* con el fin de evitar ambigüedad: en *leave-p-out cross-validation* la extracción de muestras no necesariamente es basada en individuos, de modo que no impide la presencia de muestras del mismo individuo y clase en los conjuntos de validación y entrenamiento.

En adición, la metodología LSO puede producir estimaciones más variables del error de predicción que usando una validación aleatoria cruzada. Finalmente, la metodología LOSO puede producir correlación no cero cuando se usan muestras aleatoriamente mezcladas y etiquetadas (esta prueba es usada para evaluar el posible *overfitting* del sistema), lo cual es un atributo indeseable por cuanto el resultado debería cer correlación cero[16].

Traslape y representación asimétrica de las clases: El principal problema del uso de un número reducido de muestras debido al tamaño limitado de la base de datos y la presencia de muestras con fuerte asimetría es la dificultad de representación de las clases. Las SVM son relativamente robustas a la representación asimétrica de las clases si el conjunto de entrenamiento es suficientemente numeroso [132, 36]. Sin embargo, dado el tamaño limitado de la base de datos, el número de muestras no es en general suficiente para prevenir este inconveniente. Adicionalmente, cada celda funciona como un metaclasificador débil. Esto es, la salida del conjunto de metaclasificadores por celda no es muy precisa, pero la confiabilidad del sistema global se basa en el ensamble de los metaclasificadores débiles. Cada celda es un metaclasificador débil debido al fuerte traslape y, en algunos casos, completa imposibilidad de discernimiento entre dos o más clases en una celda específica.

[16]En http://www.russpoldrack.org/2012/12/the-perils-of-leave-one-out.html se encuentra una excelente lectura acerca de los inconveniente metodológicos de LOSO en comparación con otras metodologías de validación [131].

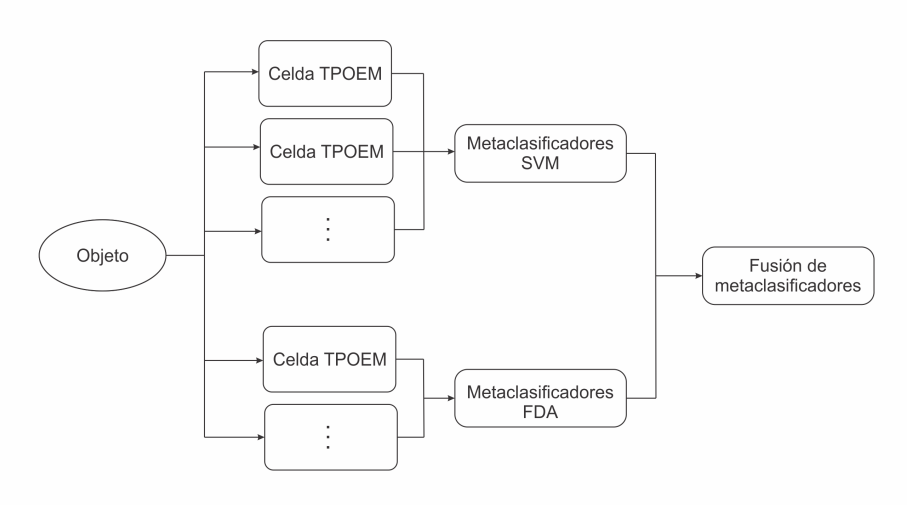
En nuestro problema, el uso de confidencia *a posteriori* con estimación Bayesiana redujo considerablemente el efecto negativo del traslape y la representación asimétrica. Sin embargo, con el fin de probar más detalladamente el efecto del traslape y la subrepresentación de ciertas clases, implementamos un sistema de clasificación con arquitectura 1 vs. 1 por celda, en un caso usando SVM-RBF y en otro caso usando un clasificador más simple con métrica por distancia Mahalanobis. La tasa global de clasificación fue generalmente superior usando SVM, pero en algunos casos la clasificación basada en SVM-rbf tuvo resultados dramáticamente inferiores que usando la aproximación más simple. Al hacer inspección individual de la salida de algunos de los metaclasificadores SVM, se encontró que en numerosas ocasiones la salida siempre fue 0 ó 1, sin importar la muestra usada como entrada. Evaluando iteración por iteración en el entrenamiento de algunas de estas SVM problemáticas, se encontró que esto sucede principalmente debido a dos razones: i. La máquina de soporte vectorial no encontró un plano de separación adecuado, por cuanto los vectores de soporte tenían un nivel tan grande de traslape que era imposible encontrar el hiperplano de separación usando las reglas de optimización. ii. Incluso usando diversos valores del parámetro C, las reglas de optimización deciden que la mejor frontera de separación es aquélla cuya salida es siempre correspondiente a la clase más numerosa, por cuanto otros hiperplanos no disminuyen considerablemente el error de clasificación de la muestra menos numerosa, mientras que sí aumentan notablemente el error de la muestra más representada.

Ajustar de manera fina cada SVM no es una solución idónea por cuanto las pruebas mostraron que esto generalmente conduce a mayor ajuste de las muestras de entrenamiento, a cambio de peor capacidad de discriminación. Por otra parte, se encontró que algunas de las celdas en las cuales la capacidad de discriminación con SVM-RBF fue limitada obtuvieron mejor desempeño en nuestras pruebas anteriores con FDA. Nuestra hipótesis es que esto ocurre principalmente cuando hay mayor asimetría de representación de clases en las muestras de entrenamiento y cuando hay fuerte traslape entre los datos. Determinar el primer evento es simple, en tanto que sabemos el número de muestras por clase por celda usadas en cada metaclasificador 1 vs. 1. El segundo caso es un poco más complejo de ponderar, por cuanto evaluar el traslape de datos en una representación de alta dimensión es un proceso difícil [17]. Nuestra idea de evaluación indirecta del posible traslape fue asumir que los datos tienen una naturaleza Gaussiana y obtener el modelo estadístico para los casos problemáticos comparado con casos de mejor resolución. El uso de técnicas más sofisticadas de estimación paramétrica, tal como la estimación regularizada de parámetros para modelos de mezclas Gaussianas (GMM) en [139] no fue de posible implementación debido al elevado número de parámetros y las muestras limitadas. Sin embargo, haciendo algunas simplificaciones, tales como asumir que la dispersión Gaussiana es isotrópica [18], se puede obtener un mode-

[17] No encontramos en la bibliografía consultada ni en respuestas de expertos en el tema un protocolo de medición de traslape de datos en alta dimensión. La mecánica más sugerida fue evaluar el desempeño de clasificación de 2 clases con clasificadores simples, pero se tornaría una lógica circular, por cuanto precisamente nuestra hipótesis es que la dificultad de clasificación es debida al traslape.

[18] Esta presunción es razonable, pues al hacer modelos Gaussianos simples por cada componente

Figura 38. Clasificación por fusión de metaclasificadores SVM y FDA



lo simplificado. Los resultados mostraron una mayor dispersión del modelo Gaussiano (menor separación entre las medias μ de los modelos por clase o mayor parámetro σ en los casos más complicados) en los casos problemáticos comparados con casos de fácil resolución, lo cual soporta nuestra hipótesis de que el rendimiento de las SVM-rbf fue altamente deteriorado por el traslape entre clases en alta dimensión.

Nuestra alternativa para evitar el inconveniente de la dificultad de las SVM-RBF en estos casos específicos fue eliminarlos del proceso de clasificación y reemplazarlos por clasificadores simples FDA. Como resultado, la arquitectura global de clasificación está mostrada en la figura 38.

Los resultados obtenidos son mostrados en las tablas 26 y 27.

Tabla 26. Reconocimiento de 7 expresiones faciales usando TPOEM y estimación por SVM+FDA, APCC), base de datos CK

	Ira	Dis.	Mie.	Ale.	Tri.	Sor.	Neu.
Ira	**89.0**	3.3	0.4	0.0	0.0	0.0	7.3
Disgusto	1.3	**94.1**	0.4	0.0	0.8	0.0	3.4
Miedo	1.1	1.8	**80.8**	4.7	2.2	0.0	9.4
Alegría	0.0	0.0	1.0	**96.5**	0.0	0.0	2.5
Tristeza	2.5	0.0	0.0	0.0	**82.6**	1.6	13.3
Sorpresa	0.4	0.0	0.9	0.0	0.0	**94.2**	4.5
Neutral	0.0	0.0	0.3	0.0	9.2	0.0	**90.5**

individual de los vectores TPOEM por clase por celda se obtuvo similar σ en las distintas orientaciones.

Tabla 27. Reconocimiento de 6 expresiones faciales usando TPOEM y estimación por SVM+FDA, APCC, base de datos CK

	Ira	Dis.	Mie.	Ale.	Tri.	Sor.
Ira	**94.7**	4.7	0.6	0.0	0.0	0.0
Disgusto	1.8	**95.9**	1.6	0.0	0.7	0.0
Miedo	1.0	0.8	**93.6**	2.5	2.1	0.0
Alegría	0.0	0.0	1.4	**97.9**	0.0	0.7
Tristeza	1.7	0.0	0.0	1.4	**95.4**	1.5
Sorpresa	0.0	0.0	0.0	0.5	0.0	**99.5**

La tasa global de reconocimiento de 7 clases es 89.7 % y de 6 clases es 96.16 %. Si bien en apariencia no es una mejoría notable comparado con los resultados iniciales de 85.34 % y 93.3 % respectivamente [19], hay dos limitaciones importantes que dificultan mejorar los resultados a partir de cierto umbral. Primero, la base de datos CK tiene aproximadamente un 2 % de las muestras que posteriormente fueron eliminadas de la base de datos CK+ por su pobre representación o su etiqueta incorrecta. En tanto que estas muestras fueron usadas en nuestro trabajo, para evitar error metodológico al escoger manualmente cuáles muestras utilizar y cuáles no, producen otra suerte de inconvenientes: i. su clasificación es "errada" (es decir, son etiquetadas aparentemente de manera incorrecta, pero en realidad su etiqueta posiblemente sea cercana a la etiqueta de clasificación) y ii. estas muestras defectuosas hacen parte del conjunto de entrenamiento en diversas iteraciones, de manera que los clasificadores son entrenados con muestras que no corresponden a la etiqueta asignada. El segundo problema es que mejorar tasas de clasificación ya tan elevadas es un problema difícil, por cuanto, si se observa desde la perspectiva del error, una mejora de 85.3 % a 89.7 % es equivalente a reducir el error en aproximadamente un 30 %.

6.3.6. Cooperación mutua entre parámetros locales

En las arquitecturas previas cada celda espacial aporta información de clasificación de manera individual, independientemente de las demás. Si bien esto es una aproximación adecuada en términos de costo, desprecia la contribución importante constituída por la información mutua entre las celdas. Una manera simple de explicar la relevancia de la información mutua es al considerar que 2 expresiones pueden tener patrones similares en las regiones superiores del rostro, tales como el miedo y la tristeza, debido a los gestos del ceño. Usando las metodologías previamente reseñadas, los puntajes en estas regiones son fuertemente disputados entre estas dos expresiones. En las regiones

[19] No obstante, en comparación con la prueba anterior, en la prueba t-test por pares se obtiene $p = 0{,}048$, que aún indica mejoría con $\alpha = 5\,\%$.

inferiores del rostro, para una entrada de tristeza, los puntajes son fuertemente disputados entre tristeza y neutral, debido a la pequeña activación de músculos de la boca en expresiones de tristeza, mientras que para una entrada de miedo hay cierto nivel de disputa entre miedo y alegría. Debido a esto, estas dos expresiones son clasificadas con cierta frecuencia como neutral y alegría respectivamente (obsérvese en las matrices de confusión previas, así como en bibliografía de clasificación de expresión cómo estos dos errores son contrubuciones muy importantes al error global de clasificación). Una metodología que incluya información mutua entre los parámetros espaciales puede reducir notablemente este tipo de error, por cuanto la clasificación no es una suma ponderada simple, sino es una clasificación compuesta, con resultados condicionales en función de una mezcla de metaclasificadores.

El uso de definiciones convencionales de información mutua no es posible en nuestro problema, por cuanto no hay una cantidad de datos suficiente par construir un modelo de distribución probabilística conjunta adecuado. Es decir, los modelos de distribución de probabilidad serían pobremente construidos. Además, el alto número de parámetros hace que la evaluación de la entropía multidimensional sea una tarea prohibitiva, por cuanto la estimación de un modelo GMM require de un número de muestras que permita representar el espacio n-dimensional. Con un espacio n-dimensional elevado, el número de muestras requerido aumenta exponencialmente, a valores muy superiores que los disponibles en la base de datos. Por último, la clasificación que se base en una colaboración entre cualquier par de combinaciones de parámetros es un problema costoso y extenso.

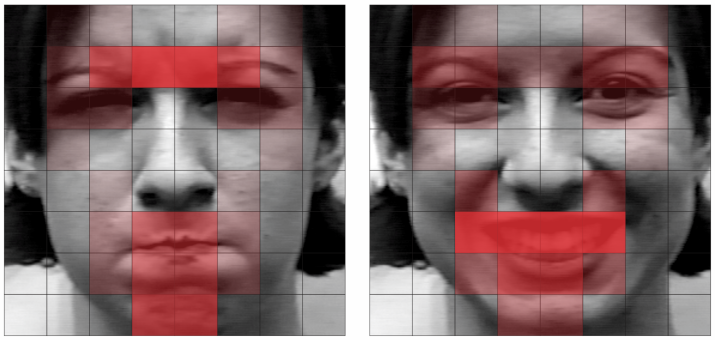
Sin embargo, hay otras técnicas menos costosas que introducen la cooperación entre celdas sin requerir del cálculo complicado de información mutual. La idea que usamos fue basada en la resolución de una cuestión simple pero frecuentemente desdeñada en los problemas de clasificación: qué hace a la clase A distinta de la clase B desde el punto de vista de los parámetros? Es decir, necesitamos encontrar las diferencias entre dos o más clases, especialmente en expresiones de difícil clasificación. Una prueba preliminar para determinar la validez de nuestra hipótesis fue realizada usando un subconjunto de entrenamiento para entrenar el conjunto de metaclasificadores y otro subconjunto disjunto para obtener la salida de este banco de metaclasificadores, y con los datos obtenidos una clasificación ponderada fue realizada para obtener cl, tal como se muestra en la ecuación 6.22.

$$X_{k,c}^{v} \rightarrow z_{i,k,c1,c2} \tag{6.22}$$

$$cl_{i,k,c} = \left(2 * \sum_{c2} z_{i,k,c1,c2} - 1\right)[c1 = c] - \left(2 * \sum_{c1} z_{i,k,c1,c2} - 1\right)[c2 = c] \tag{6.23}$$

$cl_{i,k,c}$ es un indicador de la factibilidad de los clasificadores 1 vs. 1 por celda de ganar los concursos individuales usando un subconjunto de validación. Posteriormente definimos ncl como el valor normalizado de las clases por celda por expresión, según la ecuación 6.24.

Figura 39. Visualización de la confiabilidad de celdas espaciales TPOEM en la clasificación de ira y alegría



$$ncl_{k,c} = \frac{1}{N}\sum_{i=1}^{N} cl_{i,k,c} \tag{6.24}$$

También se puede realizar una estimación similar usando una aproximación Bayesiana, tal como se muestra en la ecuación 6.25.

$$ncl'_{k,c} = p(z_{i,k,c,c_2} = 1 \mid x), \quad c_2 \neq c, \quad i = 1, ..., N \tag{6.25}$$

En este caso la salida ncl' cuantifica la probabilidad de victoria de cada clasificador 1 vs. 1 individual, asumiendo que las muestras x son tomadas de un subconjunto de datos disjunto al conjunto de entrenamiento. La comparación experimental entre ncl y ncl' fue similar, salvo un *offset* en ncl debido a que el número de muestras pertenecientes a las clases 1 y 2 en la ecuación 6.22 no es normalizado, pero el cálculo de ncl' en 6.25 es considerablemente más extenso, de manera que las estimaciones completas fueron realizadas usando ncl.

En la figura 39 se muestra la visualización espacial promedio de ncl para las expresiones de alegría y de ira, cuyos parámetros espaciales más relevantes para la determinación de la expresión facial se encuentran en las regiones inferiores y superiores del rostro respectivamente.

En la figura 39 el color es más intenso dependiendo del valor de ncl. Proporcionalmente más clasificadores 1 vs. 1 correspondientes a la mitad inferior del rostro tienen más posibilidad de ganar los concursos individuales en la expresión de alegría, mientras que en la expresión de ira los parámetros de mayor probabilidad de ganar los concursos 1 vs. 1 son los parámetros correspondientes al ceño. Como tal, nuestra hipótesis original fue confirmada. Consecuentemente, el problema es cómo integrar esta información en el sistema de clasificación. La mayor parte de trabajos que usan la división del rostro en regiones o celdas no incluyen información de cooperación mutua entre celdas, sino que generalmente la clasificación es realizada como la suma o suma poderada de las métricas o puntajes por celda. Otros trabajos usan celdas espaciales con traslape, lo

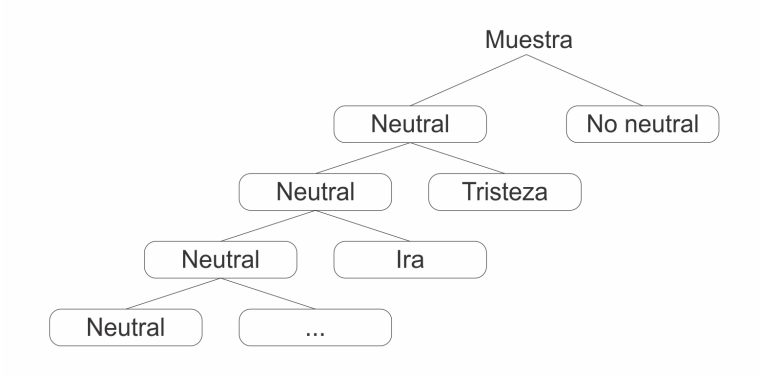
cual indirectamente proporciona información mutua, pero restringida a regiones en una vecindad espacial limitada. Por otra parte, incluir la definición formal de información mutua entre celdas por expresión fue altamente variable dependiendo del subconjunto de entrenamiento. Esto es razonable, por cuanto la determinación precisa de la información mutua de parámetros de alta dimensión no es posible con un conjunto limitado de datos, sobre todo para clases con baja representación en la base de datos. En consecuencia, se eligió usar una aproximación por árboles Bayesianos NVTree (*naïve Bayesian Trees* [102, 17]). Nuestros clasificadores previos mostraron que algunas expresiones son de relativamente fácil clasificación, principalmente alegría, sorpresa, neutral y disgusto, mientras que ira, miedo y tristeza tienen generalmente problemas de discriminación. Por fortuna, nuestras funciones de castigo hacen que la mayor parte de clasificaciones erradas sean de una expresión clasificada como neutral en vez de una expresión clasificada como otra expresión [20]. Debido a esto, una aproximación por árboles Bayesianos puede incluir esta información relevante: una muestra clasificada como expresión normalmente tiene muy bajo porcentaje de error (4.8 % de probabilidad de error de una muestra clasificada como expresión en la tabla 26), mientras que una muestra clasificada como neutral tiene considerable probabilidad de en realidad pertenecer a una expresión (36 % en la tabla 26). Usando los resultados de las pruebas preliminares de *ncl* en las ecuaciones 6.22 y 6.24, encontramos los parámetros más relevantes por expresión distinta a la instancia neutral. Los datos mostraron que los parámetros más relevantes para las expresiones de difícil clasificación son localizados en la región superior del rostro, lo cual es razonable debido a la característica expresión de ceño presente en las expresiones de ira, miedo y disgusto. Posteriormente usamos una aproximación Gaussiana *naïve*. Para ello, definimos la variable continua S_{i,k,c_1,c_2} como un puntaje entre 0 y 1 para la muestra i, celda k, concurso 1 vs. 1 entre las clases c_1 y c_2, determinado por la distancia entre la muestra y el hiperplano de separación con una función sigmoidea. Esto es, un puntaje 1 corresponde a clasificación clase 1, máxima distancia entre la muestra y el hiperplano y un puntaje 0 corresponde a clasificación clase 2, también distancia máxima entre la muestra y el hiperplano. Normalmente la aproximación Gaussiana hace discretización de las variables continuas en *bins*, en [38]. Sin embargo, numerosos estudios más recientes han mostrado que esta discretización que asume que la distribución anterior (*prior distribution*) es de naturaleza Dirichlet puede conducir a una reducción de capacidad de discriminación entre clases [78, 64]. Debido a ello, nuestra aproximación Gaussiana fue realizada según lo mostrado en la ecuación 6.26.

$$p(S_{i,k,c_1,c_2} > h|c_1) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(S_{i,k,c_1,c_2}-\mu)^2}{2\sigma^2}} \tag{6.26}$$

El valor h fue definido manualmente en 0,5 y se usó un subconjunto disjunto de

[20] Por ejemplo, los resultados en 26 muestran que el porcentaje de error de una expresión clasificada como neutral conforma el 64.33 % del error total, mientras que el valor esperado si el error fuese indiscriminado sería únicamente el 16.16 %. Como referencia, el porcentaje de error de clasificación de una muestra de expresión clasificada como neutral en algunos trabajos del estado del arte es: 33.38 % en [151], 31.58 % en [1], 45.04 % en [122] y 7.32 % en [156].

Figura 40. Arquitectura del árbol Bayesiano de clasificación



entrenamiento para obtener los valores de σ y μ de la estimación Gaussiana (es decir, cada clasificador 1 vs. 1 tiene sus propios valores de parámetros Gaussianos σ y μ).

Una vez obtenidas las estimaciones Gaussianas, se construyó el árbol Bayesiano *naïve* de manera recursiva así: i. La clasificación de una muestra como neutral tiene la mayor tasa de error, entonces es el primer nodo del árbol *naïve*. ii. Determinar la probabilidad de que la muestra pertenezca a tristeza (mayor error de clasificación de una muestra clasificada como neutral) usando los modelos Gaussianos 1 vs. 1 ponderados por los valores de *ncl* obtenidos previamente. iii. La otra rama del árbol se bifurca en un menor nivel, correspondiente a la estimación de la probabilidad de que la muestra pertenezca a miedo (segundo mayor error de clasificación de muestra clasificada como neutral). iv. Repetir sucesivamente, construyendo las ramas restantes del árbol.

Si una muestra no es clasificada como neutral la probabilidad de error de clasificación es muy pequeña, de modo que el árbol en esta variante no es frondoso y sólo tiene en cuenta algunos casos específicos de error considerable (no mostrados en la figura). Con ello evitamos problemas de falta de generalización cuando el árbol es diseñado de manera muy ajustada a las variables, tal como se muestra en [16]. La arquitectura general del árbol Bayesiano es mostrada en la figura 40.

La inclusión de los árboles Bayesianos representó una mejora notable en el desempeño de clasificación, por cuanto permitió reducir la ambigüedad de clasificación de algunas expresiones. Por otra parte, esta técnica incrementó ligeramente el error de clasificación de una muestra clasificada como expresión pese a que la instancia era neutral. Sin embargo, consideramos que este pequeño aumento de error no es notable dada la reducción de error global. Los resultados se muestran en las tablas 28 y 29.

6.3.7. Pruebas de validación para verificar la eliminación de clasificadores sobre ajustados

En esta subsección mostraremos cómo el uso de validación aleatoria cruzada basada en muestras en el problema de expresión facial con múltiples muestras por individuo por expresión puede conducir inadvertidamente a clasificadores sobre ajustados. Su-

Tabla 28. Reconocimiento de 7 expresiones faciales usando TPOEM, SVM+FDA, APCC + árboles Bayesianos, base de datos CK

	Ira	Dis.	Mie.	Ale.	Tri.	Sor.	Neu.
Ira	**90.2**	2.8	0.3	0.0	0.0	0.0	6.7
Disgusto	1.1	**94.7**	0.2	0.0	0.8	0.0	3.2
Miedo	1.2	1.5	**81.8**	4.2	2.1	0.0	9.2
Alegría	0.0	0.0	1.2	**96.9**	0.0	0.0	1.9
Tristeza	2.2	0.0	0.0	0.0	**86.3**	1.7	9.8
Sorpresa	0.0	0.0	0.7	0.0	0.0	**96.2**	3.1
Neutral	0.0	0.0	0.0	0.0	9.0	0.0	**91.0**

Tabla 29. Reconocimiento de 6 expresiones faciales usando TPOEM, SVM+FCA, APCC + árboles Bayesianos, base de datos CK

	Ira	Dis.	Mie.	Ale.	Tri.	Sor.
Ira	**95.2**	4.3	0.5	0.0	0.0	0.0
Disgusto	2.1	**96.0**	1.5	0.0	0.4	0.0
Miedo	0.8	0.5	**94.1**	2.2	2.4	0.0
Alegría	0.0	0.0	0.9	**98.4**	0.0	0.7
Tristeza	1.9	0.0	0.0	1.5	**95.4**	1.2
Sorpresa	0.0	0.0	0.0	0.0	0.0	**100.0**

pongamos un problema de 2 clases en los cuales hay N muestras en el conjunto de datos, correspondientes a $N/2$ individuos (una muestra por individuo por clase). Dado N de suficiente tamaño, en una metodología de validación aleatoria cruzada basada en muestras con 10 pliegues la probabilidad de que las dos muestras de clases I y II pertenecientes a un individuo dado pertenezcan al conjunto de validación es aproximadamente 1 %. Para $N \to \infty$, la probabilidad de al menos una de las muestras por individuo perteneciente al conjunto de entrenamiento es de 99 %. Como consecuencia, es posible que el sistema de clasificación "aprenda" las clase A para un individuo dado (muestra perteneciente al entrenamiento) y si la muestra de la clase B perteneciente al mismo individuo está en el conjunto de validación, el clasificador sabe que la muestra no pertenece a la clase A, de modo que la clasifica, correctamente, como perteneciente a la clase B. Este problema persiste en un problema de clasificación de múltiples clases, incluso en mayor extensión, por cuanto para un individuo dado probablemente existen varias muestras representadas en el subconjunto de entrenamiento, de modo que un clasificador sobre ajustado sólo debe decidir entre un número limitado de salidas.

Numerosos trabajos usan metodología aleatoria cruzada basada en muestras para la validación, incluso usando varias muestras por individuo por expresión. Por ejemplo, algunos trabajos de reconocimiento de expresión facial usando SVM como mecanismo de clasificación han mostrado resultados altamente satisfactorios. En [151] se reporta 88.9 % de clasificación de 7-clases usando codificación LBP y clasificación SVM-RBF con la base de datos CK con validación *random 10-folded*. En [84] se obtiene 99.7 % de clasificación con SVM multiclase en la base de datos CK, validación *random 10-folded*. En [156] se obtiene 94.62 % de clasificación con la base de datos CK usando SVM, *random 10-folded*. Es decir, en general los resultados usando SVM mostrados en la literatura no son consistentes con nuestros resultados usando *leave-subjects-out*. En consecuencia, decidimos hacer una prueba con los clasificadores SVM+FDA con árboles Gaussianos, usando esta vez 3 muestras por individuo por expresión, con validación *random 10-folded*. Los resultados son mostrados en la tabla 30.

Tabla 30. 7-Expression Classification using TPOEM Codification with SVM-RBF Classification, random 10-folded validation

	Ira	Dis.	Mie	Ale	Tri.	Sor.	Neu.
Ira	**98.77**	0.63	0.50	0.00	0.00	0.00	0.00
Dis.	0.98	**99.02**	0.00	0.00	0.00	0.00	0.00
Mie.	0.31	0.0	**98.56**	1.13	0.00	0.00	0.00
Ale.	0.00	0.00	0.00	**100.00**	0.00	0.00	0.00
Tri.	0.00	0.00	0.00	0.00	**97.72**	0.00	2.28
Sor.	0.00	0.00	0.11	0.00	0.00	**99.89**	0.00
Neu.	0.00	0.00	0.00	0.00	0.07	0.00	**99.93**

Estos resultados ilustran cómo el uso de metodología de validación *random n-folded* puede producir tasas de clasificación muy elevadas dado un sistema de clasificación suficientemente complejo, muy superiores que los resultados con validacion LSO. Nuestra explicación de la discrepancia debido a posible sobre ajuste inadvertido de los clasificadores es razonable, y es consistente con la bibliografía que muestra en general mejores resultados con *random n-folded* que con protocolos de validación que excluyan individuos completos del entrenamiento.

Por ejemplo, además de la ya previamente mencionada discrepancia de resultados de acuerdo con la metodología de validación en [156], hay numerosos ejemplos en la bibliografía que ilustran este mismo asunto. En [198] la tasa de clasificación usando la base de datos JAFFE y validación *Leave-one-subject-out* es de 74.32 %, mientras que usando *leave-one-image-out*, que implica que el sistema es entrenado por muestras de otras clases pertenecientes al mismo individuo, la tasa de clasificación es de 85.79 %. En [174] la tasa de clasificación usando clasificación LDA con la base de datos CK y validación *person independent* produjo resultados globales de 82.68 %, en tanto que la validación *person dependent* [21] produjo resultados de 87.27 %.

[21] Dependiendo de la definición, *person dependent* puede referirse a la validación usando muestras

6.3.8. Clasificación basada en *Deep learning*

Deep learning [10] es una alternativa interesante para proporcionar incorporación de la información proporcionada por cada parámetro. En *deep learning* convencional para clasificación basada en imágenes la clasificación es generalmente realizada en el nivel de pixeles. Las entradas de la red profunda (*deep network*) o las redes convolucionales son los pixeles de la imagen o una región de la imagen. Esta aproximación es útil para el reconocimiento de patrones pequeños, por ejemplo el reconocimiento de los dígitos manustritos de la base de datos MNIST

No obstante, adaptar las ideas generales del aprendizaje profundo puede ser una aproximación adecuada, debido a la manera como los humanos y las máquinas de aprendizaje profundo desarrollan clasificación conceptual [144, 177, 200]. Es así que los humanos realizan reconocimiento de la expresión facial al evaluar patrones particularmente distinguibles en la región facial, especialmente en la boca, ojos y ceño, y asociándolos entre ellos [51, 65].

Nuestra aproximación fue similar a *Deep learning* en la manera como éste aborda el problema como la solución de distintos micro problemas conectados entre sí. El reconocimiento de la expresión facial con patrones localizados espacialmente puede ser entendido como el ensamble de la información proporcionada por los distintos parámetros espaciales para resolver el problema principal. Usando una arquitectura basada en *Deep learning*, el problema puede ser descompuesto de manera que el sistema de clasificación obtenga respuestas de problemas jerárquicos parciales y desarrolle la solución general. Esta arquitectura de aprendizaje profundo crea cuestiones abstractas que permitan componer la estructura completa a partir de cuestiones parciales.

En los sistemas de clasificación previos basados en arquitectura 1 vs. 1, se obtuvieron 21 salidas por celda TPOEM. Típicamente la salida es binaria, pero puede ser un puntaje entre 0 y 1 para cada una de las expresiones involucradas en el concurso de acuerdo a la distancia entre la muestra y la frontera de clasificación. Una vez los puntajes por expresión por concurso son obtenidos, una red con una primera capa compuesta por 7×7 subcapas, una por clase, es diseñada. Cada subcapa es un autoencoder apilado (*stacked autoencoder*) [12]. Cada una de estas capas paralelas tiene como objetivo evaluar la probabilidad de pertenencia de una muestra a una clase particular dados los puntajes por clase. Consecuentemente, las entradas para las neuronas en los autoencoders son los puntajes individuales de cada clase. Usamos la notación S_{k,e_1,e_2,e_1} para definir el puntaje de la expresión e_1 en el concurso e_1 vs. e_2 para el parámetro k. Los puntajes definidos pueden ser los valores probabilísticos usando los valroes de distancia entre la clasificación y el hiperplano de separación mediante escalamiento Platt [130] 6.27.

$$S(y|X) = \frac{1}{1 + exp(\alpha * f(x) + \beta)} \tag{6.27}$$

del mismo individuo pero distintas clases en el conjunto de entrenamiento o incluso usar muestras del mismo individuo y la misma clase en el conjunto de entrenamiento, de modo que es técnicamente comparable a la validación *random n-folded*.

$f(x)$ es la distancia con signo entre la muestra y el hiperplano y α y β son los parámetros que deben ser optimizados. Sin embargo, no usamos esta aproximación porque el cálculo de la probabilidad *a posteriori* con escalamiento Platt requiere de costosas pruebas *n-folded* y es sujeto, por tanto, a sobre ajuste. En vez de eso, usamos un puntaje directamente basado en la distancia al hiperplano. Dada $f(x, k, e_1, e_2)$ la distancia con signo entre la muestra x y el hiperplano para el clasificador e_1 vs. e_2 en la celda k, el puntaje fue obtenido usando una función logística, según la ecuación 6.28.

$$S_{k,e_1,e_2,e_1}(x) = \frac{1}{1 + exp(-\alpha * f)} \tag{6.28}$$

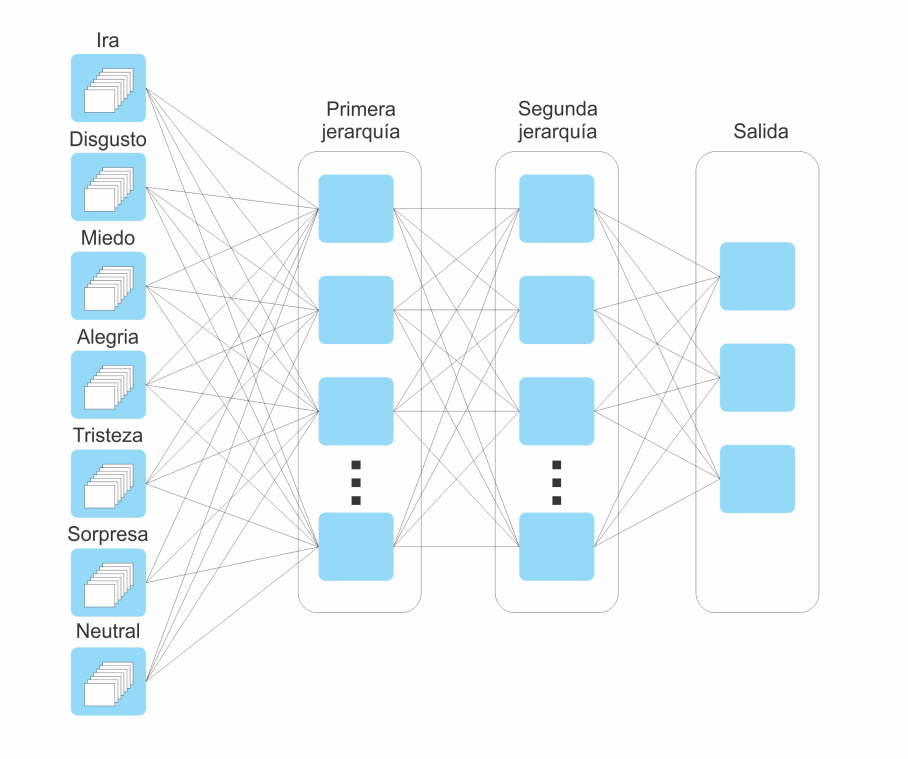
El parámetro α fue manualmente definido en 3, de manera que una distancia normalizada de 1 (esto es, la muestra está a la misma distancia de la frontera que el promedio de las muestras de la clase en el conjunto de entrenamiento) produce un puntaje $S(f) = 0{,}9526$ para la expresión ganadora y un puntaje $S(f) = 0{,}0474$ para la expresión perdedora (la función logística es simétrica); para una distancia normalizada 0.2, que es una muestra relativamente cercana a la frontera, la función logística produce un puntaje de $S(f) = 0{,}6457$ para la expresión ganadora y un puntaje $S(f) = 0{,}3543$ para la expresión perdedora. Es decir, el puntaje refleja la contundencia de la victoria en el concurso 1 vs. 1, de modo que no penaliza tan fuertemente una expresión perdedora en un concurso disputado.

Las 7 subcapas para el problema de reconocimiento de 7 expresiones son paralelas. Cada capa tiene sus propias entradas, referidas a la expresión a ser evaluada por la subcapa. Con esta arquitectura esperamos que cada subcapa en la primera capa se especialice en la evaluación de sus propios parámetros TPOEM locales, descartando la información concerniente a las otras subcapas. La tarea de la segunda capa es actuar como el enlace de comunicación entre los autoencoders de expresión precedentes. En lenguaje natural, el objetivo de esta capa es ensamblar la evaluación de las subcapas de expresión para intentar obtener una adecuada clasificación. Finalmente, la capa de salida es realizada por 7 neuronas, una por clase. La arquitectura general es mostrada en la figura 41.

Cada subcapa de la primera capa es entrenada como un autoencoder disperso. Esto es, las salidas de las subcapas están diseñadas para ser parámetros primarios de la entrada por subcapa y se espera que la representación de menor dimensión de la información de expresión esté incluida en los parámetros de entrada. La siguiente capa ejecuta las conexiones entre estos autoencoders y fue entrenada usando una aproximación *greedy layer-wise*. En teoría, esta capa debe proporcionar otra descomposición jerárquica del problema. Después hay otra etapa con menor número de neuronas, correspondiente a una segunda descomposición jerárquica. Finalmente, hay una capa softmax que discrimina las expresiones y la instancia neutral. Con el fin de probar la clasificación, usamos validación LOSO en vez de la metodología anterior debido a la necesidad de usar el mayor número posible de muestras para entrenar existosamente los clasificadores *deep learning*.

Por otra parte, LOSO tiene un inconveniente. En una validación dejando un 10 %

Figura 41. Arquitectura del sistema de aprendizaje profundo usado para clasificación de la expresión facial



de las muestras para validación por iteración, se requiere de 10 iteraciones para validar todas las muestras. En el caso de LOSO, en cada iteración se valida únicamente 1 individuo, que por lo general corresponde a evaluar 3-4 expresiones (no 7, por cuanto el promedio de expresiones por individuo en la base de datos es 3-4). Más aún, la complejidad de los algoritmos de reducción de dimensiones, selección de parámetros y entrenamiento hacen que una iteración completa tenga un costo computacional de promedio 4 horas en nuestro sistema. Este tiempo no es problemático para la clasificación, que es computacionalmente rápida, pero cada validación de 4 horas promedio, para 3-4 expresiones, hace que incluso con 125 horas de procesamiento, más que una semana de trabajo regular, no se pueda validar más que una tercera parte de las muestras.

En tanto que no es posible tener una estimación precisa de la mejor configuración de las etapas intermedias y la primera capa debe ser diseñada principalmente intuitivamente, el proceso debe ser realizado diversas ocasiones, modificando ligeramente la arquitectura global del sistema y el número de neuronas y funciones de activación de cada subcapa. Éste es otro importante problema. Algunas expresiones no tienen más que una fracción de muestras en la base de datos. Esto significa que incluso si se hace un número n de validaciones, en promedio sólo hay $0{,}22 \times n$ muestras de miedo y $0{,}25 \times n$ muestras de tristeza validadas. Esto implica que luego de 50 validaciones, únicamente 10 muestras de miedo y 11 de tristeza han sido validadas en promedio. Desde una perspectiva estadística, estos bajos valores dificultan la diferenciación del desempeño entre 2 clasificadores. Esto puede ser mostrado con un ejemplo numérico. Considérese un hipotético clasificador A cuya precisión en la evaluación de miedo es de 90 % usando 40 muestras y un clasificador B cuya precisión es 95 % usando 20 muestras. En estos casos los intervalos Bayesianos de confianza 95 % están mostrados en la tabla 31.

Tabla 31. Intervalo de confianza bayesiana para dos clasificadores

	Tasa de clasificación	**Límite inferior**	**Límite superior**
Clasificador A	0.9	0.769	0.959
Clasificador B	0.95	0.762	0.988

Según los resultados de la tabla 31, se sugiere que el clasificador B es mejor que el clasificador A, pero la certeza estadística de esta presunción no es muy elevada. En las etapas anteriores de entrenamiento y validación este problema fue atenuado al realizar numerosas etapas de LSO, por cuanto el costo de entrenamiento de los clasificadores no era considerable. En este caso el tiempo de procesamiento es prohibitivo, de modo que las pruebas no incluyen tantos casos de validación, lo que hace que los intervalos estadísticos de confianza sean considerablemente mayores. De esta forma, no es posible afirmar con certeza estadística que los resultados de clasificación realizada por *Deep learning* sean superiores que usando las técnicas precedentes, pero los resultados individuales de clasificación por expresión, así como la clasificación global, sugieren que éste es efectivamente el caso. Los resultados finales de clasificación de 7 expresiones y de 6 expresiones son mostrados en las tablas 32 y 33.

Tabla 32. Reconocimiento de 7 expresiones faciales usando TPOEM, SVM+FDA, APCC + *deep learning*, base de datos CK

	Ira	Dis.	Mie.	Ale.	Tri.	Sor.	Neu.
Ira	**90.5**	2.2	0.2	0.0	0.0	0.0	7.1
Disgusto	0.5	**94.6**	0.0	0.0	0.6	0.0	4.3
Miedo	1.4	1.5	**82.6**	3.9	2.0	0.0	8.6
Alegría	0.0	0.0	0.0	**98.8**	0.0	0.0	1.2
Tristeza	2.0	0.0	0.0	0.0	**84.4**	1.8	11.8
Sorpresa	0.0	0.0	0.8	0.0	0.0	**96.3**	2.9
Neutral	0.0	0.0	0.0	0.0	10.3	0.0	**89.7**

Tabla 33. Reconocimiento de 6 expresiones faciales usando TPOEM, SVM+FDA, APCC + *deep learning*, base de datos CK

	Ira	Dis.	Mie.	Ale.	Tri.	Sor.
Ira	**95.1**	4.1	0.8	0.0	0.0	0.0
Disgusto	1.7	**96.2**	1.6	0.0	0.5	0.0
Miedo	1.1	0.4	**94.3**	1.8	2.4	0.0
Alegría	0.0	0.0	1.0	**98.3**	0.0	0.7
Tristeza	1.4	0.0	0.0	1.4	**95.9**	1.3
Sorpresa	0.0	0.0	0.0	0.0	0.0	**100.0**

Estos resultados sugieren que el uso de esta metodología de clasificación produce mejor reconocimiento de las distintas clases, comparado con los clasificadores usados previamente [22]. La adición de un sistema basado en aprendizaje profundo debería hacer un mejor trabajo debido a su capacidad de integración de información de distintas categorías jerárquicas, tal como muestran los resultados. Sin embargo, aún quedan algunas cuestiones cuya resolución no es posible con la base de datos limitada. Hasta ahora, uno de los principales inconvenientes es que la partición de la base de datos entre entrenamiento y validación (incluyendo en el entrenamiento todas las etapas supervisadas de reducción de dimensiones y extracción de parámetros) aumenta la independencia entre el clasificador y las muestras de validación; sin embargo, requiere de un costo

[22]No obstante, en este caso no es posible afirmar categóricamente que hubo mejoría estadística. La prueba t-test produjo $p = 0{,}381$, que es un valor muy elevado y de ninguna manera permite determinar que efectivamente hay diferencia estadística entre las pruebas. Esto ocurre debido a que a diferencia de las pruebas anteriores, el número de validaciones con *deep learning* fue muy limitado. Esto, sumado a la pequeña diferencia de resultados entre las pruebas, impide obtener un valor de p que brinde más confinaza estadística.

computacional grande en la medida en que los sistemas tienen mayor complejidad. Adicionalmente, el tamaño limitado de las bases de datos hace que sea complicado determinar si el sistema cuenta con adecuada generalización, debido al relativamente grande valor de rango de confianza estadística. Algunos de estas cuestiones serán abordadas en el capítulo 7, que incluye pruebas dinámicas de expresión facial y pruebas de generalización.

6.4. Comparación de resultados con trabajos similares del estado del arte, incluyendo bases de datos CK y CK+

En la sección 6.3 se mostró el desarrollo y resultados usando la base de datos CK. Esto se hizo porque la mayor parte de trabajos del estado del arte consultado se ha realizado con esta base de datos. Realizar una comparación directa entre resultados obtenidos con la base de datos CK y resultados obtenidos con la base de datos CK+ no es muy riguroso, por cuanto en esta última se eliminó un conjunto sustancial de muestras de etiqueta incorrecta o pobre representación, de manera que es natural que se puedan obtener mejores resultados incluso con parámetros o sistemas de clasificación que no necesariamente sean superiores. De acuerdo con esta consideración, en nuestro trabajo principalmente consignamos el desarrollo teórico y experimental usando la base de datos CK, pero de manera paralela en cuanto obtuvimos autorización de uso de la base de datos CK+ se hizo el desarrollo incluyendo esta versión extendida. Es dispendioso consignar todas las matrices de confusión de resultados usando la base de datos CK+, por cuanto en general la tendencia es similar a los resultados con la base de datos CK, con mejores resultados de clasificación en cuanto los sistemas fueron desarrollados de manera más sofisticada y rigurosa. En esta sección nos enfocaremos a usar los resultados más recientes de clasificación de estas bases de datos usando la metodología que probó ser más poderosa en la sección precedente, usando codificación TPOEM, metaclasificación débil con SVM y FDA, estimación ponderada APCC y ensamble con una máquina deep learning. Así mismo, incluiremos resultados relevantes de comparación, principalmente con metodologías de parámetros similares a LBP, pero también con metodologías basadas en descriptores geométricos cuyos resultados han probado ser bastante exitosos.

En cuanto ya consignamos previamente las matrices de confusión usando la base de datos CK, en primer lugar mostramos los resultados de clasificación de 7 y 6 clases usando la base de datos CK+, en las tablas 34 y 35.

A continuación mostraremos los resultados comparado con trabajos recientes del estado de desarrollo del campo. En la figura 42 se observa la comparación de clasificación entre nuestros resultados y los obtenidos con otras metodologías para 6 y 7 clases.

Los trabajos usados para la comparación son detallados así: textura y forma en [85], psicovisual en [82], SPTS+CAPP en [98], VLBP+LBP-TOP en [193], LBP+SVM-RBF en [151], curvelet+LBP en [142], AAM en [156], LBP+KDI en [196], dynamic Haar-like en [183], eigenespacio PCA en [115], deformación geométrica en [84], SVM + AdaBoost

Tabla 34. Reconocimiento de 7 expresiones faciales usando TPOEM, SVM+FDA, APCC + *deep learning*, base de datos CK+

	Ira	Dis.	Mie.	Ale.	Tri.	Sor.	Neu.
Ira	**94.2**	0.3	0.0	0.0	0.1	0.0	5.4
Disgusto	0.0	**94.7**	0.5	0.3	0.5	0.0	4.0
Miedo	0.0	0.5	**91.0**	0.0	1.1	0.6	6.8
Alegría	0.0	0.0	0.0	**99.9**	0.0	0.0	0.1
Tristeza	0.4	0.0	0.4	0.0	**89.9**	0.0	9.3
Sorpresa	0.0	0.0	0.7	0.0	0.0	**96.2**	3.1
Neutral	0.6	0.1	0.3	0.1	4.5	0.0	**94.4**

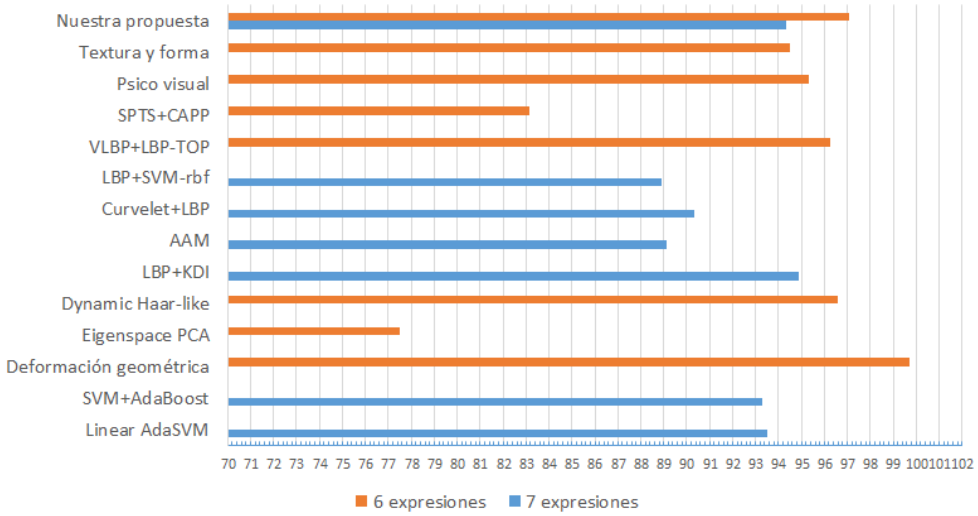
Tabla 35. Reconocimiento de 6 expresiones faciales usando TPOEM, SVM+FDA, APCC + *deep learning*, base de datos CK+

	Ira	Dis.	Mie.	Ale.	Tri.	Sor.
Ira	**96.2**	3.5	0.0	0.0	0.3	0.0
Disgusto	1.6	**96.0**	1.6	0.0	0.8	0.0
Miedo	1.0	0.0	**94.5**	1.9	2.6	0.0
Alegría	0.0	0.0	0.0	**100.0**	0.0	0.6
Tristeza	1.4	0.0	1.1	0.3	**96.2**	1.0
Sorpresa	0.0	0.0	0.6	0.0	0.0	**99.4**

en [7] y Linear AdaSVM en [92]. Se puede observar que en general nuestro sistema tiene gran tasa de acierto en reconocimiento de 6 clases y reconocimiento adecuado en reconocimiento de 7 clases. De hecho, este rendimiendo sobresaliente en reconocimiento de 6 clases comparado con 7 clases es evidencia de una característica muy deseable: la mayor parte del error está, entonces, concentrada en expresiones que son clasificadas como neutral. Naturalmente, lo ideal es reducir el error, pero también es importante que el error al menos sea de expresión clasificada como neutral en vez de clasificada como otra expresión. En tanto que nuestro sistema tiene funciones de castigo que penalizan este último caso, como consecuencia la mayor parte de expresiones dudosas son clasificadas como neutral en vez de asumir el riesgo de clasificarla como una expresión errada. Esto reduce la tasa global pero aumenta, en nuestra opinión, las prestaciones del sistema.

No obstante, hay otro inconveniente más drástico que impide la comparación directa entre resultados, incluso usando la misma base de datos, debido a la metodología de validación. Lamentablemente, en casi todos los trabajos del estado del arte la valida-

Figura 42. Comparación de resultados de clasificación de 6 y 7 clases



ción se hace por metodología *random n-folded*. Consideramos que esta metodología no es apropiada para un problema multiclase con múltiples muestras por individuo, por aprendizaje negativo y, en el caso frecuente del uso de múltiples muestras por individuo por expresión, por aprendizaje con muestras casi idénticas a las del conjunto de validación. Previamente ya mostramos cómo realizar una validación de esta naturaleza permite obtener resultados superiores al 99 % de clasificación, lo que fundamenta nuestra idea. En el capítulo 7 mostraremos pruebas adicionales de evaluación dinámica de expresión y de comparación entre el desempeño de los sistemas automáticos y la evaluación humana que permiten verificar que los valores muy elevados de clasificación de 7 clases son al menos controvertidos, especialmente con la base de datos no corregida CK.

De hecho, uno de los mayores problemas de intentar emular resultados es que la metodología de desarrollo o de validación puede ser involuntariamente equivocada, pero produciendo con ello resultados elevados, mas inverosímiles. Este escenario ocurre con altísima frecuencia en este campo, de modo que determinar cuáles resultados son confiables y cuáles pueden atribuirse a defectos metodológicos es un problema no trivial. Por ejemplo, entre los trabajos que superan nuestro desempeño o tienen resultados semejantes en 7 clases podemos encontrar algunos posibles problemas metodológicos. En [196] muestran resultados con LBP+KDI de 94.88 %. Sin embargo, en el mismo trabajo, usando la misma metodología, muestran resultado de 92.43 % con parámetros PCA. En tanto que en el estado del arte no se ha mostrado nunca ningún trabajo de desempeño aceptable con parámetros PCA, lo cual es razonable debido a la gran pérdida de información de textura, es factible que el resultado sea producto de un inconveniente metodológico. Así mismo, en el trabajo en [151], de 93.3 % con SVM-RBF, muestran resultados de clasificación de 7 clases usando los mismos parámetros pero con clasificación template matching, de 79.1 %. En el capítulo 3 mostramos cómo con

códigos POEM, que son notablemente inferiores descriptores de expresión que los códigos TPOEM, conseguimos, también con *template matching*, clasificación de 7 clases de 79.9 % (ver tabla 8), superior al 79.1 % reseñado[23]. En consecuencia, consideramos factible que el resultado de 93.3 % con SVM-rbf es impulsado por la metodología de validación. Así mismo, el resultado de 99.7 % en [85], aunque es para reconocimiento de 6 clases, es un resultado considerablemente elevado, aunque la metodología es aleatoria cruzada basada en muestras con 5 pliegues que, como ya hemos visto, puede producir resultados cercanos a clasificación perfecta. De hecho, en este mismo trabajo cuando la clasificación no se hace por un sistema SVM que es propenso a sobre ajuste en estas condiciones de validación, los resultados son de 93.1 % de clasificación de 6 clases, que es un valor más acorde con el resto del estado del arte y con la capacidad humana de clasificación de la expresión facial usando las mismas muestras.

6.5. Resultados usando distinta resolución y tamaño de los códigos TPOEM

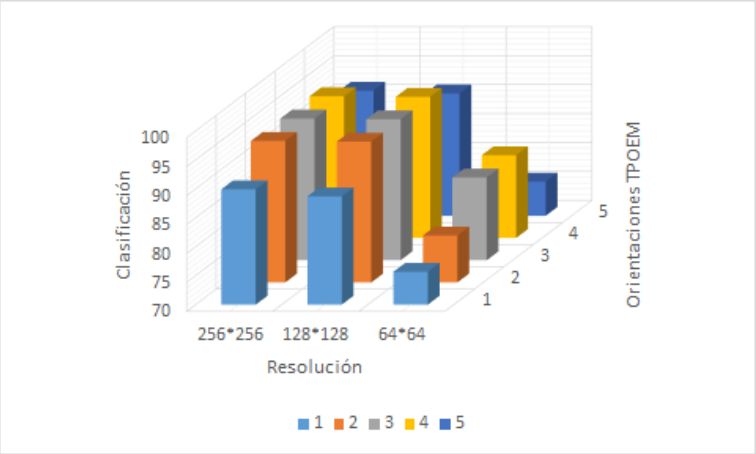
Hasta ahora los resultados mostrados han sido usando imágenes faciales de tamaño 128×128 pixeles y con 2 orientaciones espaciales y una orientación temporal TPOEM. En esta sección mostraremos los resultados obtenidos al hacer modificaciones de estos parámetros. Para ello, usamos imágenes de tamaño 64×64 y 256×256 pixeles (aunque en algunos casos los rostros detectados de la base de datos CK+ tienen tamaño ligeramente inferior a 256×256). Así mismo, hicimos entrenamiento y validación usando número variable de orientaciones TPOEM, desde 1, correspondiente a una orientación espacial, hasta 5, correspondiente a cuatro orientaciones espaciales y una temporal. Los resultados se observan en la figura 43.

Se puede determinar que aumentar la resolución a 256×256 pixeles representa una mejoría aunque no muy significativa, de 94.33 % a 94.51 %. El valor de orientaciones que tiene mejor tasa de clasificación es 4 orientaciones. A partir de 5 orientaciones los resultados son inferiores, debido a que la longitud de los códigos TPOEM es más grande, aproximadamente 4 veces más grande que usando 3 orientaciones, de manera que los histogramas no son muy representativos de las texturas locales.

El mejor resultado fue de 94.51 %, usando 4 orientaciones TPOEM e imagen de resolución 256×256. No obstante, en tanto que el uso de una imagen de 4 veces el tamaño y un código de aproximadamente 4 veces la longitud comparado con 3 orientaciones TPOEM y resolución 128×128 representa un costo de cálculo y memoria de casi 20 veces más grande, consideramos que la mejora marginal no es suficiente para optar por el uso general de estos parámetros. Sin embargo, esto muestra que los resultados obtenidos y reseñados en este trabajo no representan la máxima capacidad de tasa de

[23] Con códigos TPOEM no implementamos clasificación por *template matching*, pero teniendo en cuenta que los resultados con *template matching* ponderada con TPOEM fueron de 92.4 % (ver tabla 10), creemos que los resultados con *template matching* convencional hubiesen sido en todo caso muy superiores a 79.1 %.

Figura 43. Tasa de clasificación con la base de datos CK+ usando diversos parámetros TPOEM y resolución de imagen



clasificación de la expresión facial, sino un compromiso entre clasificación y costo de procesamiento.

6.6. Conclusiones

El diseño de un sistema de clasificación con un elevado número de parámetros para un problema multiclase fue un reto considerable. A diferencia de un problema de dos clases, en el cual el simple aporte levemente superior de la clasificación aleatoria por parte de cada clasificador débil es suficiente para construir un clasificador fuerte, en el problema de múltiples clases ésta no es la única consideración. En tanto que parámetros basados en regiones faciales, algunas expresiones tienen cercana similitud con otras, especialmente en algunas de las regiones del rostro. Esto hace que una aproximación convencional con arquitectura 1 vs. todos tienda a penalizar fuertemente aquellas expresiones que pierden puntos contra otras en zonas específicas. El caso más claro es la expresión de tristeza, que pierde numerosos concursos disputados con ira, disgusto y miedo en las regiones superiores del rostro, al tiempo que pierde puntos en la zona inferior del rostro contra la instancia neutral. Es decir, incluso si los parámetros tienen excelente correlación con las clases desde el punto de vista de clasificadores débiles (es decir, no alta correlación, pero suficiente para la construcción de clasificadores fuertes mediante ensamble), el problema multiclase representa retos considerables.

En este capítulo mostramos cómo las arquitecturas 1 vs. 1 probaron ser más exitosas que las arquitecturas 1 vs. todos. Naturalmente, esto tiene precio considerable en el entrenamiento y parcial en la validación, por cuanto 1 vs. todos es conformado por 6 clasificadores por vector TPOEM, contra 21 clasificadores por vector en arquitectura 1 vs. 1. Sin embargo, la importancia radica principalmente en la menor penalización a ciertas clases, sobre todo tristeza. Las pruebas iniciales se realizaron con clasificadores por discriminantes Fisher y posteriormente con máquinas de soporte vectorial con fun-

ciones de base radial. Finalmente, en tanto que algunos casos específicos probaron ser complicados para las SVM, especialmente en problemas con alto traslape entre clases, se reemplazaron las SVM problemáticas por FDA convencional.

En este trabajo introdujimos una idea novedosa que mejoró considerablemente las tasas de clasificación. Algunos metaclasificadores tuvieron desempeño altamente confiable cuando su salida fue una clase, pero clasifiación apenas superior al promedio aleatorio al clasificar la otra clase. Usar una aproximación ponderada convencional haría que el peso de la salida fuese estático o, usando estimación *prior*, dependiente del desempeño previo del clasificador. En nuestro caso usamos una aproximación Bayesiana para determinar la confiabilidad de cada metaclasificador dependiendo de su decisión, así que la ponderación no depende exclusivamente en el clasificador mismo, sino en su respuesta. Esto constituyó una mejora dramática en la clasificación, especialmente con datos traslapados y su uso podría ser más profundamente explorado para problemas multiclase con parámetros débiles.

Finalmente, diseñamos una arquitectura de clasificación basada en la idea general de deep learning. Para ello, diseñamos un banco de 7 subcapas paralelas consistentes en autoencoders apilados, uno por cada clase. La entrada a estos autoencoders es la salida de los clasificadores 1 vs. 1 correspondientes a cada expresión. Una segunda capa es conformada por otro autoencoder apilado, cuyo objetivo es interconectar las respuestas de cada abstracción de primer nivel para resolver el problema en una abstracción de más alto nivel. Finalmente, la clasificación fue realizada por una capa softmax de 7 salidas, correspondiente a las 6 expresiones faciales más neutral.

Los resultados sugieren que es posible obtener mejor tasa de clasificación usando una metodología basada en *deep learning*. Acotamos sugieren, por cuanto el número de iteraciones requerido con la metodología LOSO de validación es muy grande para obtener un número de validaciones equivalente al obtenido en nuestras pruebas anteriores, de manera que si bien los resultados numéricos son superiores, los intervalos de confianza estadística son más grandes. Naturalmente, para verificar con mayor certeza estadística la aparente mejoría se puede extender el proceso de validación, pero el costo completo de reducción de dimensiones, selección de parámetros y entrenamiento sucesivo de cada capa *deep learning* tarda más de 4 horas en nuestro sistema usado, así que evaluar todas las muestras CK+ requiere de más de 400 horas de evaluación para una arquitectura dada. De hecho, debido a esta limitación tampoco es posible determinar con certeza que la arquitectura usada, incluyendo el número de neuronas por capa, fuese la más apropiada, ya que hacer una validación completa por arquitectura implica costo de cálculo considerable, pero una validación incompleta proporciona menor confianza estadística. En todo caso, consideramos que los resultados muestran mejoría consistente, por lo tanto el resultado global de precisión tiene con certeza un rango de confianza mucho más reducido [24].

[24]Si los resultados por clase fuesen independientes, se podría estimar directamente el intervalo de confianza Bayesiano global con el número total de muestras evaluadas y número total de aciertos. Sin embargo, en este caso las tasas de acierto no son independientes. Ilustraremos esto con un ejemplo simple. Supongamos un sistema de 2 clases con 10.000 muestras; 5.000 de la clase I y 5.000 de la clase

Uno de los aspectos más importantes en el desarrollo de los sistemas de clasificación es la metodología de validación. Sin embargo, esta etapa es con frecuencia soslayada o pobremente implementada. En la mayor parte de los trabajos consultados en la bibliografía la validación se hace por metodología *random n-folded*. Nuestras primeras pruebas con clasificadores expertos basados en SVM mostraron resultados de clasificación superiores a 99 % usando esta metodología, en contraste con resultados notablemente inferiores usando metodología LSO con 10 repeticiones, usando el 10 % de los individuos en cada validación. Naturalmente, esta elevada discrepancia pese a que las dos técnicas usan 10 pliegues en cada iteración y validan con un número similar de muestras en conjuntos disjuntos no es razonable. En este capítulo mostramos cómo el uso de metodologías *random n-folded* constituye un sobreajuste de los clasificadores. Regularmente el sobreajuste es fácilmente detectado en la validación, pues produce pobre generalización. Pero en este tipo de problemas en el que hay muestras de varias clases para el mismo individuo, este sobreajuste no conduce a bajas tasas de validación, que generalmente son evidencia de sobreajuste en estos casos, sino, al contrario, elevan notablemente las tasas de clasificación. Más aún, numerosos trabajos de la bibliografía consultada no sólo usan esta metodología de validación, sino que sus muestras son conformadas por la totalidad de cuadros de video de las bases de datos CK o CK+. Esto agrava el inconveniente, por cuanto los clasificadores son validados con muestras que cuentan con numerosas muestras muy parecidas en el conjunto de entrenamiento. Es decir, no se viola el principio metodológico de no traslape de muestras entre entrenamiento y validación, pero en la práctica básicamente sucede esto, con muestras casi idénticas en dos conjuntos. Debido a esto nuestras pruebas, salvo para ilustrar este inconveniente, fueron realizadas con metodología LSO, donde cada *subject* es una persona de la base de datos.

Regularmente una metodología con n pliegues requiere únicamente de n validaciones para validar todas las muestras del conjunto de datos (salvo metodologías con pliegues no necesariamente disjuntos, tales como *bootstraping*). Sin embargo, hacer más iteraciones de validación ayuda a reducir el rango de confianza estadística. Esto sucede porque si bien incrementar el número de iteraciones implica que globalmente se repiten muestras en la validación, estas muestras son validadas por clasificadores entrenados usando muestras distintas cada vez. Es decir, hacer más de 10 validaciones para 10 pliegues mejora los rangos de confianza, lo cual es bastante deseable en este problema, debido

II. El sistema no es útil, clasificando todas las muestras como pertenecientes a la clase I. Es decir, el sistema tiene 100 % de clasificación de la clase A y 0 % de clasificación de la clase B. Los intervalos Bayesianos para la clase A son inferior 99.93 % y superior 100 %. Los intervalos para la clase B son inferior 0 % y superior 0.007 %. Si se usa la clasificación global para el cálculo del intervalo de confianza se obtiene inferior 49.02 % y superior 50.98 %, que es, naturalmente, igual que si el sistema hubiese tenido 50 % de clasificación de la clase A y 50 % de clasificación de la clase B, lo cual es estadísticamente poco razonable por cuanto en este caso los intervalos de confianza A y B son considerablemente más grandes que en el ejemplo inicial. Lamentablemente los expertos consultados coincidieron en que el cálculo de intervalos de confianza estadística de pruebas dependientes es un proceso altamente complejo e impreciso, de modo que sólo nos resta concluir que sin duda debe de ser inferior que el intervalo de confianza individual de cada clase, sin estimación precisa del valor real.

a que el número limitado de muestras por clase, particularmente miedo y tristeza, ocasiona que 10 iteraciones produzcan rangos de confianza considerables. En consecuencia, casi todos nuestros protocolos de validación fueron realizados con 50 a 100 iteraciones de LSO, de manera que se redujo a márgenes notables el rango de confianza en tanto que se validó con un mayor número de muestras. Lamentablemente, esto era de imposible realización para la clasificación compleja basada en *deep learning*, de manera que los intervalos de confianza en este caso no son tan pequeños, pero en general sugieren mejoría de clasificación.

Nuestro trabajo produjo aportes importantes en el problema de metaclasificación con parámetros débiles para la construcción de clasificación fuerte. Clasificadores simples con distancia chi-cuadrado o distancia Mahalanobis o incluso clasificadores más fuertes con FDA y SVM-RBF mostraron tasas de clasificación de 7 clases bastante reducidas, en numerosos casos apenas ligeramente sobre 16 % por parámetro, lo cual es sólo muy levemente superior al resultado de un clasificador aleatorio, de 14.14 %. Sin embargo, la arquitectura diseñada incluyendo parámetros ponderados, información mutua entre parámetros y APCC mostraron que se puede obtener tasas cercanas y superiores al 95 % de clasificación global incluso con estas limitaciones.

Consideramos que las capacidades del sistema global como clasificador preciso de la expresión facial pese a la pobre capacidad de cada metaclasificador individual es un importante paso hacia el entendimiendo de problemas de alta dimensión, baja discriminación de clasificación. Una técnica común es intentar mejorar el poder de clasificación de cada parámetro individual, pero nuestras pruebas mostraron que esto puede conducir a sobre ajuste. Esto acontece porque el ajuste fino de cada metaclasificador individual para mejorar su descripción del conjunto de entrenamiento usando AdaBoost condujo a menor generalización del problema. Naturalmente, haciendo esta misma metodología pero con validación *random n-folded* puede producir resultados excelentes con *boosting*, debido a que los metaclasificadores se están ajustando a datos casi idénticos a muestras usadas en la validación, pero con un protocolo más global como LSO este ajuste fino por metaclasificador no produjo resultados aceptables. En cambio nuestra aproximación global con metaclasificación simple sin ajustes a los parámetros individuales produjo resultados notables que permitieron generalización en las bases de datos CK y CK+ y, como se verá en el capítulo 7, también generalización incluyendo otras bases de datos.

7. Análisis dinámico de la expresión facial y pruebas adicionales

7.1. Introducción

En este capítulo mostraremos algunas pruebas adicionales que se realizaron en este trabajo con el fin de determinar la capacidad de generalización de la codificación TPOEM con clasificación dinámica Bayesiana, comparación de resultados entre clasificación humana y clasificación automática y pruebas dinámicas con la base de datos CK para determinar fuentes de error.

En la sección 7.2 se mostrará cómo con una aproximación *naïve* Bayesiana se consiguió hacer una evaluación dinámica de la expresión facial en secuencias de video. Para ello se usó una ponderación por puntaje entre 0 y 1 en vez de la clasificación *winner takes all*, y así establecer zonas de transición entre expresión y expresión o neutral y expresión. Con esto se entrenó un sistema de clasificación con la base de datos CK+, usando metodología LSO. La introducción de ponderación Bayesiana permitió obtener puntuación más fluida entre cuadro y cuadro y mejorar las tasas de clasificación en las secuencias de video.

En la sección 7.3 se muestra el número de individuos y el tipo de problema de clasificación en algunas de las bases de datos más usadas en la bibliografía. Con ello se ilustra la dificultad adicional del problema de reconocimieno de la expresión facial debido a la disponibilidad de datos, especialmente en comparación con bases de datos de identificación facial. Adicionalmente se describe el protocolo de entrenamiento y clasificación usando la base de datos CK+ para entrenar el sistema de clasificación y la base de datos KDEF para la validación, lo que constituye una prueba de la capacidad de generalización de la metodología usada.

En la sección 7.4 se siguió la sugerencia dada en [98] por Lucey, Kohn, Kanade et al. de usar la base de datos KDEF para determinar con precisión la comparación directa de capacidad de clasificación entre humanos y máquina. Esta comparación hasta nuestro conocimiento aún no ha sido realizada, pese a la necesidad de establecer un marco de referencia de evaluación automática de la expresión facial en relación a la clasificación realizada por humanos no expertos. Adicionalmente se hizo un estudio con voluntarios para clasificación humana de la base de datos CK+, con el fin de comparar estos resultados con los obtenidos por nuestro sistema automático. Con estos procedimientos se espera determinar la fiabilidad de los sistemas automáticos en comparación con los humanos en la valoración de la expresión facial, así como la capacidad de generalización de clasificación con una base de datos usando un sistema de clasificación entrenado con otra base de datos. Así mismo, pruebas adicionales que se realizaron permiten determinar el efecto del uso de una base de datos no estandarizada (KDEF) con numerosas muestras inapropiadas, en los resultados de clasificación o entrenamiento.

A continuación, en la sección 7.5, realizamos evaluación dinámica con la base de datos CK, con el fin de entender por qué si bien en la mayoría de trabajos de validación de sistemas de clasificación con la base de datos CK los resultados fluctúan entre 80 % y

90 %, hay algunos trabajos con tasas de clasificación muy superiores, incluso cercanas a clasificación perfecta. Para ello realizamos análisis de muestras erróneamente clasificadas comparado con muestras con clasificación incorrecta y así determinar cuáles son los factores que pueden conducir a error. En nuestra opinión, los sistemas de error ínfimo de clasificación pueden estar incurriendo en error metodológico de prueba de hipótesis o en eliminación manual de muestras de la base de datos (que ciertamente puede ser otro error metodológico, salvo que el proceso sea ciego e independiente), de manera que una comparación directa numérica entre porcentajes de clasificación no es apropiada, debido a la situación de desventaja de sistemas que han sido entrenados y validados sin eliminación de muestras o usando metodología de validación *leave-subjects-out*.

Los resultados de las pruebas realizadas en este capítulo están mostrados en la sección 7.6, así como el análisis detallado de estos resultados.

En la sección 7.7 hacemos un análisis de la complejidad de costo computacional de los algoritmos usados. Para ello separamos el sistema completo en cada una de las etapas del protocolo de validación, calculando la complejidad de costo de cada una. Se muestra cómo el algoritmo completo es factible y la complejidad de costo total es viable para el uso en tiempo real.

Por último, en la sección 7.8 mostramos las conclusiones del trabajo realizado en este capítulo, así como consideraciones generales y sugerencias de extensión de este trabajo.

Los aportes de este capítulo al estado del arte incluyen la implementación de clasificación dinámica de la expresión facial usando modelo *naïve* Bayesiano, el diseño de una aplicación para la prueba de las bases de datos CK+ y KDEF por voluntarios humanos, desarrollo de evaluación humana de la base de datos CK+, comparación entre el desempeño de la evaluación automática y la evaluación humana y pruebas que, hasta nuestro conocimiento no han sido realizadas en otros trabajos, permiten probar la capacidad de generalización de los sistemas automáticos de expresión facial usados en la validación con imágenes y secuencias de video de naturaleza desconocida para el sistema. Así mismo, nuestras pruebas dinámicas con la base de datos CK permiten añadir elementos que fundamentan nuestra hipótesis de que resultados muy superiores de clasificación con esta base de datos, frecuentemente publicados, pueden ser producto de un error metodológico de prueba de hipótesis o selección/descarte manual de muestras de la base de datos previo al entrenamiento y validación.

7.2. Caracterización dinámica de la expresión facial

Las expresiones faciales son procesos dinámicos con una transición entre una instancia neutral y una expresión ápice o entre dos expresiones. En consecuencia, hay niveles intermedios de expresión facial correspondientes a estados de transición que en muchos casos, dependiendo de su intensidad, pueden ser de difícil resolución. Uno de los principales inconvenientes de los sistemas de reconocimiento de expresión facial convencionales es que comúnmente están entrenados y probados con expresiones ápice,

despreciando estados de transición. Las razones más comunes para ignorar la evaluación de estos estados de transición son:

- Falta de datos de comparación: La mayor parte de los datos usados para la evaluación de sistemas de clasificación de la expresión facial son obtenidos de bases de datos estándar. El uso de bases de datos estandarizadas facilita la comparación entre diversas técnicas, por cuanto el uso de bases de datos originales no permite comparar directamente resultados obtenidos con otras metodologías probadas con otras bases de datos. Sin embargo, las bases de datos usan imágenes individuales o secuencias dinámicas bajo la presunción de que en algún punto durante la secuencia el individuo realiza una expresión facial particular. Teniendo en cuenta que la expresión facial es una transición continua, ha una zona gris en la cual es imposible determinar si el individuo ejecuta una expresión facial o la pose es aún neutral. Como tal, la comparación de resultados es compleja.

- Pérdida de precisión: La precisión de sistemas de reconocimiento de expresión facial que incluyan expresiones transicionales o expresiones suaves es claramente inferior que usando únicamente expresiones ápice. Si bien esta conclusión es obvia, no es deseable obtener resultados cuya comparación inmediata con otros trabajos muestre tasas de clasificación inferiores, incluso no siendo la comparación justa debido al distinto objetivo y mayor dificultad de la clasificación, especialmente en un campo en el cual tasas de reconocimiento superiores al 90 % son comúnmente publicadas. Consecuentemente, una apuesta más segura es enfocarse en expresiones faciales fuertes cuya tasa de reconocimiento es más elevada.

Sin embargo, en este trabajo decidimos incluir la evaluación de expresiones faciales en secuencias de video incluyendo cuadros de transición. Como tal, el objetivo fue medir y mejorar el desempeño del sistema al lidiar con cuadros de transición o expresiones faciales débiles. El procedimiento general fue incluir imágenes y cuadros de video de las bases de datos usadas para entrenar los clasificadores. Todas las imágenes usadas corresponden a expresiones ápice o a instancia neutral. Es decir, los subconjuntos de entrenamiento no incluyeron imágenes de transición. El fundamento de esta decisión es que no es posible definir el punto exacto de transición entre una expresión neutral y una expresión ápice sin arbitrariedad, de modo que asignar puntajes manualmente puede producir errores de clasificación. Adicionalmente, el objetivo es observar el desempeño del sistema en situaciones desconocidas, de manera que descartar muestras de transición del entrenamiento permite evaluar el comportamiento de clasificación en estos casos. Por otra parte, esto significa que los clasificadores no tienen ninguna información *a priori* acerca de esta clase de estados de transición, así que el comportamiento puede ser impredecible. No obstante, si la transición entre una instancia neutral y una expresión ápice es suave, se expera que los puntajes de expresión reflejen esta transición todo el tiempo (es decir, una expresión de ligera alegría puede tener puntaje neutral 0.8 y puntaje de alegría de 0.2, por ejemplo). Por otra parte, nuestras pruebas mostraron que esto no necesariamente es cierto. Algunos estados de transición se acercan más a

una expresión distinta que la expresión ápice de la secuencia evaluada, debido a que la actividad muscular no es simultánea, la deformación facial no ocurre con igual rapidez y la transformación entre neutral y expresión no puede ser modelada con una simple transformación geométrica debido a oclusiones (por ejemplo, es imposible modelar la transición entre neutral y sorpresa debido a la abertura de la boca, cuya información no está contenida en los cuadros con la boca cerrada). Debido a ello encontramos que en transiciones neutral a alegría hay típicamente puntaje elevado de miedo, por cuanto en estas transiciones la boca aún no forma el arco convencional de la alegría, mientras que la forma general sugiere expresión de miedo. Posteriormente mostraremos algunos casos específicos.

7.2.1. Evaluación dinámica de la expresión facial usando parámetros TPOEM

En el capítulo 6 mostramos los resultados de la evaluación de la expresión facial usando subconjuntos de muestras de instancias neutrales y expresiones ápice. En esta sección nos enfocaremos en la evaluación de secuencias completas de video. Como tal, el objetivo es probar si los clasificadores entrenados con muestras ápice son suficientemente buenos para caracterizar con precisión la transición entre una instancia neutral y una expresión ápice.

Con el fin de evaluar la transición dinámica, una metodología de clasificación *winner takes all* no es apropiada, por cuanto el resultado es una salida definitiva sin la posibilidad de una zona gris correspondiente a los cuadros de transición. En consecuencia, los algoritmos de clasificación fueron modificados de manera que se use la salida de cada clasificador individual por expresión para obtener un puntaje normalizado usando la ecuación 7.1.

$$Sn_{i,c} = \frac{S_{i,c}}{\sum_{cl} S_{i,cl}}, \quad c = 1, ..., 7 \tag{7.1}$$

donde $S_{i,c}$ es el puntaje de cada clase con el clasificador basado en *deep learning* [1].

El objetivo es probar si esta aproximación conduce a resultados de puntajes altos de instancia neutral en los cuadros iniciales de las secuencias de video, cuando la expresión aún no está formada, y posteriormente los puntajes de la expresión particular se incrementan en tanto que la expresión facial se acerca a la expresión ápice.

Un ligero inconveniente mostrado por las pruebas preliminares es que la expresión facial es generalmente bien descrita por el puntaje por muestra, pero la metodología previa hace que difícilmente una muestra, incluso ápice, tenga puntaje cercano a 1

[1] En nuestras pruebas iniciales de validación dinámica de la expresión facial aún no habíamos considerado usar el clasificador basado en *deep learning*. Es decir, la mayor parte del trabajo referente a este capítulo se hizo inicialmente con la fusión directa de metaclasificadores SVM y FDA con información mutua Bayesiana descrita en el capítulo 6. Debido a esto, los puntajes por expresión no fueron definidos en estos casos como en la ecuación 7.1, sino que se obtuvo un puntaje normalizado por cada celda TPOEM en vez de *winner takes all* y el resultado final fue un puntaje entre 0 y 1 por expresión. Posteriormente se hizo el diseño de los clasificadores basados en *deep learning* y las pruebas de desempeño dinámico de este capítulo fueron realizadas nuevamente con estos clasificadores.

para su expresión correspondiente. Esto ocurre debido a que el resto de expresiones y la instancia neutral tienen puntajes pequeños, pero no despreciables, que le restan puntuación a la expresión dominante. Debido a ello se hizo una ligera modificación con un puntaje umbral determinado manualmente. Si el puntaje de una expresión es menor que este umbral, se modifica a puntaje cero, y así no le resta puntaje a la expresión dominante.

7.2.2. Clasificación *naïve* Bayesiana

Nuestras pruebas usando la metodología descrita en la subsección anterior mostraron que la codificación TPOEM dinámica fue útil para describir las variaciones dinámicas de las expresiones faciales, incluyendo estados de transición y expresiones suaves. Sin embargo, pese a estos buenos resultados consideramos que es posible hacer algunos ajustes que permitan producir mejores resultados. El puntaje de expresión obtenido usando la metodología previa tiene cierta independencia entre cuadro y cuadro [2], pero, especialmente en zonas de transición, encontramos cierta variación fuerte entre los puntajes de cuadros vecinos, de modo que los puntajes fueron suavizados usando una función Gaussiana tal como se muestra en la ecuación

$$Sn_{i,c,m} * \frac{1}{\sqrt{2\pi\sigma}} e^{-m^2 2\sigma^2} \rightarrow Sn_{i,c,m} \tag{7.2}$$

donde $Sn_{i,c,m}$ es el puntaje de la muestra i, expresión c, cuadro m.

Posteriormente se construyó un sistema de clasificación *naïve* Gaussiano. La teoría Bayesiana *naïve* se basa en la presunción de que los datos continuos tienen distribución Gaussiana, lo cual no es el caso debido a la naturaleza de los datos y al truncamiento de los puntajes señalado previamente. Sin embargo, nuestras pruebas mostraron un buen desempeño de esta metodología incluso con los datos no Gaussianos. Debido al tamaño grande del posible conjunto de datos de salida, hicimos discretización de los vectores correspondientes a los puntajes por expresión. Es decir, en vez de usar valores continuos de puntaje por expresión, se usaron rangos de 0.2, con el fin de que el conjunto de datos de entrenamiento fuese suficiente para caracterizar una buena parte de las posibilidades de salida. Esta discretización es dada por $Sn_{i,c,m} \rightarrow fn_{i,c,m}$ y los vectores característicos por expresión están denominados como $F(c)$.

Con el fin de construir el clasificador, la siguiente etapa fue obtener las probabilidades condicionales de ocurrencia de eventos. Los parámetros usados fueron los puntajes discretizados por clase por cuadro y cuadros adyacentes. Debido a la misma razón de limitación de número de datos, decididimos usar únicamente cuatro cuadros vecinos, dos precedentes y dos futuros. Posiblemente mejores resultados son realizables con un mayor número de vecinos incluidos con el fin de hacer una descripción más fina de

[2] En realidad, no hay independencia completa entre los puntajes entre cuadro y cuadro debido a la naturaleza de la codificación TPOEM, que incluye parámetros obtenidos usando características de cuadros de video cercanos.

las variaciones dinámicas, pero el conjunto de datos no permite aumentar los estados Bayesianos considerablemente conservando adecuada capacidad de descripción. La clasificación es realizada usando la ecuación 7.3.

$$y = \arg\max_{c} P(y) \prod_{T} w_t P(F(c)_t \approx f_{i,t} \mid C = c_i) \tag{7.3}$$

donde $f_{i,t}$ es el conjunto de parámetros de la muestra i, vecindad t y $F(c)_t$ es el conjunto de parámetros característicos por clase c en la vecindad t. Para obtener la similitud entre los vectores $F(c)_t$ y $f_{i,t}$ se usa distancia Euclidiana entre los vectores con umbralización definida previamente con los datos de entrenamiento. Para ello, se determinó que un vector es similar al vector de comparación si su distancia Euclidiana respecto del vector de comparación tiene magnitud igual o menor que 3 desviaciones estándar del conjunto de datos de parámetros por clase. Con ello se garantiza que casi todas las muestras del conjunto de entrenamiento son clasificadas correctamente por el sistema y, en consecuencia, se espera que las muestras de validación sean también clasificadas correctamente.

Inicialmente los valores de los pesos w_t fueron estimados manualmente y el entrenamiento y la validación fueron realizados con estos pesos estimados, con valor de 1 para el cuadro central evaluado, 0.66 para los cuadros vecinos y 0.33 para los cuadros más alejados. Estos cuadros no son cuadros inmediatamente adyacentes en la secuencia de video, debido a la pequeña variación de las imágenes entre cuadros sucesivos en secuencias capturadas a 30 fps, sino que corresponden a diezmado temporal de modo que los cuadros están en t_o (valor actual), $t_o \pm 5$ y $t_o \pm 10$.

7.3. Reconocimiento de expresión facial en secuencias de video no estandarizadas

Uno de los problemas más importantes encontrados al hacer este trabajo es la reducida disponibilidad de bases de datos de expresión facial debidamente validadas y estandarizadas. En oposición a identificación facial, donde el problema está limitado a emparejar una imagen de un individuo con una base de datos definida, de manera que el protocolo de estandarización es simple, en el problema de reconocimiento de la expresión facial las expresiones deben ser validadas. El principal inconveniente es que la validación de la expresión facial no es un proceso simple y requiere de varios años de entrenamiento, de modo que la recopilación de una base de datos no es una tarea trivial e incluso la base de datos preliminar Cohn-Kanade tuvo considerable trabajo adicional de verificación y eliminación de muestras defectuosas para la siguiente versión. En la versión CK+ muchas muestras pertenecientes a la base de datos original fueron removidas porque no representaban con precisión la expresión etiquetada. Las muestras deben ser obtenidas y luego validadas por un panel experto, incluyendo la evaluación de las unidades de acción (AU), con el fin de garantizar que las expresiones faciales en la base de datos son muestras confiables para la prueba y validación. Debido

a esto no hay muchas bases de datos disponibles y el tamaño de estas bases de datos es en general reducido. En la tabla 36 mostramos algunas bases de datos típicamente usadas en investigación de identificación facial y expresión facial.

Tabla 36. Número de individuos en bases de datos típicamente usadas para reconocimiento facial y reconocimiento de la expresión facial

Base de datos	Número de individuos	Identificación facial	Expresión facial
Cohn-Kanade	97		x
CK+	123		x
MMI	43-75		x
JAFFE	10		x
Belfast	125		x
KDEF	70		x
FERET	1199	x	
M2VTS	295	x	
Multi-PIE	130	x	
Yale Face	15	x	x

En la tabla 36 se puede observar que la disponibilidad global de las bases de datos es limitada para expresión facial. La limitación no es únicamente numérica, sino también de alcance y viabilidad de las bases de datos. Por ejemplo, la base de datos JAFFE tiene muestras únicamente de mujeres japonesas y las bases de datos CK y CK+ tiene individuos jóvenes (estudiantes principalmente), de modo que su extrapolación universal no es directa. La base de datos Belfast corresponde a secuencias de video naturales y posteriormente categorizadas manualmente y la base de datos KDEF no es validada (es decir, todas las muestras de expresión por individuo fueron seleccionadas, sin descartar muestras que no representan adecuadamente la expresión etiquetada); de modo que este tipo de base de datos posiblemente no tenga la validación suficiente para desarrollar una descripción adecuada. En contraste, debido a las razones explicadas previamente, la disponibilidad de imágenes para identificación y reconocimiento facial es mucho más elevada en general.

Por otra parte, el uso de imágenes/secuencias de video compiladas por los autores no es un protocolo ideal en el problema de reconocimiento de expresión facial. Esto es porque, tal como se señaló previamente, el entrenamiento necesario para realizar una validación exitosa es extenso. Además, el uso de datos propios de los autores dificulta o incluso impide la comparación directa de resultados comparados con los resultados usando otras técnicas.

No obstante, en tanto que en el capítulo precedente ya mostramos con extensión la validez de los descriptores TPOEM y nuestras técnicas de entrenamiento y clasificación para el problema de reconocimiento de expresión con las bases de datos estándar Cohn-Kanade y CK+, con resultados notables, en esta sección incluimos el trabajo de reconocimiento de la expresión facial generalizada usando parámetros TPOEM de la ba-

Figura 44. Entrenamiento del sistema de clasificación usado para la prueba cruzada con la base de datos KDEF

se de datos CK+, validados con las muestras de la base de datos KDEF. La metodología es descrita en la figura 44.

En la figura 44 se observa que los clasificadores son entrenados únicamente con muestras provenientes de la base de datos CK+. Es decir, a diferencia de los protocolos previos en los cuales muestras de la misma base de datos eran usadas para entrenar (no obstante no usar muestras del mismo individuo en los dos conjuntos), en este caso las muestras del conjunto de validación son completamente desconocidas para los clasificadores en resolución, calidad y condiciones de iluminación, por ejemplo. Es decir, con esto el pequeño nivel de sobre ajuste que puede existir al entrenar y validar con imágenes cuya técnica de captura fue muy similar es puesto a prueba. Los resultados obtenidos no fueron consignados en el capítulo 6 debido a la imposibilidad de contrastar estos resultados con trabajos del estado del arte. No obstante, en la sección 7.6 mostraremos algunas de las pruebas realizadas con esta metodología, así como la visualización de la evolución dinámica de los puntajes de expresión facial en algunas de las secuencias obtenidas y evaluadas.

7.4. Pruebas de generalización con la base de datos KDEF

En el capítulo 6 todas las pruebas realizadas fueron usando las bases de datos CK y CK+ y validando con imágenes/secuencias de las mismas bases de datos. En [98], Lucey, Cohn, Kanade et al. sugieren la validación de la base de datos CK+ por evaluadores humanos y la validación de la base de datos KDEF por un sistema de clasificación automática:

«This (classification result) suggests that an automated system can do just as a good job, if not better as a naive human observer and suffer from the same confusions due to the perceived ambiguity between subtle emotions. However, human observer ratings need to be performed on the CK+ database and automated results need to be conducted on the KDEF database to test out the validity of these claims.» (Lucey et

al., 2010)

En consecuencia, y teniendo en cuenta que, hasta nuestro conocimiento, aún no se han realizado estas pruebas sugeridas, al menos en publicaciones relevantes de los últimos años, en este trabajo decidimos incluir estas validaciones. De esta forma esperamos probar la capacidad de generalización de los clasificadores y la comparación entre clasificación realizada por humanos y máquina.

Para ello, usamos la base de datos KDEF, que es una base de datos relativamente extensa de expresión facial. Esta base de datos consiste de 70 individuos, con imágenes por individuo para las 6 expresiones faciales definidas por Ekman y la instancia neutral. La principal ventaja de usar esta base de datos es que fue extensamente probada con 272 evaluadores humanos, de manera que permite una comparación directa de resultados de desempeño entre la clasificación automática y la clasificación humana. Por otra parte, un inconveniente importante de esta base de datos es que no está estandarizada. Es decir, todas las muestras por individuo por expresión son usadas en la base de datos, a diferencia de la base de datos CK+ en la cual hay un promedio menor de 3 muestras por individuo. En tanto que en la metodología de adquisición de la base de datos CK+ se solicitó a cada individuo que realizara cada una de las 6 expresiones faciales, esto implica que en el proceso de estandarización de la base de datos más de la mitad de las muestras fueron eliminadas por el panel de expertos. Teniendo en cuenta esta proporción, es probable que en la base de datos KDEF más de la mitad de las muestras no sean representativas de la expresión facial etiquetada, con el agravante de que la distribución no es homogénea, sino que algunas expresiones tienen muchas más muestras inválidas que apropiadas, lo cual, seguramente, produce problemas de validación. Sin embargo, en tanto que es la única base de datos de expresión facial con una adecuada validación con un numeroso conjunto de evaluadores humanos, se usó esta base de datos para la comparación.

La metodología del trabajo consistió inicialmente en el uso de todas las muestras de la base de datos CK+ para el entrenamiento. Debido a que ninguna de las muestras es usada en la validación, este protocolo no produce sobre ajuste. De hecho, este protocolo minimiza por completo los posibles errores de validación porque usa dos bases de datos disjuntas de naturaleza muy distinta de calidad, iluminación y tamaño, de manera que los resultados no incurren en ninguna metodología incorrecta.

Una vez el clasificador fue obtenido, se realizó la validación con las imágenes de la base de datos KDEF. Para ello no se requiere de un protocolo con *folds*, por cuanto este procedimiento sólo es necesario cuando un conjunto limitado es usado para validación con las mismas muestras iniciales, para no repetir muestras en entrenamiento y validación, pero en este caso la validación es con muestras de otra base de datos. Esto representa una ventaja adicional, por cuanto en nuestras metodologías previas con LSO se requería hacer múltiples entrenamientos y validaciones para probar todas las muestras, mientras que en este caso un solo entrenamiento es suficiente, reduciendo así el costo de cálculo, que es bastante deseable particularmente teniendo en cuenta el extenso costo de cálculo del entrenamiento de los clasificadores basados en deep learning.

Como prueba adicional, se realizó entrenamiento y validación de la base de da-

tos KDEF usando muestras de la propia base de datos, de manera similar a nuestra metodología anterior por LSO. El objetivo de esta metodología es determinar la generalización de los resultados de validación usando entrenamiento con CK+ y validación con KDEF. Esto es, si, por ejemplo, se obtienen resultados mediocres con entrenamiento CK+ pero resultados adecuados con entrenamiento y validación KDEF, esto puede mostrar que incluso el uso de metodología LSO no impide por completo cierta sobre especialización, en el sentido de que los clasificadores aprenden principalmente a clasificar imágenes/secuencias de ciertas características, pero deterioran sus resultados con datos con otro tipo de características [3]

A continuación se realizaron las pruebas de validación humana de la base de datos CK+. Para ello se usó un conjunto compuesto principalmente por voluntarios humanos de la Universidad Federal de Rio Grande del Sur, más voluntarios humanos externos. Los procesos de validación de la expresión facial por humanos tienen algunos problemas inherentes de metodología que no pueden ser eliminados, sino tan sólo atenuados. Por ejemplo, la naturaleza de elección entre 6 expresiones faciales y 1 instancia neutral hace que exista un prejucio en los evaluadores, de modo que la clasificación de la instancia neutral es fuertemente perjudicada [4]. Adicionalmente, nuestros resultados mostraron que la capacidad de clasificación de la expresión facial no es homogénea entre los individuos. Algunos individuos obtienen tasas de clasificación considerablemente buenas, mientras que otros tienen gran error de clasificación en expresiones como ira y disgusto (equívocos entre las dos) y clasificación deficiente de miedo, que reducen las tasas globales de clasificación. Sin embargo, los resultados obtenidos son coherentes con los resultados de clasificación humana de la base de datos KDEF y con tendencia de errores similar a la obtenida por la evaluación automática, de modo que son satisfactorios para realizar la comparación entre humano y máquina.

Por último, se realizó validación de la base de datos CK+ usando la base de datos KDEF como entrenamiento. Tal como reseñamos previamente, consideramos que la base de datos KDEF tiene un numeroso conjunto de muestras que no son representativas de la expresión facial etiquetada. Al validar la base de datos KDEF con entrenamiento CK+, muchas de estas muestras son clasificadas "erróneamente". Sin embargo, al validar la base de datos CK+ entrenando con la base de datos KDEF se espera que la influencia de las muestras deficientes sea mucho más grande. Esto es, un clasificador entrenado con un conjunto de muestras en el cual muchas de las muestras no son representativas de la clase etiquetada forzosamente tiene problemas de generalización. En tanto que la proporción

[3] Cabe señalar que incluso si se da el caso de que la clasificación KDEF-KDEF es muy superior a la clasificación CK+ -KDEF, esto no implica que los resultados sean incorrectos/inapropiados. Al contrario, al consultar con expertos en el tema, incluyendo asesores externos de este trabajo, el consenso es que la validación con una base de datos distinta a la usada para entrenar probablemente tendría resultados muy inferiores, no por sobre ajuste, sino por limitaciones propias e inevitables del proceso de entrenamiento con datos limitados a un solo tipo de imágenes/secuencias.

[4] Esto ocurre porque en una situación natural, por defecto asumimos que las personas tienen una expresión neutral. Sin embargo, en una prueba de clasificación ocurre lo contrario y los evaluadores tienden a atribuir expresión a la mayoría de muestras, incluso si son en realidad representativas de instancia neutral.

de las clases en la base de datos CK+ muestra que las expresiones tristeza, miedo e ira son las menos representadas, es razonable considerar que estas expresiones son las que probablemente tienen más muestras insatisfactorias en la base de datos KDEF (debido a que ninguna muestra fue eliminada). Si es así el caso, esperamos encontrar resultados muy inferiores de clasificación de estas 3 expresiones en la validación de la base de datos CK+ entrenada con la base de datos KDEF comparado con los resultados de validación CK+ - CK+.

En la sección 7.6 mostraremos los resultados y análisis de resultados de estas pruebas.

7.5. Análisis dinámico de generalización con la base de datos CK

En el capítulo 6 mostramos los resultados de validación usando inicialmente la base de datos CK y posteriormente usando las dos bases de datos CK y su versión extendida y modificada CK+. Buena parte de las modificaciones que se realizaron a la base de datos original se refieren a eliminación de muestras con etiqueta incorrecta o muestras no representativas de la expresión etiquetada. Razonablemente, se espera que un buen clasificador con capacidad adecuada de generalización debería clasificar estas muestras de baja calidad o de etiqueta incorrecta de manera "errónea" [5]. En cuanto realizamos las primeras pruebas con la versión extendida y modificada en la base de datos CK+, observamos alta discrepancia de resultados. Debido a ello, retomamos los resultados de CK, para determinar en qué imágenes/secuencias el clasificador estaba teniendo resultados erróneos. Los resultados, de manera no sorprendente, mostraron que buena parte del error estaba en muestras que posteriormente habrían de ser eliminadas en la versión extendida CK+.

Debido a este análisis preliminar, hicimos nuevamente validación de toda la base de datos CK, incluyendo no sólo las muestras ápice por individuo, sino toda la secuencia de video, para observar la evolución dinámica de los puntajes por expresión. Una vez obtenidos los puntajes por expresión por secuencia, fueron normalizados y visualizados en una gráfica puntaje vs. tiempo. Posteriormente fueron seleccionadas muestras de clasificación correcta y clasificación incorrecta correspondientes a secuencias publicables (sólo imágenes de 11 de los 97 individuos tienen autorización de publicación), con el fin de poder mostrar los resultados de puntaje vs. tiempo incluyendo la visualización de algunas de las imágenes de cada secuencia en el eje temporal, y así permitir la valoración por inspección humana. En la sección 7.6 se mostrarán los resultados de este procedimiento, así como una discusión de los mismos.

[5] En realidad no es clasificación errónea, por supuesto. Errónea desde el punto de vista de que no corresponde con la etiqueta determinada, pero si esta etiqueta no es correcta o la muestra no es representativa, la clasificación "correcta" es, en realidad, equivocada. Esto, incidentalmente, soporta nuestra afirmación de que muchos de los trabajos con tasas de reconocimiento superiores a 99 % con la base de datos CK deben tener un error metodológico de validación, por cuanto no es razonable que incluso muestras cuya etiqueta no es correcta sean clasificadas acertadamente.

7.6. Resultados

7.6.1. Resultados de comparación de desempeño entre reconocimiento automático y panel de humanos

La primera prueba realizada fue la validación de la base de datos KDEF con un sistema entrenado usando las muestras de la base de datos CK+. Los resultados se muestran en la tabla 37.

Tabla 37. Validación de la base de datos KDEF con sistema entrenado con la base de datos CK+

	Ira	Dis.	Mie.	Ale.	Tri.	Sor.	Neu.
Ira	**59.2**	26.7	5.8	2.5	5.8	0.0	0.0
Disgusto	0.0	**79.2**	15.8	3.3	1.7	0.0	0.0
Miedo	6.7	0.0	**50.0**	5.0	15.0	14.2	9.2
Alegría	0.0	0.0	0.0	**97.5**	0.0	0.0	2.5
Tristeza	3.3	4.2	37.5	0.0	**50.0**	0.0	5.0
Sorpresa	0.0	0.0	1.7	3.3	0.0	**81.7**	13.3
Neutral	2.5	0.0	0.0	0.0	0.0	0.0	**97.5**

Los resultados son muy inferiores a los obtenidos haciendo entrenamiento y validación con la base de datos CK+. No obstante, tal como se señaló previamente, la base de datos KDEF es una base de datos no validada, en la cual los resultados de clasificación por el panel de evaluadores humanos está limitado por los problemas de no estandarización. Para mostrar esto, en la tabla 38 transcribimos los resultados de validación de la base de datos KDEF realizada por un panel de individuos.

Tabla 38. Validación de la base de datos KDEF por un panel de evaluadores humanos

Ira	**Dis.**	**Mie.**	**Ale.**	**Tri.**	**Sor.**	**Neu.**
78.8	72.2	43.0	92.6	76.7	77.1	62.6

Al observar las dos tablas se puede determinar que la clasificación automática tuvo un desempeño ligeramente superior a la clasificación realizada por el panel de humanos (73.6 % vs. 71.9 %). Lamentablemente, los resultados publicados de evaluación humana no incluyen matriz de confusión, para establecer en qué situaciones los humanos tienen equívocos de clasificación. Sin embargo, se pueden determinar algunas observaciones importantes. En primer lugar, la baja tasa de clasificación de los humanos y la máquina para la expresión de miedo. En tanto que la clasificación de miedo en las pruebas previas de validación con la base de datos CK+ tuvo resultados notablemente superiores, es posible que parte del error aparente sea por calidad de la base de datos y no

por error real de la clasificación. También se puede observar el problema propio de la metodología de validación con humanos: muy baja tasa de clasificación de la instancia neutral de 64.6 % comparado con 97.5 % del sistema automático. Esto no muestra que los humanos no seamos buenos en determinar instancia facial neutral, sino que la naturaleza de la prueba induce a que los humanos atribuyan expresión facial a muestras que, en otras circunstancias naturales, no les habrían asignado ninguna expresión. Es probable, por ejemplo, que muchas de las expresiones con un ligero tono de tristeza hayan sido clasificadas como tristeza por los humanos debido a esto, lo cual explicaría la relativamente alta tasa de acierto de tristeza de 76.7 % comparada con 50.0 % de la evaluación automática. Creemos que buena parte del error de clasificación neutral es que los humanos clasificaron casi todas las muestras equívocas neutral/tristeza como tristeza, elevando así artificialmente la clasificación de tristeza, pero deteriorando fuertemente la clasificación de neutral. Por último, la expresión de ira. La clasificación humana tiene resultados notablemente superiores que la clasificación automática, de 78.8 % contra 59.2 %. Esta discrepancia es explicada por la naturaleza de las imágenes. En la base de datos validada CK+, en la expresión de ira ningún individuo muestra los dientes, lo que corresponde a las unidades de acción estándar de expresión facial. No obstante, en la base de datos KDEF aproximadamente el mitad de los individuos muestra los dientes en las muestras etiquetadas como ira. Esta ira impostada, que se acerca más a la ira caricaturesca que a la expresión natural de ira del humano, es fácilmente clasificada por un humano, pero un sistema que ha sido entrenado por muestras en las cuales ninguna expresión de ira tiene estas características caricaturescas, naturalmente clasificará las muestras como pertenecientes a otras expresiones que se acerquen más a estos patrones (disgusto y miedo). En la figura 45 se muestran muestras de ira de la base de datos KDEF y algunas muestras de ira de la base de datos CK+ que ilustran la diferencia. Se puede observar que en todas las muestras de ira de la base de datos KDEF los individuos muestran los dientes. Esto ocurre en aproximadamente 35-40 % de las muestras de ira de esta base de datos. En oposición, en ninguna de las muestras de ira de la base de datos CK+ el individuo muestra los dientes, de modo que el sistema de clasificación entrenado con esta base de datos tiene dificultad en clasificar muestras de esta naturaleza y, en cambio, las clasifica principalmente como muestras de disgusto, miedo o tristeza. Disgusto y miedo es razonable, por cuanto muchas muestras de estas dos expresiones tienen estas características. Tristeza, en cambio, ocurre de manera sistemática en 4 muestras particulares de ira de la base de datos KDEF. A nuestro modo de ver, es bastante comprensible que el sistema las clasifique como pertenecientes a la clase de tristeza, pero mostramos los casos referidos en la figura 46 para que el lector evalúe por sí mismo.

Así mismo, se observan tasas deficientes de clasificación de tristeza y de miedo. En el caso de miedo, sin embargo, incluso esta clasificación tiene resultados mejores que los de la obtenida por clasificación humana (50 % vs. 43 %). Tristeza, no obstante, es una expresión que no podemos entender bien observando la matriz de confusión. Clasificación de 50 % contra 76.7 % de humanos. Parte de esto es que los humanos tienen mediocre clasificación de neutral, entonces muchas muestras con posible equívoco entre

Figura 45. Muestras de ira de las bases de datos KDEF (arriba) y CK+ (abajo)

Figura 46. Muestras de ira de las bases de datos KDEF clasificadas como tristeza

tristeza y neutral son clasificadas por los humanos como tristeza, elevando así la tasa de clasificación, sacrificando con ello la tasa de neutral. Sin embargo, el sistema de clasificación automático clasifica muchas de las muestras de tristeza como miedo (y, como se observará posteriormente, el sistema entrenado por la misma base de datos KDEF hace lo mismo). Es decir, el error de clasificación no es tristeza clasificada como neutral sino tristeza clasificada como miedo. Al hacer observación manual de las muestras de tristeza de esta base de datos, en realidad no se observan muchas muestras de calidad dudosa que puedan producir problemas de clasificación. Nuestra única hipótesis es que en la base de datos CK+ las muestras de tristeza tienen microexpresión evidente en la boca, pero el ceño se mantiene normal. En cambio en la base de datos KDEF un buen número de muestras de tristeza tienen ceño fruncido. Posiblemente esto, que no representa inconveniente para la clasificación humana, sea suficiente para inducir a error a un sistema entrenado por muestras que no tengan estas características. Para ilustrar esto, en la figura 47 se observan muestras prototipo de tristeza de las dos bases de datos. Salvo en una de las muestras de la base de datos CK+, en ninguna hay una expresión de ceño (y, de hecho, esta muestra es una particularidad, pues en sólo aproximadamente 6 % de las muestras de la base de datos CK+ de tristeza hay algún grado de microexpresión de ceño). En cambio, en la base de datos KDEF esta expresión de ceño es mucho más frecuente, con aproximadamente 45 % de las muestras presentando algún grado de microexpresión de ceño. Debido a esto, creemos que con el fin de reducir el error global el sistema de clasificación prefiere clasificar muchas de las muestras de tristeza como miedo que clasificar muchas de las muestras de miedo como tristeza, lo que, en todo caso, sólo mejoraría marginalmente la tasa de clasificación de esta última expresión.

Figura 47. Muestras de tristeza de las bases de datos KDEF (arriba) y CK+ (abajo)

Tal como se observará más adelante, incluso al hacer entrenamiento y validación con la base de datos KDEF, el sistema automático con frecuencia clasifica muestras de tristeza como miedo, lo que constituye evidencia de nuestra hipótesis precedente.

Pese a esto, al comparar los resultados con los obtenidos con clasificación humana, se observa que los sistemas de clasificación automática son al menos comparables a la evaluación realizada por humanos, incluso con una base de datos de naturaleza completamente desconocida para los clasificadores automáticos.

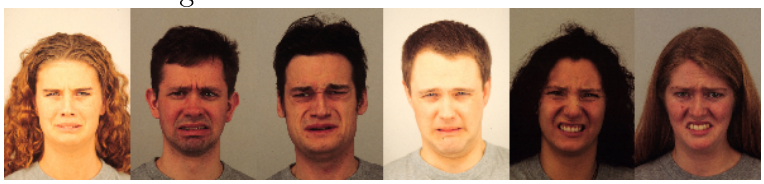
A continuación se hicieron las pruebas de validación de la base de datos KDEF entrenada con la misma base de datos. Los resultados se muestran en la tabla 39.

Tabla 39. Validación de la base de datos KDEF con sistema entrenado con la misma base de datos

	Ira	Dis.	Mie.	Ale.	Tri.	Sor.	Neu.
Ira	**69.4**	12.8	6.1	2.8	2.2	0.0	6.7
Disgusto	5.6	**83.3**	0.0	0.0	11.1	0.0	0.0
Miedo	2.2	1.1	**52.2**	2.2	15.6	17.8	8.9
Alegría	0.0	0.0	0.0	**97.2**	0.0	0.0	2.8
Tristeza	3.3	3.3	20.6	0.0	**52.8**	0.6	19.4
Sorpresa	0.0	0.0	0.0	9.4	0.0	**81.7**	8.9
Neutral	0.0	0.0	0.6	0.0	3.3	2.8	**93.3**

Los resultados de clasificación global son de 75.7 %, levemente superiores a los resultados obtenidos con la clasificación basada en sistemas entrenados con la base de datos CK+ de 73.6 %. Nótese, sin embargo, que el sistema entrenado con la KDEF está en capacidad de clasificar con mayor facilidad las muestras de ira caricaturescas, por cuanto ha sido entrenado con muestras de similares características. Esto explica que la mayor parte de la diferencia de clasificación está dada por la clasificación de ira. Adicionalmente, buena parte de las muestras de disgusto son clasificadas como miedo, a diferencia del sistema entrenado con KDEF. Consideramos que esto obedece a que

Figura 48. Muestras de disgusto de la base de datos KDEF



muchas de las muestras de disgusto efectivamente parecen muestras de miedo según las unidades de acción. En la figura 48 mostramos algunas de las muestras de disgusto de la base de datos KDEF que permiten ilustrar esto. En nuestra opinión, no es inusual que un sistema automático entrenado con otra base de datos clasifique muchas de estas muestras como miedo. De hecho, el resultado de evaluación por humanos es de 72.2 % para esta expresión, incluso inferior que el 79.2 % obtenido con la clasificación automática entrenando con la base de datos CK+, de modo que consideramos que el resultado de 83.3 % obtenido al entrenar y validar con la misma base de datos KDEF en ningún caso es muestra de dificultad de generalización.

En cualquier caso, los resultados de validación comparables en la clasificación de la expresión facial usando entrenamiento con la misma base de datos y con otra base de datos son una evidencia indiscutible de la gran capacidad de generalización. La base de datos KDEF es completamente desconocida para el sistema entrenado con CK+ en tamaño de imágenes, resolución e incluso en patrones característicos de las distintas expresiones faciales (como la predominancia de expresiones caricaturescas de ira y expresiones de tristeza exageradas), y aún así los resultados fueron similares/superiores al desempeño de los humanos o la clasificación con sistemas entrenados con la misma base de datos.

Previamente señalamos que uno de los principales tropiezos al usar la base de datos KDEF es que sus muestras no fueron tamizadas dependiendo de su calidad de representación de la expresión facial. Teniendo en cuenta que la base de datos CK+ es validada por panel de expertos y los resultados de clasificación son bastante notables, más el hecho de que las muestras menos representadas en la CK+ son miedo, tristeza e ira, nuestra hipótesis es que en la base de datos KDEF la representación de estas 3 clases es la más defectuosa. Para probar esta hipótesis, en la tabla 40 mostramos los resultados de validación de la base de datos CK+ con un sistema entrenado con la base de datos KDEF.

En primera instancia, se puede determinar que las tasas de clasificación de tristeza, ira y miedo son las más inferiores, lo que corresponde con nuestra idea inicial de que estas tres clases son las de menor calidad de representación en la base de datos no tamizada KDEF y por ello producen alta tasa de error. De esta forma se puede confirmar que la base de datos KDEF es un reto muy importante para cualquier sistema de clasificación entrenado con imágenes de otras bases de datos y pese a ello nuestro sistema de clasificación obtuvo resultados notables, superiores incluso a la clasificación humana pese a tener varias desventajas importantes: i. En la clasificación de ira los

Tabla 40. Validación de la base de datos CK+ con sistema entrenado con la base de datos KDEF

	Ira	Dis.	Mie.	Ale.	Tri.	Sor.	Neu.
Ira	**50.9**	0.0	0.0	0.0	1.8	0.0	47.3
Disgusto	8.8	**79.4**	0.0	2.4	0.0	0.0	9.4
Miedo	0.0	15.0	**53.3**	1.7	3.3	18.3	8.3
Alegría	0.0	2.6	0.0	**97.4**	0.0	0.0	0.0
Tristeza	8.3	0.0	11.7	0.0	**41.7**	5.0	33.3
Sorpresa	0.0	0.0	0.4	0.0	0.0	**95.8**	3.8
Neutral	0.0	0.0	1.8	0.0	0.4	4.6	**93.2**

humanos reconocen y clasifican correctamente las muestras impostadas caricaturescas, mientras que un sistema entrenado con muestras naturales de ira no está en capacidad de reconocer estas muestras. ii. Los humanos atribuyen en exceso expresión a la mayoría de las muestras debido a la naturaleza de la prueba, lo que se confirma con la baja tasa de clasificación de neutral (63.0 %) contra el sistema automático (97.5 %). Es razonable que en una clasificación espontánea los humanos no sean tan proclives a atribuir expresión a un rostro y con ello la clasificación de tristeza y de ira es deteriorada. iii. Esta prueba no es de máquina vs. promedio de humanos. En [60] se mostró cómo los individuos jóvenes tienen mejor reconocimiento de la expresión facial y las mujeres son notablemente superiores en el reconocimiento de expresiones negativas, particularmente disgusto e ira. No obstante, en los resultados mostrados de validación KDEF por un panel de humanos, este panel fue conformado por 272 mujeres jóvenes (media 21 años, desviación estándar 2.1, entre 18 y 37 años), de modo que en todas nuestras pruebas, incluyendo validación de KDEF con entrenamiento con otra base de datos, los resultados fueron similares o superiores contra el subconjunto de humanos con mejor desempeño de clasificación. Naturalmente, se espera que contra un conjunto más representativo del universo de humanos en edad y en sexo, la ventaja del sistema automático sea superior.

Es posible argumentar que los humanos tienen similares restricciones que la máquina si la base de datos no es tamizada. En consecuencia, nuestra siguiente prueba fue la validación de la base de datos CK+ por un panel de 41 evaluadores humanos. La metodología consistió en un software que muestra aleatoriamente ejemplos de todas las expresiones e instancia neutral y posteriormente, en intervalos de 5 segundos, visualiza nuevas muestras que el usuario debe clasificar. En promedio cada evaluador clasificó alrededor de 150 muestras, para un total de 6137 muestras evaluadas, de manera que la representación estadística de la base de datos en su totalidad está garantizada.

Los resultados están mostrados en la tabla 41.

Al contrastar entre la tabla 38 y la tabla 41 se observa que en la valoración humana de la base de datos CK+ el error de clasificación de instancia neutral es muy inferior

Tabla 41. Validación de la base de datos CK+ por un panel de humanos

	Ira	Dis.	Mie.	Ale.	Tri.	Sor.	Neu.
Ira	**66.3**	13.7	2.9	0.0	13.7	1.0	2.44
Disgusto	18.9	**78.2**	0.9	0.0	0.5	1.4	0.47
Miedo	1.6	18.2	**56.3**	1.6	4.17	16.2	2.1
Alegría	0.0	0.0	0.0	**99.0**	0.5	0.5	0.0
Tristeza	4.3	1.6	1.6	0.0	**80.5**	0.5	11.3
Sorpresa	0.0	0.0	1.4	0.0	0.0	**93.8**	4.7
Neutral	2.8	0.5	0.5	1.9	0.5	0.0	**93.9**

(6 % contra 37.4 %). Esto reivindica la menor calidad de la base de datos KDEF en la representación universal de la expresión. No obstante, pese a que no se puede hacer una comparación directa porque nuestro panel de humanos no es conformado por los mejores clasificadores (sólo 35.6 % del panel fueron mujeres), los resultados de clasificación humana de la base de datos CK+ fueron de aproximadamante 81 % global, superior al 71 % de clasificación obtenido por el panel de humanos con la base de datos KDEF (aunque muy inferiores a nuestros resultados de clasificación automática de la base de datos CK+).

7.6.2. Pruebas dinámicas con la base de datos CK

Buena parte de las secuencias de video de la base de datos CK fue eliminada para la versión extendida CK+. No obstante, nuestras pruebas se hicieron de manera paralela con las bases de datos CK y CK+, con el fin de poder contrastar nuestros resultados también con resultados publicados, mucho más numerosos, usando la base de datos CK. Sin embargo, algo que nos llamó ciertamente la atención es que hay resultados publicados con tasas de acierto muy elevadas, incluso en algunos casos cercanas a 99 % con la base de datos CK original. Es decir, clasificación casi perfecta pese a que la base de datos tenía numerosas muestras incorrectamente etiquetadas. En la tabla 42 mostramos algunos de estos casos relevantes, todos con validación realizada con la base de datos CK.

En primera instancia, en la tabla se puede observar que casi todas las metodologías de validación usan *random n-folded*. Los mejores resultados de clasificación fueron obtenidos usando esta metodología o *random LSO*[6]. Mostramos dos pares de resultados en los cuales se usó tanto validación *leave-subjects-out* ([156] y nuestra propuesta) y en

[6]Con el fin de diferenciar *leave-subjects-out* del convencional LSO, en esta tabla denominamos *leave-subjects-out* como la metodología que excluye individuos completos del conjunto de entrenamiento, en oposición a *random* LSO, en el cual se excluyen muestras aleatorias, de manera que puede haber muestras del mismo individuo tanto en entrenamiento como en validación.

Tabla 42. Comparación de resultados usando distintas metodologías de validación

Technique	Clas. rate	Validation
Linear AdaSVM [92]	93.5	Leave-one-subject-out
SVM+AdaBoost [7]	93.3	Leave-one-subject-out
Geometric deformation + SVM [84]	97.7	Random LSO
Eigenspace PCA [115]	77.5	Random folded (6 clases)
Dynamic haar-like features [183]	96.6 (6 clases)	Random folded
LBP+KDI [196]	94.88	Random 10-folded
AAM [156]	89.14	Leave-subjects-out
	94.85	Random 10-folded
Curvelet+LBP [142]	90.33	Random 5-folded
LBP+SVM-RBF [151]	88.9	Random 10-folded
VLBP+LBP-TOP [193]	96.26	Random 10-folded (6 clases)
Multiclass AdaBoost+SVM [53]	97.35	Random 5-folded
Our proposal	94.33	Leave-subjects-out
Our proposal (conventional SVM-rbf)	99.13	Random 10-folded

ambos casos se observa que la validación *random n-folded* produce resultados considerablemente más elevados de clasificación.

En nuestra opinión, las tasas muy elevadas de clasificación con la base de datos no depurada CK pueden ocurrir debido a dos posibles problemas: i. El de validación *random n-folded* convencional con un sistema experto de clasificación, que permite estas tasas de acierto pese a que la metodología en realidad no es satisfactoria. ii. Selección y eliminación manual de muestras en la base de datos CK previamente al proceso de entrenamiento y validación. Sin embargo, consideramos que esto tampoco es un proceso adecuado, por cuanto es difícil establecer la frontera entre selección manual rigurosa y eliminación de muestras de dudosa calidad. En cualquier caso, esto pone a nuestro sistema en desventaja, por cuanto en ninguna prueba excluimos o cambiamos la etiqueta de ninguna muestra manualmente, de manera que nuestro sistema tiene que lidiar (e incurrir en error) con estas muestras insatisfactorias.

En esta sección mostraremos algunos de los casos en los que nuestro sistema tuvo errores de clasificación con la base de datos CK, incluidos en las matrices de confusión, pero que en muchos casos corresponden a muestras que posteriormente habrían de ser excluídas de la base de datos para la revisión CK+. Incluimos únicamente muestras para las cuales hay autorización de publicación de imágenes, con el fin de que el lector pueda observar tanto la evolución dinámica de los puntajes de expresión facial como la imágenes correspondientes en la secuencia de video. La referencia de colores es: azul, ira; rojo, disgusto; verde, miedo; cian, alegría; negro, tristeza; amarillo, sorpresa y magenta, neutral.

Figura 49. Evolución dinámica de los puntajes de la expresión ira correctamente clasificada

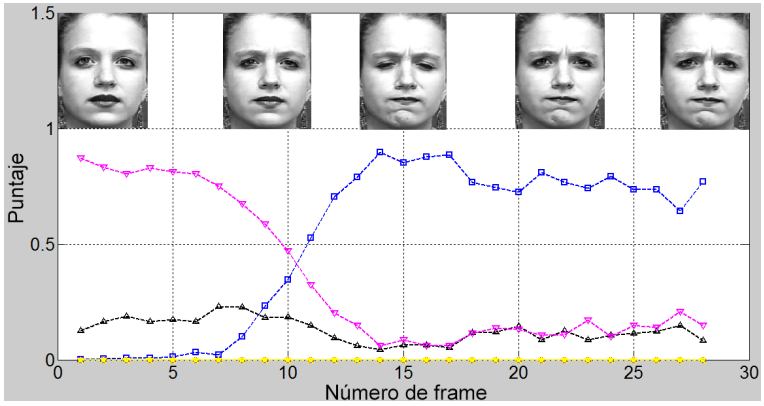
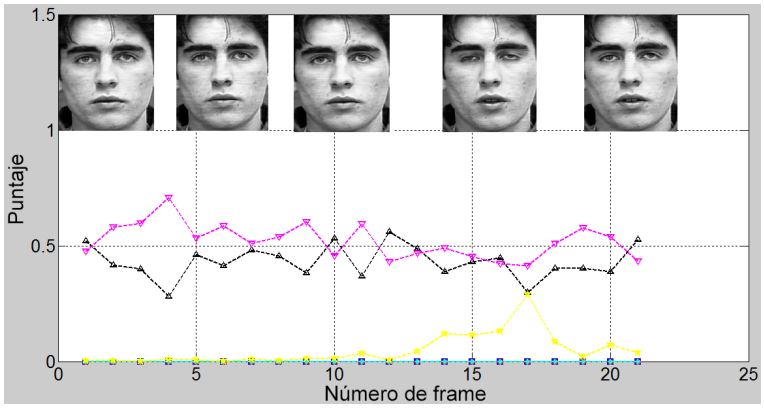


Figura 50. Evolución dinámica de los puntajes de la expresión ira incorrectamente clasificada



Ira: En la figura 49 mostramos la evolución dinámica de una muestra correspondiente a la expresión ira correctamente clasificada. En magenta se observa el puntaje de neutral. Al comienzo, cuando la expresión aún no está formada, este puntaje es máximo, correctamente. Posteriormente cuando la persona empieza a formar la expresión de ira, el puntaje correspondiente, en azul, se incrementa, también correctamente.

En un caso de clasificación incorrecta, el puntaje de ira no es el predominante. En la figura 50 se observa este caso. Durante casi toda la evolución dinámica, el puntaje es disputado entre neutral (en magenta) y tristeza (en negro), y nunca el puntaje de ira supera ni siquiera el umbral mínimo. No obstante, consideramos que esta clasificación errónea es ocasionada por la baja representación de la muestra (que posteriormente no fue incluida en la base de datos CK+, así como todos los demás ejemplos mostrados en esta subsección) y, de hecho, aunque esto puede ser bias de observación (aunque soportado por observadores independientes), la secuencia en realidad sí parece ser de entre neutral y tristeza.

Figura 51. Evolución dinámica de los puntajes de la expresión disgusto correctamente clasificada

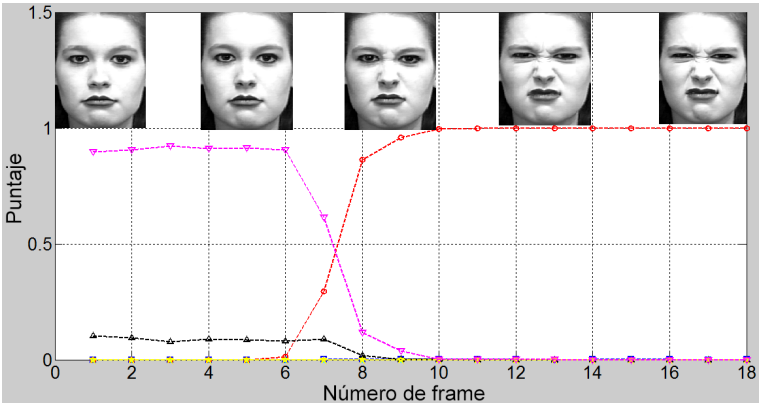
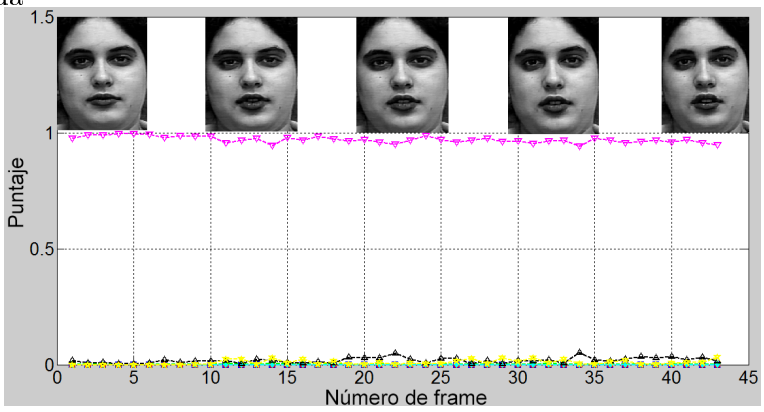


Figura 52. Evolución dinámica de los puntajes de la expresión disgusto incorrectamente clasificada



Disgusto: En la figura 51 mostramos una clasificación dinámica normal correcta de la expresión disgusto. Al inicio, cuando la expresión aún no es formada, el puntaje dominante es neutral. Posteriormente, a partir del cuadro 7, el puntaje de disgusto crece alcanza nivel máximo algunos cuadros después.

Un ejemplo de clasificación incorrecta de la expresión disgusto es mostrado en la figura 52. Se observa que el puntaje predominante en toda la secuencia es neutral. Sin embargo, nuevamente consideramos que esta muestra efectivamente parece neutral por inspección visual, pero, con el fin de tener consistencia metodológica, su clasificación fue consignada como incorrecta en la matriz de confusión.

Miedo: En la figura 53 mostramos la evolución dinámica de una expresión de miedo correctamente clasificada. Al comienzo de la secuencia el puntaje es indeciso entre tristeza y neutral (según nuestras reglas de decisión, en esos casos el cuadro es definido como neutral), aunque en cierto punto, entre los cuadros 8 y 11 hay ventaja de tristeza. No obstante, en cuanto la expresión de miedo se empieza a formar, a partir del cuadro 9,

Figura 53. Evolución dinámica de los puntajes de la expresión miedo correctamente clasificada

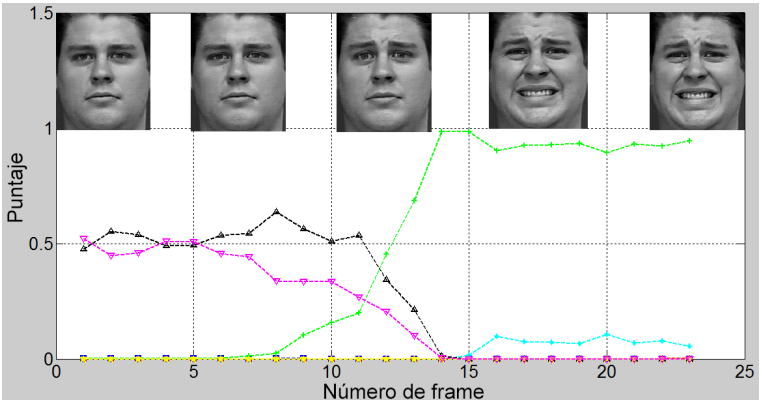
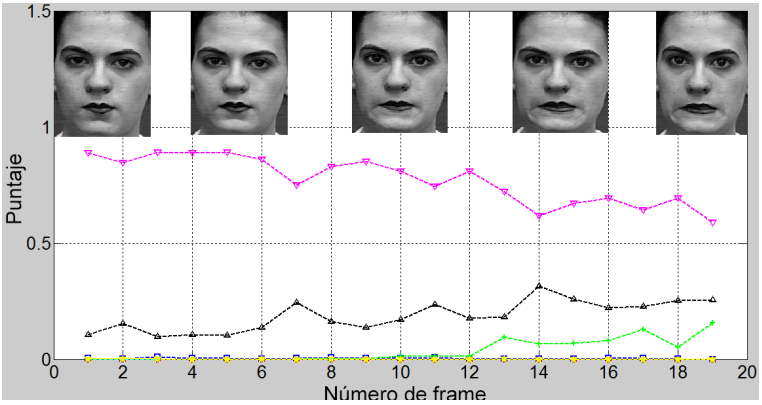


Figura 54. Evolución dinámica de los puntajes de la expresión miedo incorrectamente clasificada



el puntaje de miedo, en verde, se incrementa y algunos cuadros después alcanza valores casi máximos, clasificando correctamente la expresión.

En contraste, en la figura 54 se muestra una expresión de miedo incorrectamente clasificada, con puntaje predominante de neutral en toda la secuencia. En este caso consideramos que el error es absoluto, por cuanto a nuestro modo de ver la expresión es claramente de miedo en los últimos cuadros de la secuencia. Sin embargo, incidentalmente para esta misma persona hay otra secuencia en la base de datos con expresión de miedo más marcada. En este caso la evolución dinámica se observa en la figura 55. Nótese cómo en este caso, con expresión más fuerte, en cuanto la expresión empieza a ser marcada, alrededor del cuadro 15, el puntaje de miedo se incrementa hasta valores máximos.

Alegría: En la figura 56 se observa la evolución dinámica de la expresión de alegría clasificada correctamente. En tanto que es una de las expresiones de más fácil clasificación, el sistema no tiene ningún inconveniente en asignarle puntaje máximo, en color

Figura 55. Evolución dinámica de los puntajes de la expresión miedo correctamente clasificada, segundo caso

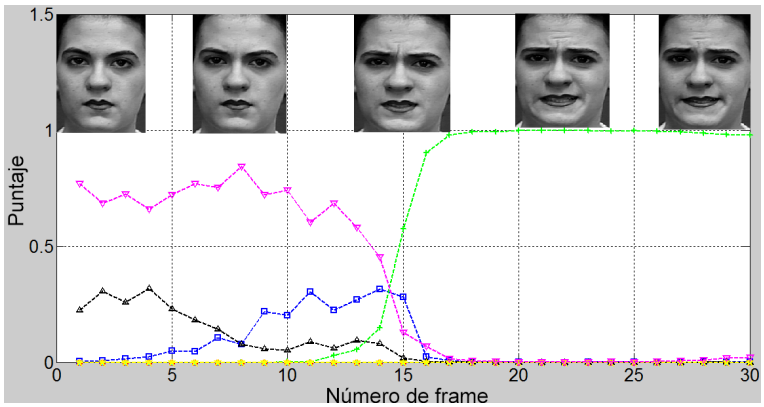
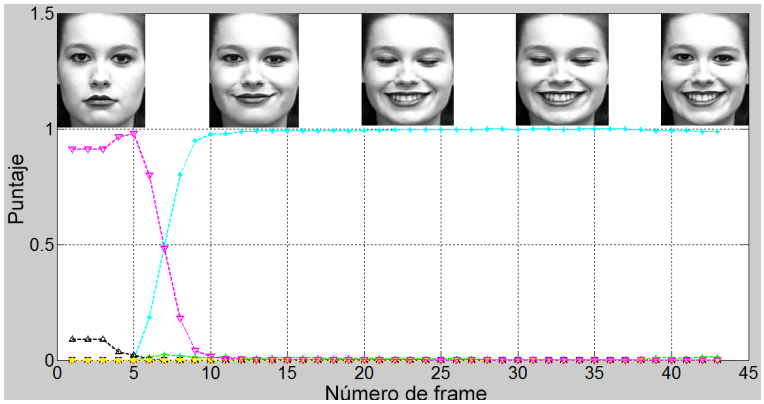


Figura 56. Evolución dinámica de los puntajes de la expresión alegría correctamente clasificada



cian, incluso cuando la alegría no está completamente formada, alrededor del cuadro 10. Naturalmente, cuando la expresión es más evidente, el puntaje continúa siendo máximo.

Lamentablemente, ninguna de las pocas muestras de alegría incorrectamente clasificadas corresponde a individuos con autorización de publicación de imágenes. Sin embargo, podemos mostrar un fenómeno muy frecuente, mostrado en la figura 57. Se observa que en el cuadro 4 el puntaje de miedo es predominante, aunque posteriormente el puntaje de alegría es máximo. En muchos casos se observó que justo previamente a la formación de la expresión de alegría ápice, hay un puntaje pico de miedo. En el caso mostrado esto no representó ningún problema de clasificación, pero en otros casos con clasificadores más simples se observó que buena parte del error de clasificación de alegría es por clasificación como miedo, tanto en nuestras pruebas anteriores en el capítulo 5 como en el estado del arte. Este error, que suena extraño porque en apariencia la expresión miedo no es cercana a la expresión alegría, entonces es, en nuestra opinión, ocasionado porque en algún momento transitorio de la formación de alegría el rostro adquiere características típicas de miedo.

Figura 57. Evolución dinámica de los puntajes de la expresión alegría correctamente clasificada, segundo caso

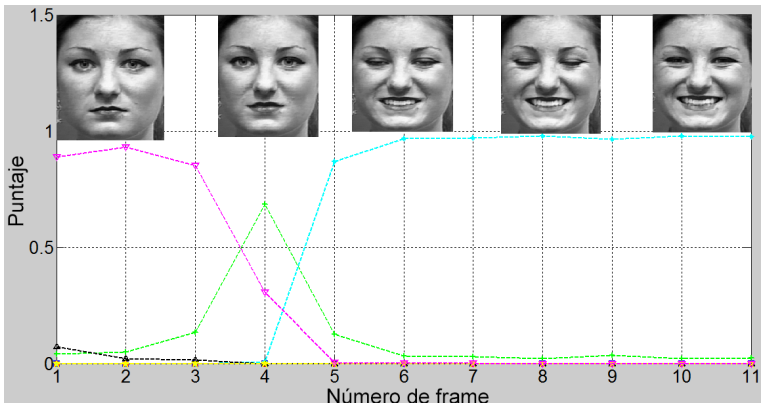
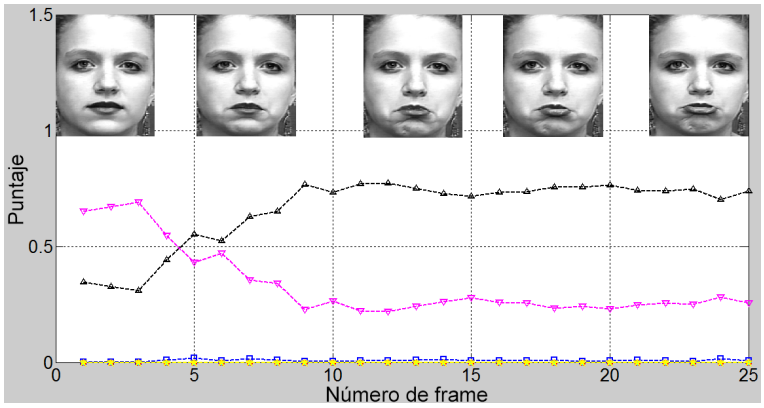


Figura 58. Evolución dinámica de los puntajes de la expresión tristeza correctamente clasificada



Tristeza: En la figura 58 se observa la clasificación correcta de la expresión tristeza, en negro. El puntaje de tristeza nunca alcanza valores máximos, pero sí suficientes para que a partir del cuadro 7 la expresión sea clasificada correctamente como tristeza sin ambigüedad.

Si bien el mayor aporte de error de clasificación de tristeza es cuando es incorrectamente clasificada como neutral, debido a la gran similitud entre las dos expresiones, no ocurrió esto en ninguna de las muestras de posible publicación. Sin embargo, hay una muestra de clasificación errada de tristeza, equívocamente entre disgusto e ira, cuya evolución dinámica se muestra en la figura 59. Nuevamente usamos un conjunto de observadores independientes y su valoración fue principalmente de muestra de enojo y no de tristeza, coincidente con nuestra opinión. De manera que consideramos que es otro caso de error producido por la calidad de la muestra, que posteriormente fue eliminada en la base de datos CK+ revisada.

Figura 59. Evolución dinámica de los puntajes de la expresión tristeza incorrectamente clasificada

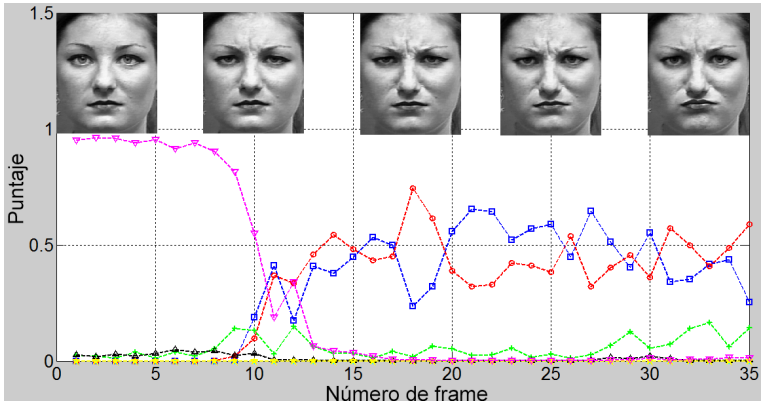
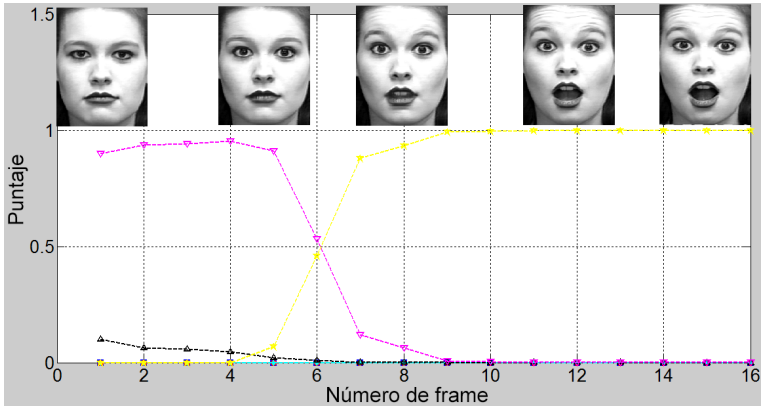


Figura 60. Evolución dinámica de los puntajes de la expresión sorpresa correctamente clasificada



Sorpresa: En la figura 60 se observa en amarillo la evolución dinámica de la expresión de sorpresa. El puntaje es claramente máximo con la expresión formada debido a la relativamente fácil clasificación de esta expresión.

En la figura 61 , no obstante, hay una muestra de clasificación incorrecta. Cerca del final de la secuencia, luego del cuadro 20, el puntaje de sorpresa se incrementa gradualmente, pero nunca alcanza a tener un valor suficiente para clasificar la muestra como sorpresa. Pese a que esta muestra, al igual que todo el resto de muestras clasificadas incorrectamente mostradas en esta sección, fue descartada por los autores de la base de datos en la versión extendida CK+, consideramos que la expresión de sorpresa es clara y podría haber sido clasificada sin inconveniente. No obstante, para esta misma persona hay otra muestra etiquetada de sorpresa, con expresión más marcada, mostrada en la figura 62.En este caso el puntaje de sorpresa es máximo con la expresión fuertemente marcada. Curiosamente, en el cuadro 9 ya hay puntaje significativo de sorpresa, pero se puede observar que en apariencia la expresión es más débil que en las de la figura previa. Creemos que esto sucede porque si bien la boca aún no está abierta, la expresión

Figura 61. Evolución dinámica de los puntajes de la expresión sorpresa incorrectamente clasificada

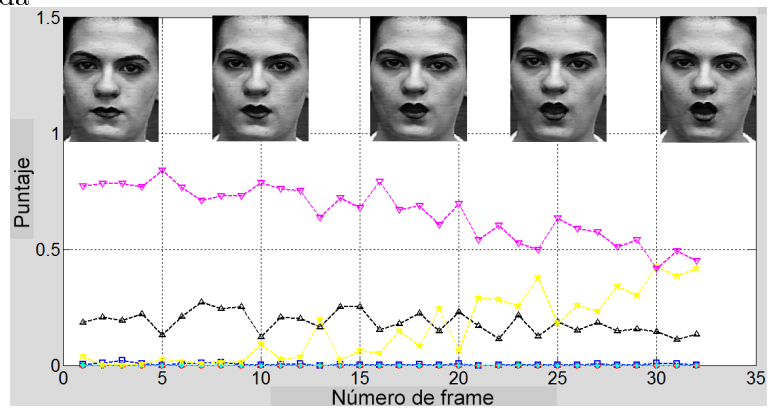
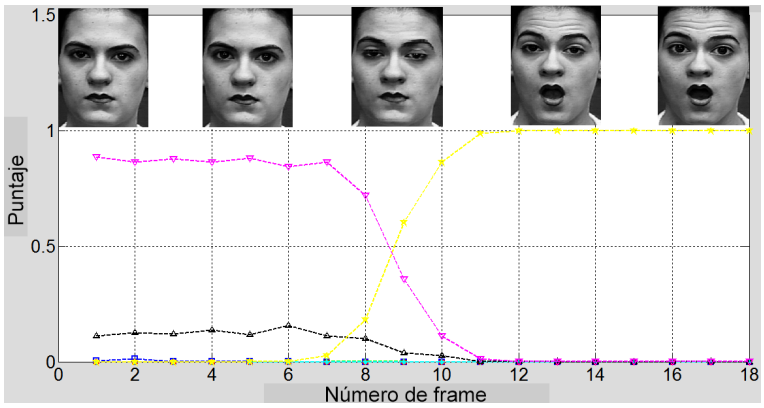


Figura 62. Evolución dinámica de los puntajes de la expresión sorpresa correctamente clasificada, segundo caso



de las cejas es suficiente para que el sistema asigne puntaje significativo a sorpresa.

Todas las muestras clasificadas erróneamente mostradas en las figuras de evolución dinámica de la expresión facial son muestras que pertenecen a la base de datos original Cohn-Kanade y no pertenecen a la versión extendida CK+. Es decir, estas muestras fueron posteriormente catalogadas como de pobre calidad, de modo que no se incluyeron en la siguiente versión. Además de estos ejemplos mostrados, hay numerosos ejemplos adicionales de muestras que fueron descartadas de la base de datos original y producían frecuentemente error de clasificación. De hecho, el error es de doble naturaleza: i. Una muestra con etiqueta incorrecta que no sea representativa de la clase, posiblemente sea clasificada en teoría erróneamente; pero en realidad la clasificación no estaba mal, sino la etiqueta de la muestra, lo cual empeora los resultados de la matriz de confusión. ii. Una muestra incorrecta hace parte frecuente del conjunto de entrenamiento en una metodología *random 10-folded* o *leave-subjects-out*. Es decir, la muestra no sólo incrementa el error de clasificación de sí misma, sino que induce a error al clasificador al entrenarlo con patrones que en realidad no son representativos de la clase etiqueta-

da. Este problema, aunque más imponderable, es posiblemente peor, pues compromete la clasificación de numerosas muestras con etiqueta correcta debido al entrenamiento inapropiado del clasificador. Debido a estas razones nos es incomprensible que existan numerosos clasificadores con resultados publicados de clasificación con la base de datos CK superiores a 95 %, incluso en algunos casos cercanos a 100 %, salvo que exista un error metodológico de sobre clasificación con metodología *random n-folded*, tal como se mostró en la tabla 30 o descarte manual de muestras de la base de datos, lo cual pone a nuestro sistema en clara desventaja para hacer una comparación directa.

7.7. Complejidad computacional

Uno de los objetivos del trabajo realizado en esta tesis es la aplicabilidad de los algoritmos para tiempo real. En esta sección realizaremos un análisis de complejidad computacional de los algoritmos usados, enfocándonos en aquéllos que representan el mayor desafío computacional, sea por requerimiento de memoria o por tiempo de procesamiento. En primera instancia, definiremos si el reconocimiento de la expresión facial usando nuestra metodología es un problema factible según la teoría de complejidad computacional.

Según la convención, un algoritmo es factible en tanto que pueda ser ejecutado en tiempo polinomial. Esto es, si existe algún polinomio p tal que el algoritmo sea ejecutado en máximo $p(n)$ para entradas de longitud n. Esta clases de problemas son denotados como P. Un problema de búsqueda definido por una relación R es NP si la relación es computacionalmente eficiente y si tales soluciones son cortas. De manera general, R es un probelma de búsqueda NP si existe un algoritmo en tiempo polinomial tal que dado x e y, decida si $(x, y) \in R$ y si hay un polinomio p tal que si $(x, y) \in R$, entonces $|y| \leq p(|x|)$. De manera análoga, un problema de decisión L es un problema NP si existe alguna relación NP tal que $x \in L$ si y sólo si existe un y, tal que $(x, y) \in R$. El problema L es un problema de decisión NP si hay un algoritmo de tiempo polinomial $V(\cdot, \cdot)$ y un polinomio p tal que $x \in L$ si y sólo si hay un y, $|y| \leq p(|x|)$ que acepta $V(x, y)$.

Para efecto del análisis de complejidad, separamos los algoritmos según las etapas de detección de rostro, extracción de parámetros, reducción de dimensiones y clasificación. Las etapas de selección de parámetros y entrenamiento no están limitadas por la restricción de costo de tiempo real, de manera que no tienen relevancia en el análisis de costo. La nomenclatura usada es $O(n)$ representa el costo de complejidad para resolución de n bits $T(n)$ es el costo del algoritmo T y $M(n)$ representa el costo de complejidad para multiplicación de n bits.

7.7.1. Detección de rostro

El algoritmo de detección require del uso de la imagen integral, que se refiere a una imagen tal que el valor de un pixel (x, y) es la suma de los pixeles arriba y a la izquierda de (x, y), inclusive, en la imagen original. Este procesamiento es computacionalmente

simple, realizándose en una sola iteración con un número de sumas equivalente al tamaño de la imagen original. Para las imágenes de 480×640 usadas en las bases de datos CK y CK+, esto equivale a 307,200 sumas, cuyo costo de procesamiento es despreciable en función del resto del algoritmo. La complejidad de costo para esta etapa es $T(n) \in O(n)$, donde n es la longitud binaria de los números (8 bits para las imágenes monocromáticas usadas en este trabajo).

La detección de rostro en este trabajo es realizada mediante el uso de ventanas de filtrado tipo Haar. Estas ventanas son un subconjunto de un total posible de alrededor de 117.941 parámetros mediante un algoritmo de aprendizaje AdaBoost. Para este trabajo se usaron 22 ventanas de tamaño promedio 18 pixeles. El algoritmo de detección truncada hace pasar cada subventana de imagen consecutivamente por los filtros Haar. Si la subventana es rechazada, pasa por el siguiente filtro y así sucesivamente hasta completar las 32 ventanas, que es el peor escenario posible, correspondiente a no detección de rostro. El uso de la imagen integral reduce notablemente el costo de cálculo de esta etapa, por cuanto no es preciso hacer la multiplicación de cada parámetro Haar por la imagen original, puesto que la imagen integral permite determinar mediante una simple suma el valor acumulado de los pixeles en una región rectangular cualquiera.

Realizaremos, entonces, un análisis de este peor escenario posible. Es importante señalar que este caso es trivial, sin embargo, por cuanto si una imagen es rechazada por el banco de filtros de detección, implica que no hay rostro en la imagen y, en consecuencia, el resto del procedimiento de reconocimiento de expresión facial no es realizado. En este peor escenario posible cada imagen original de 307,200 pixeles es multiplicada por ventanas de tamaño promedio 18 pixeles. El algoritmo de multiplicación más ineficiente (algoritmo estándar) tiene complejidad $T(n) \in O(n^2)$, donde $n = 8$ para números de 8 bits. Los sistemas de cómputo usan algoritmos más adecuados con menor costo computacional, pero para tamaño pequeño de los números binarios el costo computacional del uso de un algoritmo ineficiente es, en todo caso, despreciable.

Este caso no corresponde, no obstante, a situaciones prácticas. En promedio, el primer filtro de detección Haar eliminó 45 % de las subventanas usando una base de datos con imágenes de bases de datos faciales mezcladas con imágenes aleatorias. Así mismo, las siguientes 4 ventanas Haar eliminaron en promedio 22 % de las subventanas totales. En consecuencia, en un escenario real el costo de cálculo es considerablemente inferior al peor escenario posible, en promedio 37.84 % del costo del peor escenario posible según nuestras mediciones con las imágenes usadas.

7.7.2. Extracción de parámetros

Si bien la etapa de extracción de parámetros involucra operaciones más complejas que los filtrados tipo Haar usados en la detección de rostro, el tamaño limitado de los rostros detectados, de 128×128 pixeles, hace que el costo de cálculo sea limitado. En primer instancia, la obtención de i_{t_c,p,θ_n} tiene como costo de complejidad una función lineal dependiente del número de vecinos espaciales P y el número de orientaciones espaciales N, multiplicado por una constante dependiente del tamaño de la región L. Debido a

que se realizaron pruebas con distintos valores, estos parámetros son constantes en las pruebas de entrenamiento y validación.

El peor escenario posible de esta etapa corresponde al no uso de nuestra estrategia de reciclado de información para la obtención de la textura volumétrica descrito en el capítulo 3, subsección 3.3.2.

Las distintas etapas del procedimiento de extracción de parámetros tienen complejidad de costo a saber: obtención de los gradientes acumulados, $T(n) \in O(M(n))$; obtención de las vecindades espaciales, $T(n) \in O(M(n))n^{1/2}$; obtención del código TPOEM, $T(n) \in O(n)$ y mapeo del código TPOEM, $T(n) \in O(n)$, donde n es la resolución numérica de los datos, en este caso de 32 bits.

7.7.3. Reducción de dimensiones

El proceso de reducción de dimensiones por reducción supervisada MCML fue descrito en el capítulo 4 subsección 4.2.2. El procedimiento de reducción de dimensiones es computacionalmente complejo y, dependiendo de la función objetivo, la dimensión inicial, la dimensión reducida y el protocolo de optimización, el costo de ejecución puede tardar considerablemente, incluso varias horas en el equipo de cómputo usado para este trabajo. No obstante, una vez obtenida la matriz de transformación A, la reducción de dimensiones consiste únicamente de una multiplicación matricial de la matriz de transformación por cada vector de entrada, cuya complejidad de costo es $T(n)(nm)$, donde n es la dimensión original y m es la dimensión reducida.

7.7.4. Clasificación

La etapa de entrenamiento del sistema de clasificación es una de las más exigentes del desarrollo del sistema dependiendo del sistema de clasificación seleccionado. Un clasificador basado en metaclasificadores débiles SVM+FDA tiene relativamente bajo costo de cálculo comparado con otras etapas del proceso tales como reducción de dimensiones o selección de parámetros, aunque el uso de máquinas *deep learning* eleva el costo de entrenamiento incluso a varios días de ejecución para el protocolo completo de entrenamiento y validación por LSO.

Sin embargo, estos costos no son relevantes en la evaluación de desempeño del sistema, por cuanto esta etapa no está incluida en el protocolo de validación de una muestra nueva. Los costos de clasificación por cada metaclasificador SVM ó FDA son $T(n) \in O(M(n))$. La clasificación por *deep learning* tiene un costo adicional de $T(n) \in O(M(n)log(n))$ con uso de funciones de excitación sigmoideas.

De acuerdo con este análisis de complejidad computacional, el problema no sólo es factible, sino que además la complejidad de costo computacional de cada etapa es limitada, en comparación con algoritmos que requieran de operaciones de costo computacional más elevado como multiplicaciones matriciales, inversiones matriciales, cálculos de determinante u operaciones hipergeométricas. En términos de operaciones matemáticas, las funciones más complejas se limitan al cálculo de funciones exponenciales, polinómi-

Tabla 43. Número de operaciones por etapa

Etapa	Número de operaciones
Detección de rostro	168 MFLO
Reducción de dimensiones	32 MFLO
TPOEM	41 MFLO
Clasificación	12 MFLO

cas o trigonométricas (por ejemplo en las operaciones de excitación sigmoidea de las máquinas *deep learning*, la obtención de vecinos espaciales interpolados en las imágenes de gradiente acumulado o la metaclasificación SVM polinómica). Por otra parte, si bien el costo de complejidad computacional es limitado en cada etapa haciéndolo un problema factible, el inconveniente está dado por el número de operaciones aritméticas que se debe hacer en cada etapa, incluso siendo operaciones sencillas. En la tabla 43 se muestra el número de operaciones de punto flotante MFLO (*Mega Floating-point operations*) requerido por cada etapa en el peor escenario posible.

Si bien las etapas de extracción de parámetros TPOEM y de clasificación incluyen algunas operaciones aritméticas más complejas que las multiplicaciones y sumas en la detección de rostro y la reducción de dimensiones, el costo de complejidad computacional de estas operaciones no es elevado, de manera que la restricción temporal más significativa se encuentra en la etapa de detección de rostro [7].

Si bien la velocidad de ejecución depende del equipo de cómputo usado, usamos como criterio la capacidad de cumplimiento de requerimiento de ejecución acotado por la cadencia de video en un equipo de cómputo convencional. Para este trabajo se usaron principalmente dos equipos de cómputo. El primero con procesador i5-2410M de segunda generación y el segundo con procesador i7-4700MQ de cuarta generación, ambos con 4GB de memoria. Naturalmente, en el segundo equipo de cómputo el tiempo de cálculo total es menor; no obstante, con ambos se cumplió el requisito de tiempo de ejecución sin inconvenientes. De hecho, habida cuenta de que el número máximo de operaciones de punto flotante requerido por los algoritmos es de 253 MFLO y los equipos de cómputo caseros actuales tienen capacidad de entre 40 y 200 GFLOPS, con costo de procesador de únicamente \$0.08 por GFLOPS [8], sin duda los requerimientos de costo computacional de este trabajo son muy inferiores que los disponibles a bajo costo en el mercado. Estos datos incluso muestran que el tiempo total de ejecución

[7] No obstante, desde el desarrollo de nuestros algoritmos de detección de rostro en 2013 hasta la actualidad ha habido avances significativos en la teoría de detección de rostro, incluyendo la aplicación más extensa de filtros dependientes y el uso de SVM en las etapas de filtrado, que reducen los costos de la detección de rostro en una imagen. Adicionalmente, los valores mostrados corresponden al peor escenario posible. Sin embargo, las técnicas de detección ponderada de rostros que usamos en este trabajo, descritas en el capítulo 2, sección 2.3, restringen la búsqueda sucesiva a regiones candidatas y excluyen regiones de baja probabilidad; este protocolo reduce el costo promedio de búsqueda de rostros en una secuencia de imágenes.

[8] Ver datos en https://en.wikipedia.org/wiki/FLOPS

podría ser del orden de unos cuantos milisegundos. No obstante, el análisis de costo computacional no incluye otros tiempos requeridos en la ejecución de los algoritmos, relacionados con lectura y escritura de datos en memoria. Este tipo de procedimiento añade tiempo considerable, teniendo en cuenta que el tamaño de los datos en algunas etapas del proceso, especialmente en detección de rostro y reducción de dimensiones, es considerable, y se debe hacer escritura y lectura de datos frecuentemente. Pese a ello, los algoritmos cumplieron sobradamente el requerimiento de ejecución restringido por la cadencia de video.

7.8. Conclusiones

En este capítulo se mostraron algunas pruebas complementarias realizadas durante el desarrollo de este trabajo que sirven para extraer conclusiones importantes sobre la generalización del sistema de reconocimiento de expresión en secuencias de video independientes recopiladas por el autor usando clasificación *naïve* Bayesiana, generalización de clasificación usando la base de datos KDEF sugerida en [98], comparación entre clasificación realizada por humanos y clasificación automática y pruebas derivadas de las clasificaciones iniciales realizadas con la base de datos CK en vez de la versión extendida CK+.

En primera instancia, conseguimos mostrar cómo el sistema de clasificación modificado para incluir información temporal por clasificación Bayesiana permitió realizar reconocimiento de expresión facial con tasas adecuadas, especialmente teniendo en cuenta que los videos recopilados por el autor no fueron tamizados. Es decir, eliminar manualmente secuencias de dudosa calidad de representación es una etapa deseable, con el fin de evitar el doble problema ocasionado por este tipo de muestras: i. clasificación errada de muestras que en realidad no son representativas. ii. Entrenamiento del sistema con muestras no representativas, que lo pueden conducir a error de clasificación de muestras apropiadas. Sin embargo, este proceso de tamizaje incluye reconocimiento de unidades de acción, micro expresiones y análisis cuidadoso de las muestras, para lo cual el entrenamiento normal requiere de varios años de aprendizaje [9]. En vez de ello, todas las muestras recopiladas fueron usadas en la validación, de manera que es razonable considerar que muchos de los aparentes errores de clasificación no pueden ser atribuidos a error del sistema de clasificación, sino a muestras defectuosas. Pese a esto, los resultados son satisfactorios y muestran que el entrenamiento de un sistema con la base de datos CK+ permite extrapolar la clasificación a imágenes o secuencias de video de otra naturaleza.

Nuestras pruebas con la base de datos KDEF arrojaron conclusiones muy interesantes. En primer lugar, los resultados de clasificación automática de la expresión facial en esta base de datos con clasificadores entrenados exclusivamente con la base de datos CK+ tuvo resultados ligeramente superiores que los obtenidos por clasificación de

[9] Ésta es una de las principales razones por las cuales no existen muchas bases de datos de expresión facial con validación estándar disponibles.

un conjunto de humanos que representa a los mejores humanos en reconocimiento de expresión facial (individuos jóvenes y mujeres exclusivamente). Es decir, el sistema automático probablemente es considerablemente superior al promedio de humanos en su generalidad, pese a que el sistema automático tenía una serie de desventajas en primera instancia: i. En casi la mitad de las muestras de ira en la base de datos KDEF los individuos muestran los dientes, que es una expresión artificial de ira no incluida en ninguna muestra de la base de datos CK+, pero cuyo reconocimiento para un humano no es complicado. Debido a esto los humanos tuvieron tasa mucho mayor de reconocimiento de esta expresión (78.8 % vs. 59.4 %), pero cuando el sistema fue entrenado y validado con la base de datos KDEF, con *leave-subjects-out*, el clasificador aprende mejor estas muestras caricaturescas y su tasa de desempeño se incrementa a 69.4 %. ii. Nuestros sistemas de clasificación penalizan fuertemente el error de clasificación de instancia neutral clasificada como expresión. Los humanos, en cambio, obtuvieron únicamente un 62.6 % de tasa de acierto de instancia neutral. Si bien no hay matriz de confusión en el trabajo publicado, creemos que buena parte de este error es neutral clasificada como tristeza o como ira. Esto eleva artificialmente la tasa de clasificación de estas dos expresiones, pero no corresponde a una clasificación natural, pues espontáneamente, en contraste, los humanos no atribuimos expresión facial a un rostro con muy ligeras expresiones. En cambio nuestro sistema requiere de una expresión sin ambigüedad notoria para ser clasificada como expresión, lo cual deteriora la clasificación de expresiones difíciles. iii. Las imágenes de la base de datos KDEF eran completamente desconocidas para nuestro clasificador. Naturalmente, también eran desconocidas para el panel de humanos. Pero en el caso de la clasificación automática, entrenada y validada con imágenes de la misma resolución, tamaño y perspectiva, así como con casi idénticas condiciones de iluminación, es muy incierto su desempeño con problemas nuevos [10]. Los humanos, en cambio, no tienen este problema, pues tenemos capacidad entrenada de reconocimiento de expresión facial en diversas circunstancias.

Adicionalmente, hicimos pruebas de validaciones en distintas configuraciones: KDEF entrenada con KDEF (pero respetando metodología de validación, usando *leave-subjects-out*) y CK+ entrenada con KDEF, que nos permitieron soportar nuestra hipótesis de que la no estandarización de la base de datos KDEF hace que muchas muestras de tristeza, ira y miedo no sean representativas de la expresión facial universal y generen dificultades de validación. Incluso así, la validación de la base de datos CK+ con sistema entrenado por KDEF tuvo resultados notables de 74 % de clasificación, que es en todo caso no muy inferior que la clasificación realizada por humanos de 81 %, con la ventaja añadida de que el error de clasificación del sistema está concentrado en un

[10] En realidad, el objetivo inicial de estas pruebas era básicamente el opuesto: pretendíamos mostrar resultados de entrenamiento KDEF y validación KDEF para luego mostrar cómo la validación con sistema entrenado CK+ podría tener resultados muy inferiores, para mostrar una alta dependencia de los clasificadores con la naturaleza de las imágenes. Sin embargo, encontramos el resultado contrario, con notables tasas de clasificación pese a todas las dificultades previstas, de manera que concluimos que la capacidad de generalización de nuestros parámetros y sistemas de clasificación son mucho mayores que los que esperábamos.

55 % en muestras de expresión clasificadas como neutral en contraste con la clasificación humana en la cual esta proporción es de únicamente 26 %, lo cual implica que el error de clasificación de una expresión clasificada como otra expresión en la clasificación humana es mucho más grande, y esto, a nuestro modo de ver, es menos deseable que el error de una expresión clasificada como neutral. De cualquier manera, consideramos que es necesario realizar una validación experta de la base de datos KDEF con el fin de permitir una comparación más clara entre clasificación humana y clasificación automática, pero creemos que nuestro trabajo fue suficientemente exhaustivo y detallado para sustentar nuestra hipótesis de que el sistema automático, incluso en condiciones de clara desventaja, tiene desempeño comparable o superior al reconocimiento realizado por humanos.

Por último, hicimos pruebas dinámicas con la base de datos CK. Nuestro principal objetivo con esto era tratar de aclarar una cuestión importante: por qué en numerosos trabajos de reconocimiento de expresión facial realizados con la base de datos CK se presentan resultados publicados de tasas de reconocimiento muy elevadas, cercanas o superiores a 94 % (algunos trabajos incluso con tasas superiores a 99 %, mientras que en nuestras versiones preliminares más sofisticadas la tasa de clasificación difícilmente superaba 91 %, incluso usando codificación similar a la propuesta en algunos de estos trabajos). Para ello, realizamos análisis dinámico de la base de datos CK, visualizando gráficas de puntajes de expresión contra tiempo, especialmente para muestras problemáticas clasificadas erróneamente. Los resultados mostraron que buena parte de las muestras clasificadas incorrectamente corresponden a secuencias que posteriormente habrían de ser eliminadas en la versión extendida CK+. Es decir, estas muestras no eran representativas de la clase etiquetada. En este capítulo mostramos algunos de estos casos, por cuanto no tenemos autorización de publicar más que algunas de las secuencias de la base de datos. En la mayor parte de los casos encontrados, incluyendo los mostrados en este capítulo, observamos que el error no es evidente cuando se evalúa la calidad y capacidad de representación de la muestra. Para evitar error de observación por bias, mostramos estas muestras defectuosas a algunos voluntarios independientes que ignoraban la etiqueta de la muestra y sus conclusiones son concordantes: las muestras no fueron clasificadas de acuerdo con su etiqueta por el panel independiente. Esto nos lleva a concluir que es complicado, incluso imposible, que un sistema de clasificación automático pueda clasificar correctamente muestras cuya etiqueta sea incorrecta, salvo que exista un error metodológico (tal como el entrenamiento y validación *random n-folded* que puede conducir a tasas increíbles de clasificación) o eliminación manual de las muestras defectuosas. En cualquiera de estos casos, sin embargo, la comparación directa entre estos resultados y los nuestros es imposible, debido a la cierta desventaja que tiene nuestro sistema al incluir todas las muestras, incluso las de dudosa calidad, en todo el proceso. Sin embargo, al evaluar la puntuación dinámica de las secuencias de la base de datos CK, en nuestra opinión y la opinión de evaluadores no sesgados [11]

[11] Contrastar los resultados de evaluación dinámica obtenidos por el sistema de clasificación con las etiquetas de expresión facial no fue un protocolo ciego para al autor, por cuanto ya conocía las etiquetas previamente. Para resolver este inconveniente se solicitó la ayuda de observadores no sesgados

buena parte de los errores de clasificación con esta base de datos pueden ser atribuidos a esta calidad dudosa de las muestras e incluso errores de etiqueta.

(es decir, que no conocían las etiquetas de las muestras previamente) para que evaluaran las muestras de clasificación errada, y efectivamente se pudo comprobar que en muchas de las muestras mostradas dinámicamente en la subsección 7.6.2 la expresión facial no corresponde a la evaluación humana

8. Conclusiones generales

En este capítulo se hará una reseña de las conclusiones generales de este trabajo. En la sección 8.1 se relatan los objetivos generales y específicos determinados en la propuesta de este trabajo, así como su cumplimiento. En la sección 8.2 se harán consideraciones finales del trabajo y perspectivas futuras.

8.1. Cumplimiento de objetivos

El objetivo general de este trabajo fue el desarrollo de un sistema de reconocimiento de expresión facial en secuencias de video con capacidad de clasificación de las 6 expresiones faciales definidas por Ekman, usando algoritmos basados en LBP. En este trabajo se implementaron los códigos VPOEM y TPOEM, derivados de la idea general de LBP, que probaron ser descriptores eficientes de la expresión facial, tanto por la alta capacidad de discriminación entre clases como por el bajo costo requerido para su cálculo. Nuestras pruebas mostraron cómo el sistema desarrollado es superior o al menos similar en resultados a los trabajos más prominentes del estado del arte. Así mismo, nuestras pruebas adicionales mostraron que el sistema tiene alta capacidad de generalización, puesto que los resultados de clasificación entre distintas bases de datos tienen alta tasa de reconocimiento. Por otra parte, se hizo comparación directa de clasificación realizada por humanos y clasificación realizada por el sistema automática, tarea que hasta nuestro conocimiento aún no ha sido desarrollada, y encontramos que el sistema automático tiene mejor capacidad de clasificación incluso en condiciones desventajosas.

El primer objetivo específico es la implementación de un algoritmo de corrección de iluminación con capacidad de ejecución en tiempo real, basándose en filtros que emulen el sistema visual humano. Esta tarea fue implementada usando cascadas de filtros Gaussianos y Gabor que simulan el funcionamiento del sistema de visión animal. Sin embargo, nuestras pruebas mostraron que los bancos en cascada atenúan pequeños micro gestos que son muy relevantes en la discriminación de la expresión facial. La atenuación de micro gestos para el reconocimiento facial no es un inconveniente y, de hecho, puede ser deseable, por cuanto los parámetros determinantes en la separación entre individuo son más globales. Es así que buena parte de los trabajos de reconocimiento facial usan este tipo de procesamiento. No obstante, para reconocimiento de la expresión facial esto ocasionó deterioro de la capacidad de clasificación. Debido a esto, se decició usar otra aproximación, con ecualización de histograma, transformación adaptativa truncada de histograma y transformación limitada por contraste. Los resultados no son visualmente similares a los obtenidos por filtrado que emule la corteza visual del cerebro mamífero y, de hecho, son en apariencia de calidad inferior. Sin embargo, en tanto que se acentúan las micro expresiones del rostro sin añadir notablemente ruido en la imagen, la salida es mucho más adecuada para la tarea de reconocimiento de expresión facial. Este tipo de filtrado no obtuvo los mejores resultados. Mediante ecualización de histograma y filtrado por técnicas isotrópicas de difusión se obtuvieron imágenes aún más satisfactorias,

pero el compromiso de tiempo de cálculo hizo que la primera aproximación fuese más recomendable.

El segundo objetivo específico es el desarrollo de un algoritmo basado en LBP para la obtención de parámetros faciales en las secuencias de video. Con este fin, en este trabajo se desarrollaron los códigos VPOEM y TPOEM como descriptores dinámicos basados en textura. La codificación VPOEM y TPOEM fue probada en secuencias de video estandarizadas y se cumplieron los objetivos tanto de capacidad de discriminación de la expresión facial como de requerimiento de bajo costo de cálculo. Así mismo, esta codificación tuvo resultados excelentes de reconocimiento de expresión facial con bases de datos disjuntas a las usadas en entrenamiento, lo que probó la alta capacidad de generalización.

Nuestro siguiente objetivo específico fue la evaluación de los distintos métodos usados para la reducción de dimensiones e implementación del más adecuado para la aplicación en tiempo real. Inicialmente esta tarea parecía ser más limitada, debido al estado de desarrollo teórico en el área de reducción de dimensiones. Sin embargo, en la práctica tuvimos que solucionar una suerte de retos imprevistos. En primera instancia, los datos de codificación VPOEM y TPOEM son de alta dimensión. Los algoritmos de estimación de dimensión intrínseca tanto supervisados como no supervisados mostraron un problema generalmente relatado en la literatura, consistente en la subestimación de la dimensión intrínseca real de los datos. Debido a este inconveniente, desarrollamos el algoritmo LC-NNMLE y lo comparamos con el algoritmo convencional NNMLE usando conjuntos de datos simulados de alta dimensión (espacios de dimensión real d embebidos en un espacio de dimensión mayor n), y encontramos que LC-NNMLE produjo una mejor estimación de la dimensión real. Posteriormente implementamos técnicas convencionales de reducción de dimensiones. Esto requiere de ciertos compromisos. El uso de reducción no supervisada de dimensiones permite disponer de todos los datos para las siguientes etapas del proceso. Sin embargo, la reducción no supervisada de dimensiones embebe los datos en un espacio de menor dimensión donde se preserve la mayor cantidad de información, pero la información relevante para la expresión facial generalmente constituye una pequeña fracción de la energía total, en comparación, por ejemplo, con la información global más adecuada para identificación facial. Debido a esto la reducción no supervisada de dimensiones produjo resultados de clasificación inferiores que con datos brutos sin procesar. En oposición, la reducción supervisada de dimensiones tiene la ventaja de que en principio no elimina información relevante para la discriminación entre clases. Al contrario, muchas técnicas de reducción supervisada intentan acentuar esta información. Sin embargo, los datos usados para la reducción supervisada de dimensiones no deben ser usados en etapas posteriores de validación del sistema, por cuanto esto constituye un error metodológico de prueba de hipótesis (aunque tanto en el problema de reconocimiento de expresión facial como en problemas de clasificación en general encontramos numerosos trabajos en los que se hace reducción supervisada de dimensiones y validación de clasificación con conjuntos no disjuntos). En consecuencia, el compromiso de la reducción supervisada de dimensiones es usar una cantidad suficiente de datos que permita describir el fenómeno pero a la vez de-

jando datos suficientes para el entrenamiento y la validación. Para ello la reducción supervisada de dimensiones mediante MCML mostró los mejores resultados, de manera que esta implementación fue la usada en el resto del trabajo. De manera adicional, implementamos los códigos SFA-WM para selección de parámetros. Si bien la selección de parámetros es un procedimiento ajeno a la reducción de dimensiones, seleccionar parámetros a partir de un conjunto total permite reducir el tamaño de los códigos usados para la clasificación, lo que a su vez reduce el costo de las etapas de extracción de parámetros, reducción de dimensiones y clasificación.

El tercer objetivo específico es el diseño y la construcción de un sistema de clasificación comparando sus resultados con los sistemas tradicionales por *template matching*, SVM y redes neuronales. En principio consideramos que esta etapa no habría de tener inconvenientes significativos, por cuanto buena parte de la bibliografía consultada previamente al planteamiento de la propuesta de este trabajo mostraba metodologías simples de clasificación con altas tasas de éxito para el problema de reconocimiento de la expresión facial. No obstante, si bien algunas metodologías limitadas de clasificación, tales como métricas por distancias Euclidianas o Mahalanobis obtuvieron resultados aceptables, al evaluar las curvas de aprendizaje de estas implementaciones observamos que en apariencia los códigos permitían mejor capacidad de discriminación que la obtenida usando estas metodologías. Es decir, las curvas de aprendizaje mostraban mayor descripción de las clases en cuanto se incrementaba la complejidad de los sistemas de clasificación, en vez de saturarse hacia la región de sobreajuste a partir de cierta frontera de complejidad. En consecuencia, en este trabajo se implementaron algoritmos de clasificación con complejidad mayor que la planteada inicialmente, incluyendo el desarrollo del algoritmo novedoso APCC, fusión ponderada de metaclasificadores SVM y FDA y clasificadores usando *deep learning*, con resultados notablemente superiores, incluyendo diferencias significativas t-test, que los obtenidos con metodologías convencionales.

8.2. Consideraciones finales y trabajo futuro

El reconocimiento de la expresión facial es una tarea de alta relevancia en la actualidad. En este trabajo realizamos aportes que en nuestra opinión son muy significativos para el estado de desarrollo teórico en este campo. En particular, demostramos que los sistemas automáticos están en capacidad de realizar reconocimiento de la expresión facial incluso mejor que la realizada por humanos, pese a que esto podría considerarse imposible algunos años atrás. No obstante, creemos que este trabajo tiene expectativas interesantes de desarrollo futuro que planteamos en esta sección.

En primer lugar, en la implementación de todos nuestros algoritmos estuvo la limitación determinada por los requerimientos de tiempo real. Si bien los algoritmos usados cumplieron satisfactoriamente estos requisitos, creemos que hay alternativas viables que pueden reducir notablemente los costos de cálculo o permitir el uso de descriptores o sistemas de clasificación más complejos. Por ejemplo, los descriptores VPOEM y TPOEM son basados en celdas espaciales, de manera que es posible considerar el

uso de arquitecturas más propicias para este tipo de procesamiento, tales como CUDA o procesamiento paralelo. De esta forma el procedimiento completo tendría reducción notable en tiempo de cálculo incluso en dispositivos de procesamiento limitado.

En nuestro trabajo probamos los algoritmos de clasificación basados en *deep learning*. Estas pruebas, sin embargo, fueron de carácter limitado, debido a que fueron realizadas con LSO, dejando una muestra por fuera de cada entrenamiento. Se hizo así debido a que el entrenamiento por *deep learning* requiere del mayor número de datos posibles para describir el fenómeno, pero el entrenamiento de cada cascada de autoencoders apilados requería de un alto costo de cálculo. Es así que si bien la validación no es significativamente más lenta que la validación por nuestras otras metodologías usadas, el proceso de entrenamiento fue dispendioso y limitó el número de pruebas. Pese a ello, los resultados mostraron tasas de reconocimiento ligeramente superiores. Las pruebas t-test no mostraron diferencia estadística, pero se debe señalar que en las pruebas t-test se usaron los resultados globales de clasificación, no los resultados individuales por expresión. Con los resultados individuales por expresión se esperaría que los márgenes de diferencia estadística entre pruebas sean más pequeños, pero no hicimos esto por cuanto la clasificación de cada expresión no es independiente, y no hubo consenso entre los expertos consultados acerca de cómo hacer pruebas t-test de validación de clasificación con múltiples clasificaciones que no son completamente independientes. Sin embargo, la relevancia de nuestra implementación con *deep learning* es que muestra sin duda que los resultados no son inferiores y son potencialmente superiores que con otras metodologías de clasificación, incluyendo nuestra clasificación compleja por fusión de FDA y SVM con APCC.

El principal inconveniente del uso de bases de datos estandarizadas es que en general las secuencias son frontales y en condiciones controladas, lo que dificulta la evaluación en circunstancias espontáneas y de videos naturales. La manera directa de solucionar esto es usar secuencias de video que no pertenezcan a bases de datos. Esto, sin embargo, genera otra suerte de problemas. Validar expresiones faciales en imágenes o en secuencias de video no es un problema trivial. Al contrario, tal como relata la literatura y como mostraron nuestras pruebas, la clasificación de la expresión facial realizada por humanos tiene relativa baja tasa de acierto y alta variabilidad entre individuos, de manera que obtener *ground truth* es un problema importante. Sin embargo, a partir de pruebas preliminares usando secuencias de video recopiladas por el autor, con rostros no necesariamente frontales, se obtuvieron resultados promisorios. Para ello se hizo alineación y corrección de perspectiva del rostro mediante deformación espacial a partir de la localización de los ojos y la boca. Con los rostros alineados se hizo reconocimiento de expresión usando sistemas previamente entrenados y los resultados fueron satisfactorios, con alta clasificación de las expresiones alegría, sorpresa, ira y disgusto. Tristeza y miedo fueron problemáticas, pero esto obedece tanto a las limitaciones del sistema como a la dificultad de ejecución espontánea de estas expresiones, lo cual es resaltado en la representación de estas expresiones en la base de datos CK+, con un número muy inferior de muestras que las muestras de alegría o sorpresa.

De manera paralela a este trabajo realizamos pruebas en condiciones semicontrola-

das con datos recopilados propios de secuencias no frontales y secuencias con artefactos tales como aretes, gafas y barba. Mediante algoritmos de alineación espacial fue posible obtener resultados aceptables incluso en estas condiciones salvo con individuos con barba, debido a la textura generada por la misma. No obstante, así como genera dificultades para reconocimiento automático, también lo hace de manera similar para evaluación humana, de modo que creemos que es un obstáculo irresoluble. Otros tipos de artefactos tales como piercings o tatuajes no fueron evaluados, así que es una interesante prueba que debería realizarse en el futuro.

Bibliografía

[1] Abboud, B., Davoine, F., and Dang, M. Facial expression recognition and synthesis based on an appearance model. *Signal Processing: Image Communication 19*, 8 (2004), 723–740.

[2] Ahlberg, J. Candide-3-an updated parameterised face. Tech. rep., Linkping University, 2001.

[3] Ahonen, T., Hadid, A., and Pietikäinen, M. Face recognition with local binary patterns. In *Computer vision-eccv 2004*. Springer, 2004, pp. 469–481.

[4] Almaev, T. R., and Valstar, M. F. Local gabor binary patterns from three orthogonal planes for automatic facial expression recognition. In *Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on* (2013), IEEE, pp. 356–361.

[5] Arlot, S., Celisse, A., et al. A survey of cross-validation procedures for model selection. *Statistics surveys 4* (2010), 40–79.

[6] Bartlett, M. S., Littlewort, G., Fasel, I., and Movellan, J. R. Real time face detection and facial expression recognition: Development and applications to human computer interaction. In *Computer Vision and Pattern Recognition Workshop, 2003. CVPRW'03. Conference on* (2003), vol. 5, IEEE, pp. 53–53.

[7] Bartlett, M. S., Littlewort, G., Frank, M., Lainscsek, C., Fasel, I., and Movellan, J. Recognizing facial expression: machine learning and application to spontaneous behavior. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on* (2005), vol. 2, IEEE, pp. 568–573.

[8] Belhumeur, P. N., Hespanha, J. P., and Kriegman, D. J. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. In *Computer Vision—ECCV'96*. Springer, 1996, pp. 43–58.

[9] Bellman, R. Dynamic programming and lagrange multipliers. *Proceedings of the National Academy of Sciences of the United States of America 42*, 10 (1956), 767.

[10] Bengio, Y. Learning deep architectures for ai. *Foundations and trends® in Machine Learning 2*, 1 (2009), 1–127.

[11] Bengio, Y., Courville, A., and Vincent, P. Representation learning: A review and new perspectives. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 35*, 8 (2013), 1798–1828.

[12] Bengio, Y., and LeCun, Y. Scaling learning algorithms towards ai. *Large-scale kernel machines 34* (2007), 1–41.

[13] Beyer, K., Goldstein, J., Ramakrishnan, R., and Shaft, U. When is nearest neighbor meaningful? In *Database TheoryICDT99*. Springer, 1999, pp. 217–235.

[14] Bourel, F., Chibelushi, C. C., and Low, A. A. Robust facial expression recognition using a state-based model of spatially-localised facial dynamics. In *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on* (2002), IEEE, pp. 106–111.

[15] Boyle, E. A., Anderson, A. H., and Newlands, A. The effects of visibility on dialogue and performance in a cooperative problem solving task. *Language and speech 37*, 1 (1994), 1–20.

[16] Breiman, L. Random forests. *Machine learning 45*, 1 (2001), 5–32.

[17] Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. *Classification and regression trees*. CRC press, 1984.

[18] Bruce, V. What the human face tells the human mind: Some challenges for the robot-human interface. *Advanced Robotics 8*, 4 (1993), 341–355.

[19] Burgoon, J. K., Saine, T., et al. *The unspoken dialogue: An introduction to nonverbal communication*. Houghton Mifflin Boston, 1978.

[20] Busso, C., Deng, Z., Yildirim, S., Bulut, M., Lee, C. M., Kazemzadeh, A., Lee, S., Neumann, U., and Narayanan, S. Analysis of emotion recognition using facial expressions, speech and multimodal information. In *Proceedings of the 6th international conference on Multimodal interfaces* (2004), ACM, pp. 205–211.

[21] Calder, A. J., Burton, A. M., Miller, P., Young, A. W., and Akamatsu, S. A principal component analysis of facial expressions. *Vision research 41*, 9 (2001), 1179–1208.

[22] Calder, A. J., Young, A. W., Perrett, D. I., Etcoff, N. L., and Rowland, D. Categorical perception of morphed facial expressions. *Visual Cognition 3*, 2 (1996), 81–118.

[23] Castrillón, M., Déniz, O., Hernández, D., and Lorenzo, J. A comparison of face and facial feature detectors based on the viola–jones general object detection framework. *Machine Vision and Applications 22*, 3 (2011), 481–494.

[24] Cherkassky, V., and Ma, Y. Selection of meta-parameters for support vector regression. In *Artificial Neural NetworksICANN 2002*. Springer, 2002, pp. 687–693.

[25] Choi, H.-C., and Oh, S.-Y. Realtime facial expression recognition using active appearance model and multilayer perceptron. In *SICE-ICASE, 2006. International Joint Conference* (2006), IEEE, pp. 5924–5927.

[26] Cohen, I., Sebe, N., Garg, A., Chen, L. S., and Huang, T. S. Facial expression recognition from video sequences: temporal and static modeling. *Computer Vision and Image Understanding 91*, 1 (2003), 160–187.

[27] Coifman, R. R., Lafon, S., Lee, A. B., Maggioni, M., Nadler, B., Warner, F., and Zucker, S. W. Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps. *Proceedings of the National Academy of Sciences of the United States of America 102*, 21 (2005), 7426–7431.

[28] Comon, P. Independent component analysis, a new concept? *Signal processing 36*, 3 (1994), 287–314.

[29] Copas, J. B. Regression, prediction and shrinkage. *Journal of the Royal Statistical Society. Series B (Methodological)* (1983), 311–354.

[30] Cowie, R., Douglas-Cowie, E., Tsapatsoulis, N., Votsis, G., Kollias, S., Fellenz, W., and Taylor, J. G. Emotion recognition in human-computer interaction. *Signal Processing Magazine, IEEE 18*, 1 (2001), 32–80.

[31] Craw, I., Tock, D., and Bennett, A. Finding face features. In *Computer Vision—ECCV'92* (1992), Springer, pp. 92–96.

[32] Curran, K., Li, X., and McCaughley, N. Neural network face detection. *The Imaging Science Journal 53*, 2 (2005), 105–115.

[33] Darwin, C. *The expression of the emotions in man and animals*. Oxford University Press, 1998.

[34] de Gelder, B., Teunisse, J.-P., and Benson, P. J. Categorical perception of facial expressions: Categories and their internal structure. *Cognition & Emotion 11*, 1 (1997), 1–23.

[35] Deng, S., Xu, Y., Li, L., Li, X., and He, Y. A feature-selection algorithm based on support vector machine-multiclass for hyperspectral visible spectral analysis. *Journal of Food Engineering 119*, 1 (2013), 159–166.

[36] Denil, M., and Trappenberg, T. Overlap versus imbalance. In *Advances in Artificial Intelligence*. Springer, 2010, pp. 220–231.

[37] Dijkstra, E. W. A note on two problems in connexion with graphs. *Numerische mathematik 1*, 1 (1959), 269–271.

[38] Domingos, P., and Pazzani, M. On the optimality of the simple bayesian classifier under zero-one loss. *Machine learning 29*, 2-3 (1997), 103–130.

[39] Donato, G., Bartlett, M. S., Hager, J. C., Ekman, P., and Sejnowski, T. J. Classifying facial actions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 21*, 10 (1999), 974–989.

[40] Donoho, D. L., et al. High-dimensional data analysis: The curses and blessings of dimensionality. *AMS Math Challenges Lecture* (2000), 1–32.

[41] Douglas, R., and Sejnowski, T. Future challenges for the science and engineering of learning. Tech. rep., National Science Foundation, July 2007.

[42] Edwards, G. J., Taylor, C. J., and Cootes, T. F. Interpreting face images using active appearance models. In *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on* (1998), IEEE, pp. 300–305.

[43] Efroymson, M. Multiple regression analysis. *Mathematical methods for digital computers 1* (1960), 191–203.

[44] Ekman, P. Facial expression and emotion. *American psychologist 48*, 4 (1993), 384.

[45] Ekman, P., and Friesen, W. V. Constants across cultures in the face and emotion. *Journal of personality and social psychology 17*, 2 (1971), 124.

[46] Ekman, P., Friesen, W. V., O'Sullivan, M., Chan, A., Diacoyanni-Tarlatzis, I., Heider, K., Krause, R., LeCompte, W. A., Pitcairn, T., Ricci-Bitti, P. E., et al. Universals and cultural differences in the judgments of facial expressions of emotion. *Journal of personality and social psychology 53*, 4 (1987), 712.

[47] Fan, J., and Li, R. Statistical challenges with high dimensionality: Feature selection in knowledge discovery. *arXiv preprint math/0602133* (2006).

[48] Feichtinger, H. G., and Strohmer, T. *Gabor analysis and algorithms: Theory and applications*. Springer, 1998.

[49] Feng, X., Pietikainen, M., and Hadid, A. Facial expression recognition with local binary patterns and linear programming. *Pattern Recognition And Image Analysis C/C of Raspoznavaniye Obrazov I Analiz Izobrazhenii 15*, 2 (2005), 546.

[50] Fisher, R. A. The use of multiple measurements in taxonomic problems. *Annals of eugenics 7*, 2 (1936), 179–188.

[51] Fridlund, A. J. *Human facial expression: An evolutionary view.* Academic Press, 1994.

[52] Georghiades, A., Belhumeur, P., and Kriegman, D. From few to many: Illumination cone models for face recognition under variable lighting and pose. *IEEE Trans. Pattern Anal. Mach. Intelligence 23*, 6 (2001), 643–660.

[53] Ghimire, D., and Lee, J. Geometric feature-based facial expression recognition in image sequences using multi-class adaboost and support vector machines. *Sensors 13*, 6 (2013), 7714–7734.

[54] Globerson, A., and Roweis, S. T. Metric learning by collapsing classes. In *Advances in neural information processing systems* (2005), pp. 451–458.

[55] Gokcen, I., and Peng, J. Comparing linear discriminant analysis and support vector machines. In *Advances in Information Systems*. Springer, 2002, pp. 104–113.

[56] Gu, Q., Li, Z., and Han, J. Generalized fisher score for feature selection. *arXiv preprint arXiv:1202.3725* (2012).

[57] Gu, W., Xiang, C., Venkatesh, Y., Huang, D., and Lin, H. Facial expression recognition using radial encoding of local gabor features and classifier synthesis. *Pattern Recognition 45*, 1 (2012), 80–91.

[58] Guyon, I., and Elisseeff, A. An introduction to variable and feature selection. *The Journal of Machine Learning Research 3* (2003), 1157–1182.

[59] Guyon, I., Gunn, S., Nikravesh, M., and Zadeh, L. Feature extraction. *Foundations and applications* (2006), 1–25.

[60] Hampson, E., van Anders, S. M., and Mullin, L. I. A female advantage in the recognition of emotional facial expressions: Test of an evolutionary hypothesis. *Evolution and Human Behavior 27*, 6 (2006), 401–416.

[61] Han, H., Shan, S., Chen, X., and Gao, W. A comparative study on illumination preprocessing in face recognition. *Pattern Recognition 46*, 6 (2013), 1691–1699.

[62] Hanczar, B., Courtine, M., Benis, A., Hennegar, C., Clément, K., and Zucker, J.-D. Improving classification of microarray data using prototype-based feature selection. *ACM SIGKDD Explorations Newsletter 5*, 2 (2003), 23–30.

[63] Hanczar, B., Hua, J., Sima, C., Weinstein, J., Bittner, M., and Dougherty, E. Small-sample precision of roc-related estimates. *Bioinformatics 26*, 6 (2010), 822–830.

[64] Hand, D. J., and Yu, K. Idiot's bayesnot so stupid after all? *International statistical review 69*, 3 (2001), 385–398.

[65] Haxby, J. V., Hoffman, E. A., and Gobbini, M. I. The distributed human neural system for face perception. *Trends in cognitive sciences 4*, 6 (2000), 223–233.

[66] Hazewinkel, M. Maximum-likelihood method. In *Encyclopedia of Mathematic*, J. Fagerberg, D. Mowery, and R. Nelson, Eds. Springer, Oxford, 2001.

[67] Hegstrom, T. G. Message impact: What percentage is nonverbal? *Western Journal of Communication (includes Communication Reports) 43*, 2 (1979), 134–142.

[68] Heikkilä, M., Pietikäinen, M., and Schmid, C. Description of interest regions with local binary patterns. *Pattern recognition 42*, 3 (2009), 425–436.

[69] Heusch, G., Cardinaux, F., and Marcel, S. Lighting normalization algorithms for face verification. *IDIAP-com 05 3* (2005).

[70] Hinton, G., Osindero, S., and Teh, Y.-W. A fast learning algorithm for deep belief nets. *Neural computation 18*, 7 (2006), 1527–1554.

[71] Hoffmann, H., Kessler, H., Eppel, T., Rukavina, S., and Traue, H. C. Expression intensity, gender and facial emotion recognition: Women recognize only subtle facial emotions better than men. *Acta psychologica 135*, 3 (2010), 278–283.

[72] Huffman, D. A., et al. A method for the construction of minimum redundancy codes. *proc. IRE 40*, 9 (1952), 1098–1101.

[73] Hughes, G. On the mean accuracy of statistical pattern recognizers. *Information Theory, IEEE Transactions on 14*, 1 (1968), 55–63.

[74] Hyvärinen, A., and Oja, E. Independent component analysis: algorithms and applications. *Neural networks 13*, 4 (2000), 411–430.

[75] Jaccard coefficient: Jaccard, P. *Étude comparative de la distribution florale dans une portion des Alpes et des Jura*, vol. 37. Impr. Corbaz, 1901.

[76] Jafri, R., and Arabnia, H. R. A survey of face recognition techniques. *JIPS 5*, 2 (2009), 41–68.

[77] Jin, J. Impossibility of successful classification when useful features are rare and weak. *Proceedings of the National Academy of Sciences 106*, 22 (2009), 8859–8864.

[78] John, G. H., and Langley, P. Estimating continuous distributions in bayesian classifiers. In *Proceedings of the Eleventh conference on Uncertainty in artificial intelligence* (1995), Morgan Kaufmann Publishers Inc., pp. 338–345.

[79] Jolliffe, I. *Principal component analysis*. Wiley Online Library, 2005.

[80] Juang, B.-H., and Katagiri, S. Discriminative learning for minimum error classification [pattern recognition]. *Signal Processing, IEEE Transactions on 40*, 12 (1992), 3043–3054.

[81] Kecman, V. *Learning and soft computing: support vector machines, neural networks, and fuzzy logic models*. MIT press, 2001.

[82] Khan, R. A., Meyer, A., Konik, H., and Bouakaz, S. Human vision inspired framework for facial expressions recognition. In *Image Processing (ICIP), 2012 19th IEEE International Conference on* (2012), IEEE, pp. 2593–2596.

[83] Kohavi, R., and John, G. H. Wrappers for feature subset selection. *Artificial intelligence 97*, 1 (1997), 273–324.

[84] Kotsia, I., and Pitas, I. Facial expression recognition in image sequences using geometric deformation features and support vector machines. *Image Processing, IEEE Transactions on 16*, 1 (2007), 172–187.

[85] Kotsia, I., Zafeiriou, S., and Pitas, I. Texture and shape information fusion for facial expression and facial action unit recognition. *Pattern Recognition 41*, 3 (2008), 833–851.

[86] Kullback, S., and Leibler, R. A. On information and sufficiency. *The Annals of Mathematical Statistics* (1951), 79–86.

[87] Lal, T. N., Chapelle, O., Weston, J., and Elisseeff, A. Embedded methods. In *Feature extraction*. Springer, 2006, pp. 137–165.

[88] Lapakko, D. Three cheers for language: A closer examination of a widely cited study of nonverbal communication. *Communication Education 46*, 1 (1997), 63–67.

[89] Leszczyński, M. Image preprocessing for illumination invariant face verification. *Journal of Telecommunications and Information Technology* (2010), 19–25.

[90] Levina, E., and Bickel, P. J. Maximum likelihood estimation of intrinsic dimension. In *Advances in neural information processing systems* (2004), pp. 777–784.

[91] Lippert, E. High-dimensional spaces are counterintuitive, part two, May 2005.

[92] Littlewort, G., Bartlett, M. S., Fasel, I., Susskind, J., and Movellan, J. Dynamics of facial expression extracted automatically from video. *Image and Vision Computing 24*, 6 (2006), 615–625.

[93] Liu, C. H., Chen, W., Han, H., and Shan, S. Effects of image preprocessing on face matching and recognition in human observers. *Applied Cognitive Psychology 27*, 6 (2013), 718–724.

[94] Liu, Y. A comparative study on feature selection methods for drug discovery. *Journal of chemical information and computer sciences 44*, 5 (2004), 1823–1828.

[95] Long, F., Wu, T., Movellan, J. R., Bartlett, M. S., and Littlewort, G. Learning spatiotemporal features by using independent component analysis with application to facial expression recognition. *Neurocomputing 93* (2012), 126–132.

[96] Long, P. M., and Servedio, R. A. Random classification noise defeats all convex potential boosters. *Machine Learning 78*, 3 (2010), 287–304.

[97] Lowe, D. G. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on* (1999), vol. 2, Ieee, pp. 1150–1157.

[98] Lucey, P., Cohn, J. F., Kanade, T., Saragih, J., Ambadar, Z., and Matthews, I. The extended cohn-kanade dataset (ck+): A complete dataset for action unit and emotion-specified expression. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on* (2010), IEEE, pp. 94–101.

[99] Lundqvist, D., Flykt, A., and Öhman, A. The karolinska directed emotional faces (kdef). *CD ROM from Department of Clinical Neuroscience, Psychology section, Karolinska Institutet* (1998), 91–630.

[100] Lutu, P. E., and Engelbrecht, A. P. A decision rule-based method for feature selection in predictive data mining. *Expert Systems with Applications 37*, 1 (2010), 602–609.

[101] Lyons, M., Akamatsu, S., Kamachi, M., and Gyoba, J. Coding facial expressions with gabor wavelets. In *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on* (1998), IEEE, pp. 200–205.

[102] Maron, M. E. Automatic indexing: an experimental inquiry. *Journal of the ACM (JACM) 8*, 3 (1961), 404–417.

[103] Martinez, A., and Du, S. A model of the perception of facial expressions of emotion by humans: Research overview and perspectives. *The Journal of Machine Learning Research 13*, 1 (2012), 1589–1608.

[104] Matsumoto, D., LeRoux, J., Wilson-Cohn, C., Raroque, J., Kooken, K., Ekman, P., Yrizarry, N., Loewinger, S., Uchida, H., Yee, A., et al. A new test to measure emotion recognition ability: Matsumoto and ekman's japanese and caucasian brief affect recognition test (jacbart). *Journal of Nonverbal Behavior 24*, 3 (2000), 179–209.

[105] Mattera, D., and Haykin, S. Support vector machines for dynamic reconstruction of a chaotic system. In *Advances in kernel methods* (1999), MIT Press, pp. 211–241.

[106] McCullagh, P., and Nelder, J. A. Generalized linear models. *Monographs on Statistics and Applied Probability* (1989).

[107] McLachlan, G. *Discriminant analysis and statistical pattern recognition*, vol. 544. John Wiley & Sons, 2004.

[108] Mehrabian, A. *Nonverbal communication*. Transaction Publishers, 1977.

[109] Mehrabian, A. *Intercultural encounters: The fundamentals of intercultural communication*, 3rd edition ed. Morton Publishing Company, 1995.

[110] Mehrabian, A., and Ferris, S. R. Inference of attitudes from nonverbal communication in two channels. *Journal of consulting psychology 31*, 3 (1967), 248.

[111] Meng, J., and Yang, Y. Symmetrical two-dimensional pca with image measures in face recognition. *Int J Adv Robotic Sy 9*, 238 (2012).

[112] Mita, T., Kaneko, T., and Hori, O. Joint haar-like features for face detection. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on* (2005), vol. 2, IEEE, pp. 1619–1626.

[113] Mohammed, A. A., Minhas, R., Jonathan Wu, Q., and Sid-Ahmed, M. A. Human face recognition based on multidimensional pca and extreme learning machine. *Pattern Recognition 44*, 10 (2011), 2588–2597.

[114] Moore, S., and Bowden, R. Local binary patterns for multi-view facial expression recognition. *Computer Vision and Image Understanding 115*, 4 (2011), 541–558.

[115] Murthy, G., and Jadon, R. Effectiveness of eigenspaces for facial expressions recognition. *International Journal of Computer Theory and Engineering 1*, 5 (2009), 1793–8201.

[116] Naab, P. J., and Russell, J. A. Judgments of emotion from spontaneous facial expressions of new guineans. *Emotion 7*, 4 (2007), 736.

[117] Nakariyakul, S., and Casasent, D. P. An improvement on floating search algorithms for feature subset selection. *Pattern Recognition 42*, 9 (2009), 1932–1940.

[118] Ojala, T., and Pietikäinen, M. Unsupervised texture segmentation using feature distributions. *Pattern Recognition 32*, 3 (1999), 477–486.

[119] Ojala, T., Pietikainen, M., and Harwood, D. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. In *Pattern Recognition, 1994. Vol. 1-Conference A: Computer Vision & Image Processing., Proceedings of the 12th IAPR International Conference on* (1994), vol. 1, IEEE, pp. 582–585.

[120] Ojala, T., Pietikäinen, M., and Harwood, D. A comparative study of texture measures with classification based on featured distributions. *Pattern recognition 29*, 1 (1996), 51–59.

[121] Osuna, E., Freund, R., and Girosi, F. Training support vector machines: an application to face detection. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on* (1997), IEEE, pp. 130–136.

[122] Owusu, E., Zhan, Y., and Mao, Q. R. A neural-adaboost based facial expression recognition system. *Expert Systems with Applications 41*, 7 (2014), 3383–3390.

[123] Pantic, M., and Rothkrantz, L. J. An expert system for multiple emotional classification of facial expressions. In *Tools with Artificial Intelligence, 1999. Proceedings. 11th IEEE International Conference on* (1999), IEEE, pp. 113–120.

[124] Pantic, M., Valstar, M., Rademaker, R., and Maat, L. Web-based database for facial expression analysis. In *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on* (2005), IEEE, pp. 5–pp.

[125] Pearson, K. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science 2*, 11 (1901), 559–572.

[126] Penev, P. S., and Atick, J. J. Local feature analysis: A general statistical theory for object representation. *Network: computation in neural systems 7*, 3 (1996), 477–500.

[127] Pettis, K. W., Bailey, T. A., Jain, A. K., and Dubes, R. C. An intrinsic dimensionality estimator from near-neighbor information. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 1*, 1 (1979), 25–37.

[128] Phillips, M. L., Young, A. W., Senior, C., Brammer, M., Andrew, C., Calder, A. J., Bullmore, E. T., Perrett, D., Rowland, D., Williams, S., et al. A specific neural substrate for perceiving facial expressions of disgust. *Nature 389*, 6650 (1997), 495–498.

[129] Pizer, S. M., Amburn, E. P., Austin, J. D., Cromartie, R., Geselowitz, A., Greer, T., ter Haar Romeny, B., Zimmerman, J. B., and Zuiderveld, K. Adaptive histogram equalization and its variations. *Computer vision, graphics, and image processing 39*, 3 (1987), 355–368.

[130] Platt, J. C. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in large margin classifiers* (1999), Citeseer.

[131] Poldrack, R. The perils of leave-one-out crossvalidation for individual difference analyses, December 2012.

[132] Prati, R. C., Batista, G. E., and Monard, M. C. Class imbalances versus class overlapping: an analysis of a learning system behavior. In *MICAI 2004: Advances in Artificial Intelligence*. Springer, 2004, pp. 312–321.

[133] Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P. *Numerical recipes in C*, vol. 2. Citeseer, 1996.

[134] Pudil, P., Novovičová, J., and Kittler, J. Floating search methods in feature selection. *Pattern recognition letters 15*, 11 (1994), 1119–1125.

[135] Pudil, P., Novovičová, J., and Kittler, J. Floating search methods in feature selection. *Pattern recognition letters 15*, 11 (1994), 1119—-1125.

[136] Rencher, A. C., and Pun, F. C. Inflation of r2 in best subset regression. *Technometrics 22*, 1 (1980), 49–53.

[137] Roweis, S. T., and Saul, L. K. Nonlinear dimensionality reduction by locally linear embedding. *Science 290*, 5500 (2000), 2323–2326.

[138] Rowley, H. A., Baluja, S., and Kanade, T. Neural network-based face detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 20*, 1 (1998), 23–38.

[139] Ruan, L., Yuan, M., and Zou, H. Regularized parameter estimation in high-dimensional gaussian mixture models. *Neural computation 23*, 6 (2011), 1605–1622.

[140] Ruiz-del Solar, J., and Quinteros, J. Illumination compensation and normalization in eigenspace-based face recognition: A comparative study of different pre-processing approaches. *Pattern Recognition Letters 29*, 14 (2008), 1966–1979.

[141] Safayani, M., Manzuri Shalmani, M. T., and Khademi, M. Extended two-dimensional pca for efficient face representation and recognition. In *Intelligent Computer Communication and Processing, 2008. ICCP 2008. 4th International Conference on* (2008), IEEE, pp. 295–298.

[142] Saha, A., and Wu, Q. J. Facial expression recognition using curvelet based local binary patterns. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on* (2010), IEEE, pp. 2470–2473.

[143] Salah, A. A., and Alpaydin, E. Incremental mixtures of factor analysers. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on* (2004), vol. 1, IEEE, pp. 276–279.

[144] Salakhutdinov, R., Tenenbaum, J. B., and Torralba, A. Learning with hierarchical-deep models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 35*, 8 (2013), 1958–1971.

[145] Scharf, L. L. Minimum mean-squared error estimators. In *Statistical signal processing: detection, estimation, and time series analysis*, vol. 98. Addison-Wesley Reading, MA, 1991, ch. 8, pp. 326–329.

[146] Schölkopf, B., Smola, A., and Müller, K.-R. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation 10*, 5 (1998), 1299–1319.

[147] Scholkopft, B., and Mullert, K.-R. Fisher discriminant analysis with kernels. *Neural networks for signal processing IX* (1999).

[148] Sebe, N., Lew, M. S., Sun, Y., Cohen, I., Gevers, T., and Huang, T. S. Authentic facial expression analysis. *Image and Vision Computing 25*, 12 (2007), 1856–1863.

[149] Shalizi, C. Non-linear dimensionality reduction ii: Diffusion maps, 2009.

[150] Shan, C., Gong, S., and McOwan, P. W. Robust facial expression recognition using local binary patterns. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on* (2005), vol. 2, IEEE, pp. II–370.

[151] Shan, C., Gong, S., and McOwan, P. W. Facial expression recognition based on local binary patterns: A comprehensive study. *Image and Vision Computing 27*, 6 (2009), 803–816.

[152] Shan, S., Yang, P., Chen, X., and Gao, W. Adaboost gabor fisher classifier for face recognition. In *Analysis and Modelling of Faces and Gestures*. Springer, 2005, pp. 279–292.

[153] Somol, P., Pudil, P., Novovicová, J., and Paclik, P. Adaptive floating search methods in feature selection. *Pattern Recognigion Letters 20*, 11 (1999), 1157—-1163.

[154] Song, Q., Ni, J., and Wang, G. A fast clustering-based feature subset selection algorithm for high-dimensional data. *Knowledge and Data Engineering, IEEE Transactions on 25*, 1 (2013), 1–14.

[155] Sun, D., and Zhang, D. Bagging constraint score for feature selection with pairwise constraints. *Pattern Recognition 43*, 6 (2010), 2106–2118.

[156] Taner Eskil, M., and Benli, K. S. Facial expression recognition based on anatomy. *Computer Vision and Image Understanding 119* (2014), 1–14.

[157] Tenenbaum, J. B., De Silva, V., and Langford, J. C. A global geometric framework for nonlinear dimensionality reduction. *Science 290*, 5500 (2000), 2319–2323.

[158] Thayer, S. The effect of expression sequence and expressor identity on judgments of the intensity of facial expression. *Journal of Nonverbal Behavior 5*, 2 (1980), 71–79.

[159] Tian, Y.-l. Evaluation of face resolution for expression analysis. In *Computer Vision and Pattern Recognition Workshop, 2004. CVPRW'04. Conference on* (2004), IEEE, pp. 82–82.

[160] Trausan-Matu, S., Boyer, K., Crosby, M., and Panourgia, K. *Intelligent Tutoring Systems: 12th International Conference, ITS 2014, Honolulu, HI, USA, June 5-9, 2014. Proceedings*, vol. 8474. Springer, 2014.

[161] Turk, M. A., and Pentland, A. P. Face recognition using eigenfaces. In *Computer Vision and Pattern Recognition, 1991. Proceedings CVPR'91., IEEE Computer Society Conference on* (1991), IEEE, pp. 586–591.

[162] Uchida, S., and Sakoe, H. A survey of elastic matching techniques for handwritten character recognition. *IEICE transactions on information and systems 88*, 8 (2005), 1781–1790.

[163] Universite Montreal, M. L. L. Udem machine learning lab (lisa), August 2008.

[164] Valstar, M. F., and Pantic, M. Fully automatic recognition of the temporal phases of facial actions. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 42*, 1 (2012), 28–43.

[165] van der Maaten, L. J., Postma, E. O., and van den Herik, H. J. Dimensionality reduction: A comparative review. *Journal of Machine Learning Research 10*, 1-41 (2009), 66–71.

[166] Vapnik, V. *The nature of statistical learning theory*. springer, 2000.

[167] Viola, P., and Jones, M. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on* (2001), vol. 1, IEEE, pp. I–511.

[168] Vu, N.-S., and Caplier, A. Illumination-robust face recognition using retina modeling. In *Image Processing (ICIP), 2009 16th IEEE International Conference on* (2009), IEEE, pp. 3289–3292.

[169] Vu, N.-S., and Caplier, A. Face recognition with patterns of oriented edge magnitudes. In *Computer Vision–ECCV 2010*. Springer, 2010, pp. 313–326.

[170] Vu, N.-S., Dee, H. M., and Caplier, A. Face recognition using the poem descriptor. *Pattern Recognition 45*, 7 (2012), 2478–2488.

[171] Wagner, H. L. The accessibility of the term contempt and the meaning of the unilateral lip curl. *Cognition & Emotion 14*, 5 (2000), 689–710.

[172] Wang, H., Li, S. Z., and Wang, Y. Face recognition under varying lighting conditions using self quotient image. In *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on* (2004), IEEE, pp. 819–824.

[173] Wang, H., Li, S. Z., Wang, Y., and Zhang, J. Self quotient image for face recognition. In *Image Processing, 2004. ICIP'04. 2004 International Conference on* (2004), vol. 2, IEEE, pp. 1397–1400.

[174] Wang, J., and Yin, L. Static topographic modeling for facial expression recognition and analysis. *Computer Vision and Image Understanding 108*, 1 (2007), 19–34.

[175] Wang, X., Han, T. X., and Yan, S. An hog-lbp human detector with partial occlusion handling. In *Computer Vision, 2009 IEEE 12th International Conference on* (2009), IEEE, pp. 32–39.

[176] Watts, D. J., and Strogatz, S. H. Collective dynamics of small-worldnetworks. *nature 393*, 6684 (1998), 440–442.

[177] Waxman, S. R. Linguistic biases and the establishment of conceptual hierarchies: Evidence from preschool children. *Cognitive Development 5*, 2 (1990), 123–150.

[178] Webb, A. R. *Statistical pattern recognition*. John Wiley & Sons, 2003.

[179] Weickert, J. *Anisotropic diffusion in image processing*, vol. 1. Teubner Stuttgart, 1998.

[180] Wolpert, D. H. The existence of a priori distinctions between learning algorithms. *Neural Computation 8*, 7 (1996), 1391–1420.

[181] Wolpert, D. H. The lack of a priori distinctions between learning algorithms. *Neural computation 8*, 7 (1996), 1341–1390.

[182] Yang, M.-H., Kriegman, D., and Ahuja, N. Detecting faces in images: A survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 24*, 1 (2002), 34–58.

[183] Yang, P., Liu, Q., and Metaxas, D. N. Boosting encoded dynamic features for facial expression recognition. *Pattern Recognition Letters 30*, 2 (2009), 132–139.

[184] Yang, Q., and Ding, X. Symmetrical pca in face recognition. In *Image Processing. 2002. Proceedings. 2002 International Conference on* (2002), vol. 2, IEEE, pp. II–97.

[185] Yu, L., and Liu, H. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *ICML* (2003), vol. 3, pp. 856–863.

[186] Yu, L., and Liu, H. Redundancy based feature selection for microarray data. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining* (2004), ACM, pp. 737–742.

[187] Zhang, C., and Zhang, Z. A survey of recent advances in face detection. Tech. rep., Tech. rep., Microsoft Research, 2010.

[188] Zhang, D., and Zhou, Z.-H. (2d) 2pca: Two-directional two-dimensional pca for efficient face representation and recognition. *Neurocomputing 69*, 1 (2005), 224–231.

[189] Zhang, D., Zhou, Z.-H., and Chen, S. Diagonal principal component analysis for face recognition. *Pattern Recognition 39*, 1 (2006), 140–142.

[190] Zhang, G., Huang, X., Li, S. Z., Wang, Y., and Wu, X. Boosting local binary pattern (lbp)-based face recognition. In *Advances in biometric person authentication*. Springer, 2005, pp. 179–186.

[191] Zhang, Y., Hornfeck, K., and Lee, K. Adaptive face recognition for low-cost, embedded human-robot interaction. In *Intelligent Autonomous Systems 12*. Springer, 2013, pp. 863–872.

[192] Zhao, G., Ahonen, T., Matas, J., and Pietikainen, M. Rotation-invariant image and video description with local binary pattern features. *Image Processing, IEEE Transactions on 21*, 4 (2012), 1465–1477.

[193] Zhao, G., and Pietikainen, M. Dynamic texture recognition using local binary patterns with an application to facial expressions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on 29*, 6 (2007), 915–928.

[194] Zhao, G., and Pietikäinen, M. Dynamic texture recognition using volume local binary patterns. In *Dynamical Vision*. Springer, 2007, pp. 165–177.

[195] Zhao, G., and Pietikainen, M. Improving rotation invariance of the volume local binary pattern. *Mach. Vis. Appl.* (2007), 323–330.

[196] Zhao, X., and Zhang, S. Facial expression recognition based on local binary patterns and kernel discriminant isomap. *Sensors 11*, 10 (2011), 9573–9588.

[197] Zhao, Z., Wang, L., Liu, H., and Ye, J. On similarity preserving feature selection. *Knowledge and Data Engineering, IEEE Transactions on 25*, 3 (2013), 619–632.

[198] Zheng, W., Zhou, X., Zou, C., and Zhao, L. Facial expression recognition using kernel canonical correlation analysis (kcca). *Neural Networks, IEEE Transactions on 17*, 1 (2006), 233–238.

[199] Zuiderveld, K. Contrast limited adaptive histogram equalization. In *Graphics gems IV* (1994), Academic Press Professional, Inc., pp. 474–485.

[200] Zupan, B., Bohanec, M., Demšar, J., and Bratko, I. Learning by discovering concept hierarchies. *Artificial Intelligence 109*, 1 (1999), 211–242.