COMPARACIÓN DE MÉTODOS DE REDUCCIÓN DE DIMENSIONALIDAD EN LA PREDICCIÓN DE ENFERMEDADES OCULARES MEDIANTE REDES NEURONALES CONVOLUCIONALES

KATHERIN LICETH REYES ENCISO

UNIVERSIDAD INDUSTRIAL DE SANTANDER

FACULTAD DE CIENCIAS

ESCUELA DE MATEMÁTICAS

BUCARAMANGA

2026

COMPARACIÓN DE MÉTODOS DE REDUCCIÓN DE DIMENSIONALIDAD EN LA PREDICCIÓN DE ENFERMEDADES OCULARES MEDIANTE REDES NEURONALES CONVOLUCIONALES

KATHERIN LICETH REYES ENCISO

Trabajo de grado presentado como requisito parcial para optar al título de

Matemática

Director

PhD. en Ciencias Estadística Andrés Sebastián Ríos Gutiérrez

UNIVERSIDAD INDUSTRIAL DE SANTANDER

FACULTAD DE CIENCIAS

ESCUELA DE MATEMÁTICAS

BUCARAMANGA

2026

DEDICATORIA

A Dios,
por ser mi fortaleza en los momentos de duda,
por escucharme cuando no encontraba palabras
y por no permitir que me rindiera
cuando el camino parecía demasiado largo.

A mis padres,
por enseñarme que los sueños no tienen fecha de vencimiento
y que el esfuerzo siempre encuentra su recompensa.
Este logro lleva su nombre.

A mí,
por no rendirme cuando quise hacerlo,
por seguir adelante cuando el camino se puso difícil
y por cumplir esta promesa que me hice a mí misma.

AGRADECIMIENTOS

Este logro es el resultado de un camino largo y lleno de aprendizajes, en el que nunca estuve sola.

A mis **padres**, por su amor incondicional y su confianza inquebrantable en mí. Por cada palabra de aliento en los momentos en que quise rendirme, por su paciencia ante mis ausencias y por enseñarme que no importa cuán largo o difícil sea el camino, siempre vale la pena recorrerlo. Este título también les pertenece.

A mi **hermana**, por su compañía, su escucha y por saber siempre cómo hacerme sentir capaz cuando más lo necesitaba. Gracias por estar presente no solo en los momentos de celebración, sino también en los de duda y cansancio.

A la **Universidad Industrial de Santander**, por una formación académica sólida y por experiencias que van mucho más allá del aula. Con orgullo podré decir que soy su egresada.

A mi **director de tesis, el profesor Andrés Sebastián Ríos Gutiérrez**, por su paciencia, orientación y compromiso durante todo este proceso. Gracias a su dedicación y consejos, una idea se convirtió en un proyecto real y hoy puedo dar un paso más hacia la meta.

A mis amigos, profesores, compañeros y a todas las personas que han pasado por mi vida durante este camino llamado universidad, por cada aprendizaje, sonrisa y lección que hicieron este camino más llevadero.

Este título no es únicamente el resultado de mi esfuerzo, sino el reflejo de todas las personas que, directa o indirectamente, creyeron en mí.

Este trabajo de grado de pregrado se enmarca dentro de los productos del Grupo de Investigación en Procesos Estocásticos de la Universidad Nacional de Colombia sede Bogotá, registrado en el GrupLAC de MinCiencias. Corresponde a un producto dirigido por el profesor Andrés Ríos, en el contexto de su Beca de Excelencia Doctoral del Bicentenario Corte I, perteneciente a la convocatoria del Fondo de Ciencia, Tecnología e Innovación del Sistema General de Regalías para la conformación de priorizados y aprobados por el OCAD.

CONTENIDO

pág.

LISTA DE FIGURAS

pág.

LISTA DE TABLAS

pág.

LISTA DE CUADROS

pág.

LISTA DE ANEXOS

pág.

RESUMEN

TÍTULO: COMPARACIÓN DE MÉTODOS DE REDUCCIÓN DE DIMENSIONALIDAD EN LA PREDICCIÓN DE ENFERMEDADES OCULARES MEDIANTE REDES NEURONALES CONVOLUCIONALES*

AUTORA: KATHERIN LICETH REYES ENCISO**

PALABRAS CLAVE: reducción de dimensionalidad, enfermedades oculares, redes neuronales convolucionales, análisis de componentes principales, análisis factorial, UMAP, clasificación de imágenes médicas, desbalanceo de clases.

DESCRIPCIÓN:

Las enfermedades oculares representan un desafío para la salud pública a nivel global. Según la Organización Mundial de la Salud, más de 2.200 millones de personas sufren algún tipo de discapacidad visual, y aproximadamente la mitad de estos casos se habrían podido prevenir con una detección temprana. El diagnóstico tradicional, basado en la revisión manual de imágenes de retina, es costoso, subjetivo y difícil de implementar en regiones con escasez de especialistas. Este trabajo desarrolla y evalúa modelos de clasificación automática de enfermedades oculares a partir del conjunto de datos ODIR-5K (*Ocular Disease Intelligent Recognition*), aplicando tres métodos de reducción de dimensionalidad: PCA, Análisis Factorial y UMAP, conservando 30 componentes. Sobre estas representaciones se entrenaron cuatro modelos: Regresión Logística, SVM-RBF, CNN y ViT, evaluados con y sin estrategia de pesos de clase, y comparados contra modelos entrenados directamente sobre imágenes completas, incluyendo EfficientNetB3. Los resultados indican que, aunque EfficientNetB3 alcanza la mayor exactitud global, los modelos clásicos sobre representaciones reducidas son competitivos y viables cuando los recursos computacionales son limitados.

*Trabajo de Grado.

**Facultad de Ciencias. Escuela de Matemáticas. Director: PhD. Andrés Sebastián Ríos Gutiérrez.

ABSTRACT

TITLE: COMPARISON OF DIMENSIONALITY REDUCTION METHODS FOR OCULAR DISEASE PREDICTION USING CONVOLUTIONAL NEURAL NETWORKS*

AUTHOR: KATHERIN LICETH REYES ENCISO**

KEYWORDS: dimensionality reduction, ocular diseases, convolutional neural networks, principal component analysis, factor analysis, UMAP, medical image classification, class imbalance.

DESCRIPTION:

Ocular diseases represent a challenge for public health worldwide. According to the World Health Organization, more than 2.2 billion people suffer from some form of visual impairment, and approximately half of these cases could have been prevented through early detection. Traditional diagnosis, based on manual review of fundus images, is costly, subjective, and difficult to implement in regions with limited access to medical specialists. This work develops and evaluates automatic classification models for ocular diseases using the ODIR-5K dataset (*Ocular Disease Intelligent Recognition*), applying three dimensionality reduction methods: PCA, Factor Analysis, and UMAP, retaining 30 components. Four classification models were trained on these reduced representations: Logistic Regression, SVM-RBF, CNN, and ViT, evaluated with and without class weighting strategy, and compared against models trained directly on full images, including EfficientNetB3. Results indicate that, although EfficientNetB3 achieves the highest overall accuracy, classical models on reduced representations are competitive and viable when computational resources are limited.

*Bachelor's Thesis.

**Faculty of Sciences. School of Mathematics. Advisor: PhD. Andrés Sebastián Ríos Gutiérrez.

INTRODUCCIÓN

La salud visual es un componente esencial del bienestar humano y del desarrollo de las actividades cotidianas. No obstante, en las últimas décadas se ha evidenciado un incremento significativo en la prevalencia de enfermedades oculares, especialmente en poblaciones envejecidas y pacientes con condiciones como la diabetes. De acuerdo con el *World Report on Vision* de la Organización Mundial de la Salud (OMS), más de 2.200 millones de personas presentan algún grado de discapacidad visual, y cerca de la mitad de estos casos podrían haberse prevenido mediante una detección temprana y un acceso oportuno a servicios de salud visual de calidad[1].

Tradicionalmente, el diagnóstico de enfermedades oculares se realiza mediante el análisis manual de imágenes de retina (retinografías) por parte de especialistas en oftalmología. Sin embargo, este proceso resulta costoso en términos de tiempo y recursos, presenta un componente subjetivo asociado a la interpretación clínica y es difícil de implementar en regiones con acceso limitado a personal médico especializado[2]. Las técnicas de inteligencia artificial, y en particular el aprendizaje profundo, han surgido como herramientas prometedoras para asistir en el análisis automático de imágenes médicas, lo que permite identificar patrones complejos y apoya la detección temprana de diversas patologías[3] [4].

No obstante, el desarrollo de modelos de clasificación para el diagnóstico de enfermedades oculares enfrenta desafíos importantes relacionados con la naturaleza de los datos: la alta dimensionalidad de las imágenes de retina y el marcado desbalance entre categorías diagnósticas. Abordar estos desafíos de manera conjunta y rigurosa

[1] WORLD HEALTH ORGANIZATION. *World Report on Vision* [en línea]. Geneva: WHO, 2019. Disponible en: https://www.who.int/publications-detail-redirect/9789241516570

[2] LITJENS, Geert et al. A survey on deep learning in medical image analysis. En: *Medical Image Analysis*. 2017, vol. 42, pp. 60-88.

[3] Ibid.

[4] TING, Daniel S.W. et al. Development and validation of a deep learning system for diabetic retinopathy and related eye diseases. En: *JAMA*. 2017, vol. 318, nro. 22, pp. 2211-2223.

es fundamental para construir modelos que sean útiles en la práctica clínica[5] [6].

Este trabajo estudia el papel de las técnicas de reducción de dimensionalidad en la clasificación automática de enfermedades oculares a partir de imágenes de retina, evaluando su efecto sobre distintos modelos de clasificación tanto clásicos como basados en aprendizaje profundo.

Este documento se organiza de la siguiente manera: en el Capítulo 1 se presenta el planteamiento del problema; en el Capítulo 2 se definen los objetivos de la investigación; en el Capítulo 3 se desarrolla el marco teórico de las técnicas de reducción de dimensionalidad; en el Capítulo 4 se describen los métodos de clasificación para imágenes empleados en la investigación; en el Capítulo 5 se presentan los resultados obtenidos; en el Capítulo 6 se exponen las conclusiones; y en el Capítulo 7 se plantean recomendaciones y posibles líneas de trabajo futuro.

[5] GOODFELLOW, Ian; BENGIO, Yoshua y COURVILLE, Aaron. *Deep Learning*. Cambridge, MA: MIT Press, 2016. Disponible en: https://www.deeplearningbook.org

[6] LITJENS. Op. cit.

1. PLANTEAMIENTO DEL PROBLEMA

El análisis automático de imágenes de retina mediante técnicas de inteligencia artificial ha mostrado resultados prometedores en la detección de enfermedades oculares[7] [8]. Sin embargo, este problema presenta dos desafíos estructurales que condicionan el diseño de cualquier solución: la alta dimensionalidad de las imágenes, que introduce redundancia y eleva el costo computacional[9], y el marcado desbalance entre clases diagnósticas, que lleva a los modelos a favorecer las categorías más frecuentes y reduce la capacidad de detectar las enfermedades minoritarias[10] [11]. Esta situación puede observarse en el conjunto de datos utilizado en este trabajo, donde la categoría Normal concentra el 44,9 % de las imágenes, mientras que categorías como Hipertensión representan apenas el 2 %.

Las técnicas de reducción de dimensionalidad transforman los datos a representaciones más compactas preservando la información relevante, y pueden influir directamente en el comportamiento de los modelos de clasificación entrenados sobre dichas representaciones[12]. En este trabajo se consideran tres métodos que difieren en su naturaleza matemática: el Análisis de Componentes Principales (PCA), un método lineal que construye nuevas variables maximizando la varianza explicada[13]; el Análisis Factorial (AF), un método también lineal pero orientado a descubrir factores ocultos que explican las correlaciones entre las variables observadas[14]; y la Aproximación y Proyección Uniforme de Variedades (UMAP), un método no lineal capaz de preservar tanto estructuras locales como globales en los datos [15].

[7] Ibid.

[8] TING. Op. cit.

[9] JOLLIFFE, Ian T. *Principal Component Analysis*. 2 ed. Springer, 2002.

[10] HE, Haibo y GARCIA, Edwardo A. Learning from Imbalanced Data. En: *IEEE Transactions on Knowledge and Data Engineering*. 2009, vol. 21, nro. 9, pp. 1263-1284.

[11] KOTSIANTIS, S. B.; ZAHARAKIS, I. y PINTELAS, P. Supervised machine learning: A review of classification techniques. En: *Informatica*. 2007, vol. 31, pp. 249-268.

[12] JOLLIFFE. Op. cit.

[13] Ibid.

[14] ANDERSON, T. W. *An Introduction to Multivariate Statistical Analysis*. Wiley, 2003.

[15] McINNES, Leland; HEALY, John y MELVILLE, James. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction [en línea]. 2020. [Consultado: fecha de consulta]. Disponible en: https://arxiv.org/abs/1802.03426

No obstante, los trabajos previos presentan una limitación importante: en la mayoría de los casos se aplica PCA de forma independiente como paso de preprocesamiento[16], sin contrastarlo con otros métodos de reducción ni evaluar su efecto sobre el desempeño de los modelos. En particular, no se ha encontrado un estudio que compare de manera rigurosa estos tres métodos como estrategias de reducción aplicadas sobre imágenes de retina, evaluando su efecto sobre múltiples modelos de clasificación tanto clásicos como basados en aprendizaje profundo, y considerando el problema del desbalance de clases.

A partir de lo anterior, se formula la siguiente pregunta de investigación:

¿Cómo se comparan los distintos métodos de reducción de dimensionalidad en el desempeño de modelos de clasificación de enfermedades oculares a partir de imágenes de la retina, y en qué medida estos métodos mejoran o afectan los resultados frente al uso directo de las imágenes sin reducción previa?

[16] LITJENS. Op. cit.

2. OBJETIVOS

2.1. OBJETIVO GENERAL

Evaluar y comparar el impacto de distintos métodos de reducción de dimensionalidad en el desempeño de modelos de clasificación de enfermedades oculares a partir de imágenes de la retina, para modelos clásicos de aprendizaje automático y arquitecturas de aprendizaje profundo, contrastando los resultados con representaciones reducidas frente al uso directo de las imágenes.

2.2. OBJETIVOS ESPECÍFICOS

1. Aplicar técnicas de reducción de dimensionalidad, Análisis de Componentes Principales (PCA), Análisis Factorial (AF) y Aproximación y Proyección Uniforme de Variedades (UMAP), para obtener representaciones de menor dimensión a partir de imágenes de retina.
2. Entrenar modelos de clasificación, Regresión Logística, Máquinas de Soporte Vectorial (SVM), Redes Neuronales Convolucionales (CNN) y Transformadores de Visión (ViT) y EfficientNetB3, utilizando tanto las representaciones reducidas como las imágenes originales.
3. Comparar el desempeño de los modelos al emplear diferentes métodos de reducción de dimensionalidad mediante métricas de clasificación y análisis por clase.
4. Analizar las diferencias en el desempeño de los modelos al trabajar con representaciones reducidas frente al uso directo de las imágenes, identificando ventajas y limitaciones de cada enfoque.
5. Analizar el efecto del desbalance de clases en el desempeño de los modelos bajo los distintos enfoques considerados.

3. TÉCNICAS DE REDUCCIÓN DE LA DIMENSIONALIDAD

El conjunto de datos utilizado en este estudio corresponde al conjunto de datos ODIR-5K[17], una base de datos oftalmológica estructurada que contiene imágenes de retina recopiladas de distintos hospitales y centros médicos en China. Las imágenes fueron preprocesadas a un tamaño de 128×128 píxeles con tres canales de color (RGB), lo que equivale a un total de 49.152 características por imagen. Trabajar directamente con esta dimensionalidad introduce desafíos computacionales significativos y acentúa los efectos de la *"maldición de la dimensionalidad"* [18]. Este fenómeno indica que, aunque podría parecer que aumentar el número de variables explicativas siempre mejora un modelo, en realidad los datos en espacios de alta dimensión presentan mayor dispersión. Como consecuencia, agregar características irrelevantes puede dificultar que los modelos identifiquen patrones generalizables y aumentar el riesgo de un bajo ajuste, lo que se refleja en los datos de prueba.

En este capítulo se presentan distintas técnicas de reducción de la dimensionalidad[19] cuyo propósito es transformar el espacio de representación original en un conjunto con menor cantidad de variables. Esta reducción preserva la información descrita por la varianza y reduce el costo computacional del entrenamiento.

3.1. PRELIMINARES

En esta sección se introducen los conceptos estadísticos fundamentales que sustentan los métodos de reducción de dimensionalidad desarrollados en este capítulo: el Análisis de Componentes Principales (PCA), el Análisis Factorial (AF) y la Aproximación y Proyección Uniforme de Variedades (UMAP). Concretamente, se define: el

[17] LARXEL. Ocular Disease Recognition ODIR5K [en línea]. Kaggle, 2020. [Consultado: 2024]. Disponible en: https://www.kaggle.com/datasets/andrewmvd/ocular-disease-recognition-odir5k

[18] JAMES, Gareth; WITTEN, Daniela; HASTIE, Trevor y TIBSHIRANI, Robert. *An Introduction to Statistical Learning: with Applications in R*. New York: Springer, 2013. ISBN 978-1-4614-7137-0.

[19] VAN DER MAATEN, L. J. P.; POSTMA, E. O. y VAN DEN HERIK, H. J. Dimensionality Reduction: A Comparative Review. En: *Journal of Machine Learning Research*. 2009, vol. 10, pp. 1-41.

vector aleatorio, el vector de medias, la matriz de covarianza y la matriz de correlación, junto con su interpretación estadística en el contexto multivariado.

3.1.1. Vector aleatorio. Sea $\mathbf{X} \in \mathbb{R}^p$ con $\mathbf{X} = (X_1, X_2, \ldots, X_p)^\mathsf{T}$, donde cada componente X_j , es una variable aleatoria real con esperanza y varianza finitas. Este vector modela un conjunto de p variables cuantitativas observadas simultáneamente.

3.1.2. Vector de medias. La media del vector aleatorio $\mathbf{X} = (X_1, \ldots, X_p)^\mathsf{T}$ se define como el vector

$$\boldsymbol{\mu} = \mathbb{E}[\mathbf{X}] = \begin{bmatrix} \mathbb{E}[X_1] \\ \vdots \\ \mathbb{E}[X_p] \end{bmatrix}.$$

Cada componente $\mu_j = \mathbb{E}[X_j]$ representa el valor medio de la variable aleatoria X_j.

En la práctica, la distribución poblacional de $\mathbf{X}$ suele ser desconocida y se dispone únicamente de los datos $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ de una muestra. En este caso, el vector de medias se estima mediante el promedio muestral

$$\hat{\boldsymbol{\mu}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i.$$

El vector de medias desempeña un papel central en el análisis multivariante, ya que numerosas técnicas, como el análisis de componentes principales (PCA) y el análisis factorial (AF), se aplican sobre datos *centrados*, es decir, expresados en términos de desviaciones respecto a la media[20] [21].

3.1.3. Matriz de covarianza. La covarianza es una medida estadística que cuantifica cómo varían conjuntamente dos variables aleatorias[22]. A partir de esta medida se

[20] ANDERSON. Op. cit.
[21] TUSELL, Fernando. *Análisis multivariante*. F. Tusell, 1999.
[22] ANDERSON. Op. cit.

construye la matriz de covarianza, cuyos elementos diagonales corresponden a las varianzas de cada variable y cuyos elementos fuera de la diagonal representan las covarianzas entre pares de variables[23] [24].

Sean X_i y X_j dos variables aleatorias reales con medias $\mu_i = \mathbb{E}[X_i]$ y $\mu_j = \mathbb{E}[X_j]$. La covarianza poblacional entre X_i y X_j se define como:

$$\mathrm{Cov}(X_i, X_j) = \mathbb{E}[(X_i - \mu_i)(X_j - \mu_j)]\,.$$

Cuando $i = j$, esta expresión coincide con la varianza de la variable X_i, es decir,

$$\mathrm{Var}(X_i) = \mathrm{Cov}(X_i, X_i),$$

la cual es una medida de dispersión de X_i alrededor de su media.

En el contexto multivariado, se considera un vector aleatorio $\mathbf{X} = (X_1, \ldots, X_p)^\mathsf{T}$. Para describir de manera conjunta la variabilidad de todas sus componentes, se introducen las covarianzas entre cada par de variables, las cuales se organizan en la denominada matriz de covarianza.

La matriz de covarianza de $\mathbf{X}$, denotada por $\boldsymbol{\Sigma} \in \mathbb{R}^{p\times p}$, se define como

$$\boldsymbol{\Sigma} = \mathbb{E}\big[(\mathbf{X} - \boldsymbol{\mu})(\mathbf{X} - \boldsymbol{\mu})^\mathsf{T}\big]\,,$$

donde $\boldsymbol{\mu} = \mathbb{E}[\mathbf{X}]$ es el vector de medias.

El elemento (i, j) de la matriz $\boldsymbol{\Sigma}$ está dado por

$$\Sigma_{ij} = \mathrm{Cov}(X_i, X_j).$$

De manera explícita,

[23] Ibid.
[24] TUSELL. Op. cit.

$$\boldsymbol{\Sigma} = \begin{pmatrix} \mathrm{Var}(X_1) & \mathrm{Cov}(X_1, X_2) & \cdots & \mathrm{Cov}(X_1, X_p) \\ \mathrm{Cov}(X_2, X_1) & \mathrm{Var}(X_2) & \cdots & \mathrm{Cov}(X_2, X_p) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{Cov}(X_p, X_1) & \mathrm{Cov}(X_p, X_2) & \cdots & \mathrm{Var}(X_p) \end{pmatrix}.$$

La matriz de covarianza es simétrica y semidefinida positiva, propiedades que resultan fundamentales en técnicas de análisis multivariado como el análisis de componentes principales (PCA) y el análisis factorial (AF), donde su estructura desempeña un papel central[25] [26].

3.1.4. Matriz de correlación. Aunque la covarianza permite cuantificar la correlación lineal entre dos variables aleatorias, su interpretación depende de las unidades de medida originales, lo que dificulta la comparación directa entre distintos pares de variables[27]. Para superar esta limitación, se introduce la correlación de Pearson, la cual puede interpretarse como una covarianza estandarizada [28].

La correlación se obtiene normalizando la covarianza mediante las desviaciones estándar de las variables involucradas, lo que da lugar a una medida adimensional y acotada en el intervalo $[-1, 1]$. Esta propiedad de acotamiento permite evaluar de forma objetiva la dependencia lineal entre variables y establecer un grado de dependencia de una respecto a la otra [29] [30].

El coeficiente de correlación de Pearson entre dos variables X_i y X_j se define como [31] [32]

$$\rho_{ij} = \frac{\mathrm{Cov}(X_i, X_j)}{\sqrt{\mathrm{Var}(X_i)\,\mathrm{Var}(X_j)}}.$$

[25] ANDERSON. Op. cit.
[26] JOLLIFFE. Op. cit.
[27] ANDERSON. Op. cit.
[28] TUSELL. Op. cit.
[29] HERNÁNDEZ LALINDE, J. D. et al. Sobre el uso adecuado del coeficiente de correlación de Pearson. En: *Sociedad Venezolana de Farmacología Clínica y Terapéutica*. 2018.
[30] PINILLA, J. O. y RICO, A. F. ¿Pearson y Spearman, coeficientes intercambiables? En: *Comunicaciones en Estadística*. 2021.
[31] ANDERSON. Op. cit.
[32] TUSELL. Op. cit.

Este coeficiente toma valores en el intervalo $[-1, 1]$, donde[33]:

- $\rho_{ij} = 1$ indica una relación lineal positiva perfecta,
- $\rho_{ij} = -1$ indica una relación lineal negativa perfecta,
- $\rho_{ij} = 0$ indica ausencia de relación lineal.

La *matriz de correlación* $\mathbf{R} \in \mathbb{R}^{p\times p}$ se construye a partir de estos coeficientes y tiene la forma

$$\mathbf{R} = \begin{pmatrix} 1 & \rho_{12} & \cdots & \rho_{1p} \\ \rho_{21} & 1 & \cdots & \rho_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{p1} & \rho_{p2} & \cdots & 1 \end{pmatrix}.$$

En técnicas como el análisis de componentes principales (PCA) y análisis factorial (AF), la matriz de correlación se utiliza cuando se desea eliminar el efecto de las unidades de medida y analizar únicamente las relaciones lineales estandarizadas entre las variables[34] [35].

3.2. ANÁLISIS DE COMPONENTES PRINCIPALES

El **Análisis de Componentes Principales (PCA)** es un método estadístico de reducción de dimensionalidad que, a partir de un conjunto de variables originales construye un nuevo conjunto de variables no correlacionadas, denominadas *componentes principales*. Cada componente principal se obtiene como una combinación lineal de las variables originales y se define de manera que explique, de forma sucesiva, la mayor cantidad posible de la varianza total de los datos, bajo la restricción de ortogonalidad con respecto a los componentes previamente definidos [36] [37].

[33] ANDERSON. Op. cit.
[34] HAIR, J. et al. *Multivariate Data Analysis*. Cengage, 2019.
[35] HERNÁNDEZ LALINDE. Op. cit.
[36] ANDERSON. Op. cit.
[37] JOLLIFFE. Op. cit.

En este trabajo, el PCA se aplica sobre imágenes, donde cada observación corresponde a una imagen y cada variable representa la intensidad de un píxel o de una región de la imagen.

Tanto en el PCA como en el AF, la variabilidad presente en los datos puede ser explicada a través de los valores propios o autovalores asociados a la matriz de covarianza o de correlación [38] [39].

Autovalores y autovectores. Sea $\mathbf{A} \in \mathbb{R}^{p \times p}$ una matriz cuadrada. Un escalar $\lambda \in \mathbb{R}$ se denomina **autovalor** (o valor propio) de $\mathbf{A}$ si existe un vector no nulo $\mathbf{v} \in \mathbb{R}^p \setminus \{\mathbf{0}\}$ tal que:

$$\mathbf{A}\mathbf{v} = \lambda \mathbf{v}.$$

Esta ecuación puede reescribirse como un sistema lineal homogéneo:

$$(\mathbf{A} - \lambda \mathbf{I})\mathbf{v} = \mathbf{0},$$

donde $\mathbf{I}$ es la matriz identidad de orden p.

Para que este sistema tenga una solución no trivial (es decir, $\mathbf{v} \neq \mathbf{0}$), la matriz $(\mathbf{A} - \lambda \mathbf{I})$ debe ser singular. Esto impone la condición de que su determinante sea cero, lo que da lugar al **polinomio característico**:

$$p(\lambda) = \det(\mathbf{A} - \lambda \mathbf{I}) = 0.$$

Las raíces de este polinomio corresponden a los autovalores de la matriz. Una vez determinado un autovalor λ, los **autovectores** asociados se obtienen resolviendo el sistema lineal $(\mathbf{A} - \lambda \mathbf{I})\mathbf{v} = \mathbf{0}$ [40] [41] .

[38] ANDERSON. Op. cit.
[39] JOLLIFFE. Op. cit.
[40] HORN, Roger A. y JOHNSON, Charles R. *Matrix Analysis*. 2 ed. Cambridge University Press, 2013.
[41] STRANG, Gilbert. *Introduction to Linear Algebra*. 5 ed. Wellesley, MA: Wellesley-Cambridge Press, 2016.

Fundamento matemático. Sea $\mathbf{X} \in \mathbb{R}^p$ un vector aleatorio cuyas componentes han sido previamente estandarizadas, es decir,

$$\mathbb{E}[\mathbf{X}] = \mathbf{0} \quad \text{y} \quad \mathrm{Cov}(\mathbf{X}) = \mathbf{R}.$$

La estandarización implica que cada variable tiene media cero y varianza uno, lo cual resulta especialmente apropiado cuando las variables originales se encuentran en escalas distintas, ya que evita que aquellas con mayor variabilidad dominen el análisis y permite que la construcción de los componentes principales dependa exclusivamente de las relaciones lineales entre las variables [42].

El objetivo del PCA es encontrar direcciones en las que la variabilidad de los datos sea máxima. Este problema se reduce a diagonalizar la matriz de correlación $\mathbf{R}$, es decir, encontrar una base en la que $\mathbf{R}$ tenga una representación diagonal. Sin embargo, para que esta diagonalización exista, sea única y produzca direcciones ortogonales entre sí, se requiere que la matriz sea simétrica. Dado que $\mathbf{R}$ satisface esta condición por construcción, el Teorema Espectral para matrices reales simétricas garantiza la existencia de dicha descomposición [43].

Teorema 1 (Teorema Espectral para matrices reales simétricas [44])**.** Sea $A \in \mathbb{R}^{p\times p}$ una matriz simétrica. Entonces existen una matriz ortogonal $Q \in \mathbb{R}^{p\times p}$ y una matriz diagonal $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_p)$, con $\lambda_j \in \mathbb{R}$, tales que

$$A = Q\Lambda Q^\top,$$

donde las columnas de Q forman una base ortonormal de autovectores de A.

Por el Teorema espectral existe una matriz ortogonal $V \in \mathbb{R}^{p\times p}$ y una matriz diagonal $\Lambda = \mathrm{diag}(\lambda_1, \ldots, \lambda_p)$ tales que

$$\mathbf{R} = V\Lambda V^\top.$$

donde, $V = [v_1, v_2, \ldots, v_p] \in \mathbb{R}^{p\times p}$ es una matriz ortogonal cuyas columnas v_j son los

[42] JOLLIFFE. Op. cit.
[43] STRANG. Op. cit.
[44] Ibid.

autovectores de R, y

$$\Lambda = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_p)$$

es la matriz diagonal de autovalores, ordenados de manera no creciente[45]:

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0.$$

Proposición 1. Dado que V es ortogonal y Λ es diagonal, si v_j denota la columna j-ésima de V, entonces

$$\mathbf{R}v_j = \lambda_j v_j,$$

Demostración. Se parte de la descomposición

$$\mathbf{R} = V\Lambda V^\top.$$

Entonces

$$\mathbf{R}v_j = V\Lambda V^\top v_j.$$

Como V es ortogonal, sus columnas forman un sistema ortonormal, por lo que

$$V^\top v_j = e_j,$$

donde $e_j \in \mathbb{R}^p$ es el vector canónico, es decir, el vector cuyos componentes son todos cero excepto un 1 en la posición j:

Por tanto,

$$\mathbf{R}v_j = V\Lambda e_j.$$

Ahora bien, dado que Λ es una matriz diagonal, al multiplicarla por el vector canónico

[45]JOLLIFFE. Op. cit.

e_j se obtiene

$$\Lambda e_j = \lambda_j e_j,$$

ya que la única componente no nula de e_j corresponde precisamente a la entrada diagonal λ_j.

En consecuencia,

$$\mathbf{R}v_j = V(\lambda_j e_j) = \lambda_j V e_j = \lambda_j v_j.$$

Por lo tanto, v_j es el autovector de $\mathbf{R}$ asociado al autovalor λ_j. □

Dado que cada autovalor λ_j cuantifica la variabilidad de los datos en la dirección definida por su autovector asociado v_j, seleccionar los autovectores asociados a los mayores autovalores equivale a retener las direcciones que concentran la mayor cantidad de variabilidad presente en los datos. Esto permite construir una representación de menor dimensión que preserva la mayor parte de la información original[46].

Sea $k < p$ el número de autovectores seleccionados (asociados a los mayores autovalores) [47] [48]. Se define la matriz

$$W_k = [v_1, v_2, \ldots, v_k] \in \mathbb{R}^{p \times k},$$

cuyas columnas son los k autovectores seleccionados.

El subespacio generado por las columnas de W_k es un subespacio de dimensión k de $\mathbb{R}^p$. Proyectar el vector aleatorio $\mathbf{X}$ sobre este subespacio equivale a expresar $\mathbf{X}$ como combinación lineal de dichos vectores [49].

[46] JOLLIFFE. Op. cit.
[47] ANDERSON. Op. cit.
[48] JOLLIFFE. Op. cit.
[49] Ibid.

En forma matricial, esta proyección se obtiene mediante

$$Z = W_k^\top \mathbf{X} \ \in \mathbb{R}^k,$$

donde cada componente está dado por $Z_j = v_j^\top \mathbf{X}, \quad j = 1, \ldots, k.$

El vector aleatorio Z corresponde a la representación de $\mathbf{X}$ en el espacio reducido generado por los primeros k componentes principales [50] [51].

Definición 1. Sea $\mathbf{X} \in \mathbb{R}^p$ un vector aleatorio estandarizado, y sea $v_j \in \mathbb{R}^p$ el autovector de la matriz de correlación $\mathbf{R}$ asociado al autovalor λ_j, donde $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$ [52] [53].

El j-ésimo componente principal, denotado por PC_j, se define como la variable aleatoria

$$PC_j = v_j^\top \mathbf{X}.$$

De manera explícita,

$$PC_j = v_{1j}X_1 + v_{2j}X_2 + \cdots + v_{pj}X_p = \sum_{i=1}^{p} v_{ij}X_i.$$

Recordemos que una combinación lineal de las variables aleatorias $X_1, \ldots, X_p$ es toda expresión de la forma $a_1X_1 + \cdots + a_pX_p$, con $a_i \in \mathbb{R}$ [54]. Dado que los coeficientes $v_{1j}, \ldots, v_{pj}$ son números reales, se concluye que cada componente principal es una combinación lineal de las variables originales estandarizadas.

Una vez establecida la naturaleza de los componentes principales como combinaciones lineales de las variables estandarizadas, el siguiente paso es analizar su importancia estadística. A continuación, se justificará por qué los autovalores de $\mathbf{R}$ no son

[50] ANDERSON. Op. cit.
[51] JOLLIFFE. Op. cit.
[52] ANDERSON. Op. cit.
[53] JOLLIFFE. Op. cit.
[54] STRANG. Op. cit.

simples escalares, sino que representan con precisión la cantidad de varianza capturada por cada componente [55] [56]

Sea $\mathbf{X} = (X_1, \ldots, X_p)^\top$ un vector aleatorio de variables estandarizadas, y sea $a = (a_1, \ldots, a_p)^\top \in \mathbb{R}^p$. Considérese la combinación lineal

$$Y = \sum_{i=1}^{p} a_i X_i = a^\top \mathbf{X}.$$

De acuerdo con [57], la varianza de una combinación lineal puede escribirse como

$$\operatorname{Var}\left(\sum_{i=1}^{p} a_i X_i\right) = \sum_{i=1}^{p} a_i^2 \operatorname{Var}(X_i) + 2 \sum_{1 \le i < j \le p} a_i a_j \operatorname{Cov}(X_i, X_j). \tag{1}$$

Observe que la parte derecha de (1) puede ser simplificada con la siguiente expresión:

$$\sum_{i=1}^{p} \sum_{j=1}^{p} a_i R_{ij} a_j. \tag{2}$$

Esta doble suma incluye todos los pares (i, j).

Caso 1: Cuando $i = j$,

$$a_i R_{ii} a_i = a_i^2 \operatorname{Var}(X_i),$$

que coincide exactamente con los términos de la primera suma en (1).

Caso 2: $i \neq j$**.**

[55] ANDERSON. Op. cit.

[56] JOLLIFFE. Op. cit.

[57] WASSERMAN, Larry. *All of Statistics: A Concise Course in Statistical Inference*. New York: Springer, 2004.

Para cada par $i \neq j$, aparecen dos términos:

$$a_i R_{ij} a_j \quad \text{y} \quad a_j R_{ji} a_i.$$

Como la matriz de correlación es simétrica,

$$R_{ij} = R_{ji},$$

entonces

$$a_i R_{ij} a_j + a_j R_{ji} a_i = 2 a_i a_j R_{ij}.$$

Esto explica por qué el factor 2 en (1) no desaparece: queda implícito al sumar ambos términos simétricos en la doble suma.

Ahora, como las variables están estandarizadas, $\mathrm{Cov}(X_i, X_j) = R_{ij}$, por lo que (1) puede escribirse como

$$\mathrm{Var}(Y) = \sum_{i=1}^{p} \sum_{j=1}^{p} a_i R_{ij} a_j.$$

A partir de esto se demuestra que:

Proposición 2. Dado que $\sum_{i=1}^{p} \sum_{j=1}^{p} a_i R_{ij} a_j = a^\top \mathbf{R}\, a$, entonces

$$\mathrm{Var}(Y) = a^\top \mathbf{R}\, a.$$

Demostración. Sea

$$a = \begin{pmatrix} a_1 \\ \vdots \\ a_p \end{pmatrix}, \qquad a^\top = (a_1, \ldots, a_p).$$

Primero se calcula el producto $\mathbf{R}a$:

$$\mathbf{R}a = \begin{pmatrix} \sum_{j=1}^{p} R_{1j}a_j \\ \sum_{j=1}^{p} R_{2j}a_j \\ \vdots \\ \sum_{j=1}^{p} R_{pj}a_j \end{pmatrix}.$$

Ahora se multiplica a la izquierda por $a^\top$:

$$a^\top(\mathbf{R}a) = (a_1, \ldots, a_p) \begin{pmatrix} \sum_{j=1}^{p} R_{1j}a_j \\ \sum_{j=1}^{p} R_{2j}a_j \\ \vdots \\ \sum_{j=1}^{p} R_{pj}a_j \end{pmatrix}.$$

Esto equivale a

$$\sum_{i=1}^{p} a_i \left(\sum_{j=1}^{p} R_{ij}a_j \right),$$

y distribuyendo,

$$\sum_{i=1}^{p} \sum_{j=1}^{p} a_i R_{ij} a_j.$$

Por lo tanto,

$$\mathrm{Var}(Y) = a^\top \mathbf{R} a.$$

□

Proposición 3. La Proposición 2 vale para cualquier vector $a \in \mathbb{R}^p$. En particular, tomando $a = v_j$ como el autovector ortonormal de $\mathbf{R}$ asociado al autovalor λ_j, se obtiene que

$$\mathrm{Var}(PC_j) = \lambda_j.$$

Demostración. Tomando $a = v_j$ y sea v_j un autovector de $\mathbf{R}$ asociado al autovalor λ_j,

es decir,

$$\mathbf{R}v_j = \lambda_j v_j.$$

Se define el j-ésimo componente principal como

$$PC_j = v_j^\top \mathbf{X}.$$

Aplicando el resultado general,

$$\mathrm{Var}(PC_j) = v_j^\top \mathbf{R} v_j.$$

Sustituyendo la ecuación propia,

$$v_j^\top (\lambda_j v_j) = \lambda_j v_j^\top v_j.$$

Como los autovectores de una matriz simétrica pueden elegirse ortonormales,

$$v_j^\top v_j = 1,$$

por tanto,

$$\mathrm{Var}(PC_j) = \lambda_j.$$

□

Así queda demostrado que cada autovalor representa exactamente la varianza explicada por el componente principal correspondiente [58].

Si se seleccionan los primeros k componentes principales, la varianza total explicada

[58] JOLLIFFE. Op. cit.

por dichos componentes es

$$\sum_{j=1}^{k} \lambda_j.$$

Como la varianza total del sistema es la suma de las varianzas explicadas por todos los componentes principales,

$$\sum_{j=1}^{p} \lambda_j,$$

la proporción de varianza explicada por los primeros k componentes principales está dada por

$$\frac{\sum_{j=1}^{k} \lambda_j}{\sum_{j=1}^{p} \lambda_j}.$$

Esta cantidad permite evaluar cuánta variabilidad del conjunto de datos original es retenida al proyectar sobre los primeros k componentes principales[59].

3.3. MODELIZACIÓN CON ANÁLISIS DE COMPONENTES PRINCIPALES

La elección del número adecuado de componentes k es un paso fundamental dentro del análisis de componentes principales, donde $k < p$ y $p = 49.152$ que corresponde al número total de variables explicativas. Conservar muy pocos componentes puede llevar a una pérdida de información relevante, mientras que conservar demasiados, incluso hasta $k = p$, introduce ruido que no aporta al proceso de clasificación y anula el objetivo de reducción dimensional [60] [61].

Para tomar esta decisión de forma objetiva, se aplicaron criterios estadísticos ampliamente utilizados, evitando seleccionar un valor de k de manera arbitraria. Estos criterios se evaluaron exclusivamente sobre el conjunto de entrenamiento, con el fin de prevenir cualquier fuga de información hacia el conjunto de prueba.

[59] Ibid.
[60] ANDERSON. Op. cit.
[61] JOLLIFFE. Op. cit.

Debido al gran número de variables explicativas del conjunto original (p = 49.152), trabajar directamente con la matriz de correlación completa resulta computacionalmente inviable. Por tanto, se adoptó un enfoque iterativo basado en submuestreos aleatorios: en cada iteración se extrajo aleatoriamente un subconjunto de 2.000 observaciones del conjunto de entrenamiento y se aplicaron los criterios de Yeomans-Golder y el Análisis Paralelo de Horn para determinar el número óptimo de componentes. Este proceso se repitió tres veces con distintos subconjuntos, con el fin de verificar la estabilidad y consistencia de los resultados obtenidos [62].

3.3.1. Criterios de selección aplicados.

- **Criterio de Yeomans-Golder (Kaiser):** Este criterio propone conservar únicamente los componentes cuyo autovalor sea mayor que 1 $(\lambda > 1)$. La idea central es que cada componente retenido debe explicar al menos tanta variabilidad como una variable original estandarizada, lo que permite descartar componentes que aportan una proporción mínima de información[63].

- **Análisis paralelo de Horn:** El análisis paralelo de Horn es un método para determinar el número de componentes o factores a retener para aplicar reducción de la dimensionalidad [64]. Cabe tener en cuenta que la matriz de correlación calculada a partir de una muestra finita no será exactamente la matriz identidad, incluso cuando las variables sean independientes en la población. Como consecuencia, pueden aparecer autovalores mayores a 1 incluso en ausencia de estructura real en los datos[65].

  El método consiste en comparar los autovalores obtenidos a partir de la matriz de correlación original con los autovalores obtenidos a partir de datos simulados de las mismas dimensiones $n \times p$. Un componente se retiene únicamente si su autovalor, calculado sobre la matriz de correlación original, supera al autovalor correspondiente obtenido en los datos simulados.

[62] YEOMANS, K. A. y GOLDER, P. A. The Guttman-Kaiser criterion as a predictor of the number of common factors. En: *Journal of the Royal Statistical Society. Series D (The Statistician)*. 1982, vol. 31, nro. 3, pp. 221-229.

[63] JOLLIFFE. Op. cit.

[64] HORN, John L. A Rationale and Test for the Number of Factors in Factor Analysis. En: *Psychometrika*. 1965, vol. 30, nro. 2, pp. 179-185.

[65] ANDERSON. Op. cit.

El procedimiento para obtener los autovalores asociados a los datos simulados es el siguiente:

1. Se generan S matrices de datos aleatorias de dimensión $n \times p$. Los valores se simulan de forma independiente con distribución $\mathcal{N}(0, 1)$, de modo que no existe correlación entre las variables.
2. Para cada matriz simulada se calcula su matriz de correlación y se obtienen sus autovalores ordenados de mayor a menor.
3. Para cada posición j, se calcula el promedio de los autovalores simulados en esa posición:

$$\bar{\lambda}_j = \frac{1}{S} \sum_{s=1}^{S} \lambda_j^{(s)}, \quad j = 1, \ldots, p.$$

Regla de decisión. Se retiene el componente j si su autovalor observado supera el promedio simulado en esa misma posición:

$$\lambda_j^{\text{real}} > \bar{\lambda}_j.$$

En caso contrario, el componente se descarta [66] [67].

Relación con el criterio de Kaiser. El criterio de Kaiser propone retener los componentes tales que $\lambda_j > 1$.

Sin embargo, en muestras finitas es frecuente obtener autovalores mayores que 1 incluso cuando las variables son independientes, lo que puede llevar a sobreestimar el número de factores retenidos.

El análisis paralelo mejora este criterio al reemplazar el umbral fijo de 1 por un valor de referencia obtenido mediante simulación, que depende de n y p. Por esta razón, suele considerarse un procedimiento más robusto para determinar el número de factores y componentes [68] [69].

[66] HORN, John L. Op. cit.
[67] JOLLIFFE. Op. cit.
[68] HORN, John L. Op. cit.
[69] JOLLIFFE. Op. cit.

3.4. ANÁLISIS FACTORIAL

El Análisis Factorial (AF) es una técnica estadística que busca explicar por qué varias variables observadas se encuentran relacionadas entre sí. Su objetivo es identificar un número reducido de factores que permitan describir las relaciones existentes entre las variables originales, de manera que la información compartida entre ellas pueda representarse de forma más compacta.

A diferencia de otros métodos de reducción de dimensionalidad, el AF parte de la idea de que la variabilidad de cada variable puede separarse en dos componentes: una parte que es compartida con otras variables y una parte que le es propia. La parte compartida se denomina varianza común, ya que explica las correlaciones entre las variables. La parte propia se denomina varianza específica o varianza única, y corresponde a características particulares de cada variable que no se comparten con las demás [70].

Aunque el AF suele compararse con el PCA, ambos métodos tienen propósitos distintos. El PCA busca construir nuevas variables que expliquen la mayor cantidad posible de variabilidad total presente en los datos, sin distinguir si dicha variabilidad es compartida entre variables o si es propia de cada una. En cambio, el AF se centra exclusivamente en explicar la variabilidad que las variables tienen en común, es decir, aquella que origina sus correlaciones [71] [72].

En la práctica, la elección entre ambas técnicas depende del objetivo del análisis. El PCA resulta apropiado cuando se busca resumir la información en menos variables, sin hacer supuestos sobre la estructura de los datos. El AF, en cambio, es más adecuado cuando se considera que las correlaciones entre las variables se deben a la influencia de factores no observados directamente. En este trabajo, ambas técnicas se emplean con fines de reducción dimensional, aunque difieren en su formulación matemática y

[70] JOHNSON, Richard A. y WICHERN, Dean W. *Applied Multivariate Statistical Analysis*. 6 ed. Pearson, 2007.
[71] Ibid.
[72] JOLLIFFE. Op. cit.

en los supuestos que imponen sobre los datos [73] [74].

Desde el punto de vista del modelo matemático, la diferencia es estructural: en PCA se construyen nuevas variables como combinaciones lineales de las variables originales. En el AF ocurre lo contrario: cada variable observada se expresa como una combinación lineal de los factores comunes más un término adicional que representa su parte específica, usualmente denotado por ε_i. Este término recoge la variabilidad propia de la variable que no es explicada por los factores comunes [75] [76].

Dado que las variables consideradas fueron previamente estandarizadas, el AF se realiza a partir de la matriz de correlación $\mathbf{R}$, en lugar de la matriz de covarianza. Esta elección permite trabajar con variables en una misma escala y analizar directamente la estructura de asociación entre ellas. La definición formal de la matriz de correlación fue presentada en la sección 3.1.

3.4.1. Fundamento teórico. El modelo factorial postula que cada variable observada X_i puede expresarse como una combinación lineal de los factores más un término de error [77] [78] [79]:

$$X_i = l_{i1}F_1 + l_{i2}F_2 + \cdots + l_{im}F_m + \varepsilon_i, \quad \text{para } i = 1, 2, \ldots, p,$$

donde:

- l_{ij} es la **carga factorial**, que cuantifica la contribución del factor F_j en la explicación de la variable X_i.
- F_j es el j-ésimo factor común, responsable de explicar la variabilidad compartida entre las variables. Los factores comunes se modelan como variables aleatorias

[73] HARMAN, Harry H. *Modern Factor Analysis*. 3 ed. Chicago: University of Chicago Press, 1976.
[74] JOHNSON, Richard A. Op. cit.
[75] HARMAN. Op. cit.
[76] JOHNSON, Richard A. Op. cit.
[77] HARMAN. Op. cit.
[78] JOHNSON, Richard A. Op. cit.
[79] LAWLEY, D. N. y MAXWELL, A. E. *Factor Analysis as a Statistical Method*. 2 ed. London: Butterworths, 1971.

centradas, con varianza unitaria e incorrelacionadas entre sí [80] [81], es decir,

$$\mathbb{E}(\mathbf{F}) = 0, \qquad \mathrm{Cov}(\mathbf{F}) = I_m \tag{1}$$

- ε_i es el **término específico** o **factor único** asociado a X_i, que recoge la parte de la variabilidad propia de la variable no explicada por los factores comunes. Estos términos se modelan como variables aleatorias centradas, con varianza ψ_i, incorrelacionadas entre sí y también incorrelacionadas con los factores comunes. En forma matricial,

$$\mathbb{E}(\varepsilon) = 0, \qquad \mathrm{Cov}(\varepsilon) = \Psi, \tag{2}$$

  donde Ψ es una matriz diagonal con entradas positivas [82] [83].

Las condiciones anteriores definen el modelo clásico del análisis factorial [84] [85] [86] .

La ecuación (2) puede escribirse de forma matricial como $\mathbf{X} = L\mathbf{F} + \boldsymbol{\varepsilon}$, donde $\mathbf{X} \in \mathbb{R}^p$ es el vector aleatorio de variables observadas, $L \in \mathbb{R}^{p\times m}$ es la matriz de cargas factoriales, $\mathbf{F} = (F_1, \ldots, F_m)^\top \in \mathbb{R}^m$ es el vector de factores comunes y $\boldsymbol{\varepsilon} \in \mathbb{R}^p$ es el vector de términos específicos[87].

El principal objetivo del AF es determinar la matriz de cargas factoriales L, pues esta describe la relación entre las variables observadas y los factores comunes [88] [89]. Sin embargo, ni los factores $\mathbf{F}$ ni los términos específicos $\boldsymbol{\varepsilon}$ son observables directamente, ya que no se trata de variables medidas en los datos, sino de cantidades que el modelo asume como existentes para explicar las correlaciones entre las variables observadas. Por esta razón, la ecuación estructural $\mathbf{X} = L\mathbf{F} + \boldsymbol{\varepsilon}$ no puede resolverse directamente para obtener las cargas factoriales L, pues contiene más incógnitas que información

[80] ANDERSON. Op. cit.
[81] LAWLEY. Op. cit.
[82] HARMAN. Op. cit.
[83] LAWLEY. Op. cit.
[84] ANDERSON. Op. cit.
[85] HARMAN. Op. cit.
[86] LAWLEY. Op. cit.
[87] ANDERSON. Op. cit.
[88] HARMAN. Op. cit.
[89] JOHNSON, Richard A. Op. cit.

disponible [90] [91].

En consecuencia, la identificación de L debe basarse en la estructura de dependencia de las variables observadas. Esto requiere expresar el modelo en términos de la matriz de covarianza poblacional de $\mathbf{X}$. En particular, se busca obtener una representación de la forma

$$\Sigma = LL^\top + \Psi,$$

ya que esta descomposición permitirá posteriormente estimar las cargas factoriales a partir de la matriz de correlación observada.

Para establecer dicha expresión, el modelo clásico incorpora además las siguientes condiciones [92] [93]:

$$\mathrm{Cov}(\mathbf{F}, \boldsymbol{\varepsilon}) = 0,$$

lo cual implica que los factores comunes y los términos específicos son incorrelacionados, garantizando que la variabilidad común y la variabilidad específica permanezcan separadas.

A continuación se presentan dos propiedades de la covarianza necesarias para demostrar dicha descomposición.

Lema 1 (Covarianza de la suma de vectores aleatorios[94])**.** Sean $\mathbf{U}$ y $\mathbf{V}$ dos vectores aleatorios. Entonces

$$\mathrm{Cov}(\mathbf{U} + \mathbf{V}) = \mathrm{Cov}(\mathbf{U}) + \mathrm{Cov}(\mathbf{V}) + \mathrm{Cov}(\mathbf{U}, \mathbf{V}) + \mathrm{Cov}(\mathbf{V}, \mathbf{U}).$$

Demostración. Sean $\mathbf{U}$ y $\mathbf{V}$ dos vectores aleatorios. Por definición de covarianza,

$$\mathrm{Cov}(\mathbf{U} + \mathbf{V}) = \mathbb{E}\left[(\mathbf{U} + \mathbf{V} - \mathbb{E}[\mathbf{U} + \mathbf{V}])(\mathbf{U} + \mathbf{V} - \mathbb{E}[\mathbf{U} + \mathbf{V}])^\top\right].$$

[90] ANDERSON. Op. cit.
[91] LAWLEY. Op. cit.
[92] HARMAN. Op. cit.
[93] JOHNSON, Richard A. Op. cit.
[94] ANDERSON. Op. cit.

Usando la linealidad de la esperanza y definiendo $\tilde{\mathbf{U}} = \mathbf{U} - \mathbb{E}[\mathbf{U}]$ y $\tilde{\mathbf{V}} = \mathbf{V} - \mathbb{E}[\mathbf{V}]$,

$$= \mathbb{E}\left[(\tilde{\mathbf{U}} + \tilde{\mathbf{V}})(\tilde{\mathbf{U}} + \tilde{\mathbf{V}})^\top\right].$$

Expandiendo el producto,

$$= \mathbb{E}\left[\tilde{\mathbf{U}}\tilde{\mathbf{U}}^\top + \tilde{\mathbf{U}}\tilde{\mathbf{V}}^\top + \tilde{\mathbf{V}}\tilde{\mathbf{U}}^\top + \tilde{\mathbf{V}}\tilde{\mathbf{V}}^\top\right].$$

Usando la linealidad de la esperanza,

$$= \mathbb{E}[\tilde{\mathbf{U}}\tilde{\mathbf{U}}^\top] + \mathbb{E}[\tilde{\mathbf{U}}\tilde{\mathbf{V}}^\top] + \mathbb{E}[\tilde{\mathbf{V}}\tilde{\mathbf{U}}^\top] + \mathbb{E}[\tilde{\mathbf{V}}\tilde{\mathbf{V}}^\top].$$

Reconociendo la definición de covarianza en cada término,

$$= \mathrm{Cov}(\mathbf{U}) + \mathrm{Cov}(\mathbf{U}, \mathbf{V}) + \mathrm{Cov}(\mathbf{V}, \mathbf{U}) + \mathrm{Cov}(\mathbf{V}).$$

□

Lema 2 (Covarianza de una transformación lineal[95])**.** Sea $\mathbf{U}$ un vector aleatorio y A una matriz constante. Entonces

$$\mathrm{Cov}(A\mathbf{U}) = A\,\mathrm{Cov}(\mathbf{U})\,A^\top.$$

Demostración. Por definición de covarianza,

$$\mathrm{Cov}(A\mathbf{U}) = \mathbb{E}\left[(A\mathbf{U} - \mathbb{E}[A\mathbf{U}])(A\mathbf{U} - \mathbb{E}[A\mathbf{U}])^\top\right].$$

Usando la linealidad de la esperanza y factorizando A,

$$= \mathbb{E}\left[A(\mathbf{U} - \mathbb{E}[\mathbf{U}])(\mathbf{U} - \mathbb{E}[\mathbf{U}])^\top A^\top\right].$$

Como A es constante, sale fuera de la esperanza,

$$= A\,\mathbb{E}\left[(\mathbf{U} - \mathbb{E}[\mathbf{U}])(\mathbf{U} - \mathbb{E}[\mathbf{U}])^\top\right] A^\top.$$

[95] Ibid.

Reconociendo la definición de covarianza,

$$= A \operatorname{Cov}(\mathbf{U}) A^{\top}.$$

□

Con estos resultados es posible demostrar la siguiente proposición.

Proposición 4. Bajo los supuestos del modelo factorial definido previamente, la matriz de covarianza de $\mathbf{X}$ está dada por

$$\Sigma = LL^{\top} + \Psi.$$

Demostración. La matriz de covarianza poblacional de un vector aleatorio $\mathbf{X}$ se define como

$$\Sigma = \operatorname{Cov}(\mathbf{X}).$$

Sustituyendo el modelo factorial en la definición anterior se obtiene

$$\Sigma = \operatorname{Cov}(L\mathbf{F} + \boldsymbol{\varepsilon}).$$

Usando el lema 1,

$$\Sigma = \operatorname{Cov}(L\mathbf{F}) + \operatorname{Cov}(\boldsymbol{\varepsilon}) + \operatorname{Cov}(L\mathbf{F}, \boldsymbol{\varepsilon}) + \operatorname{Cov}(\boldsymbol{\varepsilon}, L\mathbf{F}).$$

Ahora, usando el lema 2,

$$\operatorname{Cov}(L\mathbf{F}) = L \operatorname{Cov}(\mathbf{F}) L^{\top}.$$

Dado que el modelo establece que $\operatorname{Cov}(\mathbf{F}, \boldsymbol{\varepsilon}) = 0$, los términos cruzados se anulan, es decir,

$$\operatorname{Cov}(L\mathbf{F}, \boldsymbol{\varepsilon}) = 0, \qquad \operatorname{Cov}(\boldsymbol{\varepsilon}, L\mathbf{F}) = 0.$$

En consecuencia,

$$\Sigma = L \operatorname{Cov}(\mathbf{F}) L^{\top} + \operatorname{Cov}(\boldsymbol{\varepsilon}).$$

Como $\mathrm{Cov}(\mathbf{F}) = I_m$ y $\mathrm{Cov}(\boldsymbol{\varepsilon}) = \Psi$, se obtiene finalmente

$$\Sigma = LL^\top + \Psi.$$

□

Esta expresión muestra que el modelo factorial parametriza la matriz de covarianza poblacional mediante dos componentes: la variabilidad común $LL^\top$ y la variabilidad específica Ψ. En consecuencia, el problema central del análisis factorial consiste en estimar la matriz de cargas factoriales L, mientras que la matriz Ψ se determina simultáneamente a partir de la descomposición estructural de Σ [96] [97].

En la práctica, la matriz de covarianza poblacional Σ es desconocida, ya que para calcularla se necesitaría conocer todos los posibles valores que puede tomar $\mathbf{X}$, lo cual no es posible, pues en la realidad solo se dispone de una muestra finita de observaciones. Por esta razón, se trabaja con variables estandarizadas $\mathbf{Z}$, como se describe a continuación.

Sin pérdida de generalidad, es habitual estandarizar las variables a nivel poblacional mediante la transformación

$$Z_j = \frac{X_j - \mathbb{E}(X_j)}{\sqrt{\mathrm{Var}(X_j)}}, \quad j = 1, \ldots, p,$$

de modo que

$$\mathbb{E}(\mathbf{Z}) = \mathbf{0}, \qquad \mathrm{Var}(Z_j) = 1, \quad j = 1, \ldots, p.$$

Bajo esta parametrización, la matriz de covarianza poblacional coincide con la matriz de correlación de Pearson, la cual se denota por $\mathbf{R}$ [98]. Así, el modelo de análisis factorial queda expresado como

$$\mathbf{R} = LL^\top + \Psi.$$

[96] Ibid.
[97] LAWLEY. Op. cit.
[98] JOHNSON, Richard A. Op. cit.

Para comprender la naturaleza de esta aproximación, es necesario precisar las dimensiones de los elementos involucrados. Si se analizan p variables observadas y se postulan m factores comunes, (con $m < p$), entonces:

- La matriz de cargas factoriales L tiene dimensiones $p \times m$.
- El producto $LL^\top$ es una matriz simétrica de dimensión $p \times p$.
- La matriz de varianzas específicas Ψ es diagonal de tamaño $p \times p$.

Por tanto, la matriz $\mathbf{R} = LL^\top + \Psi$ es de dimensión $p \times p$, lo cual es consistente con las dimensiones de la matriz de correlación de Pearson $\mathbf{R}$ [99] [100].

3.4.1.1. Estimación de los parámetros de las matrices L y Ψ. Sea $\mathbf{x}_1, \ldots, \mathbf{x}_n$ una muestra aleatoria independiente del vector aleatorio $\mathbf{X} \in \mathbb{R}^p$. Como se mencionó anteriormente, en la práctica, la matriz poblacional $\mathbf{R}$ es desconocida. Dado que las variables han sido estandarizadas, la matriz de correlación muestral puede expresarse como

$$\hat{\mathbf{R}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{x}_i \mathbf{x}_i^\top,$$

El objetivo es estimar los parámetros de L y Ψ de modo que la matriz modelo $LL^\top + \Psi$ reproduzca adecuadamente la matriz de correlación poblacional $\mathbf{R}$, utilizando para ello la información contenida en $\hat{\mathbf{R}}$.

Supuesto de normalidad: Para estimar los parámetros mediante máxima verosimilitud, se asume que el vector aleatorio sigue una distribución normal multivariada con vector de medias $\mathbf{0}$ y matriz de covarianza dada por la estructura factorial, es decir,

$$\mathbf{x}_i \sim \mathcal{N}_p(\mathbf{0}, \mathbf{R}), \quad i = 1, \ldots, n,$$

[99] ANDERSON. Op. cit.
[100] LAWLEY. Op. cit.

donde $\mathbf{R} = LL^\top + \Psi$ [101].

Función de verosimilitud: Bajo el supuesto de normalidad multivariada, dado que $\boldsymbol{\mu} = 0$ por la estandarización previa de las variables, la densidad de $\mathbf{x}_i$ viene dada por [102]:

$$f(\mathbf{x}_i) = \frac{1}{(2\pi)^{p/2}|\mathbf{R}|^{1/2}} \exp\left(-\frac{1}{2}\mathbf{x}_i^\top \mathbf{R}^{-1}\mathbf{x}_i\right).$$

Dado que las observaciones son independientes, la función de verosimilitud para las n observaciones está dada por [103]:

$$\mathcal{L}(L, \Psi) = \frac{1}{(2\pi)^{np/2}|\mathbf{R}|^{n/2}} \exp\left(-\frac{1}{2}\sum_{i=1}^{n}\mathbf{x}_i^\top \mathbf{R}^{-1}\mathbf{x}_i\right).$$

El término dentro del exponente puede reescribirse usando la definición de $\hat{\mathbf{R}}$

$$\sum_{i=1}^{n}\mathbf{x}_i^\top \mathbf{R}^{-1}\mathbf{x}_i = n\,\mathrm{tr}\left(\mathbf{R}^{-1}\hat{\mathbf{R}}\right),$$

donde $\mathrm{tr}(\cdot)$ denota la traza de una matriz, definida como la suma de sus elementos diagonales [104]. En consecuencia, la función de verosimilitud queda expresada como [105] [106]:

$$\mathcal{L}(L, \Psi) = \frac{1}{(2\pi)^{np/2}|\mathbf{R}|^{n/2}} \exp\left(-\frac{n}{2}\mathrm{tr}\left(\mathbf{R}^{-1}\hat{\mathbf{R}}\right)\right).$$

[101] ANDERSON. Op. cit.
[102] Ibid.
[103] Ibid.
[104] Ibid.
[105] Ibid.
[106] LAWLEY. Op. cit.

Log-verosimilitud: Para simplificar la optimización se trabaja con el logaritmo natural de la verosimilitud. Tomando logaritmos se obtiene [107]

$$\ell(L, \Psi) = -\frac{np}{2}\log(2\pi) - \frac{n}{2}\log|\mathbf{R}| - \frac{n}{2}\mathrm{tr}\left(\mathbf{R}^{-1}\hat{\mathbf{R}}\right).$$

El primer término no depende de los parámetros L y Ψ, por lo que puede omitirse sin afectar el proceso de maximización. Así, la log-verosimilitud relevante es [108] [109]:

$$\ell(L, \Psi) = -\frac{n}{2}\left[\log|\mathbf{R}| + \mathrm{tr}\left(\mathbf{R}^{-1}\hat{\mathbf{R}}\right)\right].$$

Recordando que $\mathbf{R} = LL^\top + \Psi$, la log-verosimilitud depende únicamente de L y Ψ.

Función de discrepancia Maximizar la log-verosimilitud es equivalente a minimizar la función [110] [111]

$$F(L, \Psi) = \log|LL^\top + \Psi| + \mathrm{tr}\left((LL^\top + \Psi)^{-1}\hat{\mathbf{R}}\right).$$

Es habitual añadir los términos $-\log|\hat{\mathbf{R}}| - p$, que no dependen de los parámetros, para obtener la siguiente función denominada función de discrepancia [112] [113]:

$$F(L, \Psi) = \log|LL^\top + \Psi| + \mathrm{tr}\left((LL^\top + \Psi)^{-1}\hat{\mathbf{R}}\right) - \log|\hat{\mathbf{R}}| - p.$$

La expresión anterior define la función que se minimiza para obtener los estimadores de máxima verosimilitud de L y Ψ [114] [115] [116].

[107] ANDERSON. Op. cit.
[108] Ibid.
[109] LAWLEY. Op. cit.
[110] JÖRESKOG, Karl G. Some Contributions to Maximum Likelihood Factor Analysis. En: *Psychometrika*. 1967, vol. 32, nro. 4, pp. 443-482.
[111] LAWLEY. Op. cit.
[112] JÖRESKOG. Op. cit.
[113] LAWLEY. Op. cit.
[114] ANDERSON. Op. cit.
[115] JÖRESKOG. Op. cit.
[116] LAWLEY. Op. cit.

Intuitivamente, esta función compara dos matrices:

- la matriz de correlación observada en la muestra, $\hat{\mathbf{R}}$,
- y la matriz de correlación que el modelo factorial predice, $LL^\top + \Psi$.

Si ambas matrices fueran exactamente iguales, el modelo reproduciría perfectamente las correlaciones observadas. En ese caso, el valor de la función sería cero.

En general, las dos matrices no coinciden exactamente. La función $F(L, \Psi)$ cuantifica qué tan distintas son. Cuanto menor es su valor, más cerca está la matriz del modelo de la matriz observada.

Por esta razón, estimar el modelo por máxima verosimilitud consiste simplemente en buscar los valores de L y Ψ que hacen lo más pequeño posible el valor de $F(L, \Psi)$ [117] [118]. Es decir ,

$$(\hat{L}, \hat{\Psi}) = \arg\min_{L,\Psi} F(L, \Psi).$$

En la práctica, este problema no puede resolverse mediante una fórmula explícita, ya que las ecuaciones resultantes son no lineales. Por ello se emplean algoritmos numéricos iterativos, como el método de Newton–Raphson, entre otros [119] [120].

3.4.1.2. Reducción de dimensión. El modelo poblacional del análisis factorial establece que

$$\mathbf{X} = L\mathbf{F} + \boldsymbol{\varepsilon}.$$

Además, se conoce que en el modelo teórico, las matrices L y Ψ son parámetros desconocidos, ya que determinan la estructura de la matriz de correlación poblacional $\mathbf{R}$ y que dicha matriz no es observable y debe estimarse a partir de una muestra. Por esta

[117] ANDERSON. Op. cit.
[118] LAWLEY. Op. cit.
[119] ANDERSON. Op. cit.
[120] JÖRESKOG. Op. cit.

razón, fue necesario obtener los estimadores $\hat{L}$ y $\hat{\Psi}$ mediante máxima verosimilitud, como se desarrolló en la subsección anterior.

Una vez obtenidos los estimadores $\hat{L}$ y $\hat{\Psi}$ en el contexto muestral, el modelo factorial permite construir, para cada observación, una representación en términos de los factores comunes. Esta representación implica la reducción de dimensión proporcionada por el modelo.

Sea $\mathbf{x}_i \in \mathbb{R}^{p\times 1}$ la observación correspondiente al individuo i. Bajo el supuesto de normalidad multivariante, el estimador más utilizado para los factores es el estimador de regresión, definido como la esperanza condicional de $\mathbf{F}$ dado $\mathbf{X} = \mathbf{x}_i$ [121] [122]:

$$\hat{\mathbf{f}}_i = (\hat{L}^\top \hat{\Psi}^{-1} \hat{L})^{-1} \hat{L}^\top \hat{\Psi}^{-1} \mathbf{x}_i, \qquad \hat{\mathbf{f}}_i \in \mathbb{R}^{m\times 1}.$$

Este estimador es una combinación lineal de las variables observadas y depende únicamente de las estimaciones de L y Ψ.

Si se dispone de una muestra de tamaño n, formada por observaciones independientes $\mathbf{x}_1, \ldots, \mathbf{x}_n$, estas pueden organizarse en la matriz de datos

$$\mathbf{X}_n = \begin{pmatrix} \mathbf{x}_1^\top \\ \vdots \\ \mathbf{x}_n^\top \end{pmatrix} \in \mathbb{R}^{n\times p}.$$

Agrupando los estimadores individuales, los puntajes factoriales para toda la muestra se expresan en forma matricial como

$$\hat{\mathbf{F}} = \mathbf{X}_n \hat{\Psi}^{-1} \hat{L} (\hat{L}^\top \hat{\Psi}^{-1} \hat{L})^{-1}, \qquad \hat{\mathbf{F}} \in \mathbb{R}^{n\times m}.$$

Cada fila de $\hat{\mathbf{F}}$ contiene el vector de dimensión m que representa a un individuo en el espacio factorial. La matriz original de datos $\mathbf{X}_n \in \mathbb{R}^{n\times p}$ es reemplazada por $\hat{\mathbf{F}} \in \mathbb{R}^{n\times m}$, con $m < p$, lo que constituye la reducción de dimensión proporcionada por el modelo

[121] ANDERSON. Op. cit.
[122] LAWLEY. Op. cit.

factorial.

El método de máxima verosimilitud fue seleccionado para estimar los parámetros del modelo factorial por tres razones principales.

- Métodos como mínimos cuadrados ordinarios buscan minimizar la diferencia entre la matriz de correlación observada y la matriz del modelo, sin asumir ninguna distribución sobre los datos. Esto impide hacer afirmaciones formales sobre qué tan buenas son las estimaciones. La máxima verosimilitud, en cambio, al incorporar el supuesto de normalidad multivariada, garantiza que a medida que se dispone de más datos las estimaciones se acercan cada vez más a los valores reales[123].
- Permite determinar si el número de factores escogido es adecuado comparando, mediante la razón de verosimilitud, qué tan bien ajusta el modelo con m factores frente a un modelo sin restricciones. Si la diferencia es pequeña, se concluye que m factores es suficiente para explicar la estructura de correlación de los datos [124].
- A diferencia de métodos como mínimos cuadrados ordinarios cuyos resultados pueden variar según si las variables están estandarizadas o no, la máxima verosimilitud produce resultados que no dependen de la escala de las variables, lo que lo hace especialmente apropiado cuando, como en este caso, las variables han sido previamente estandarizadas [125].

3.4.2. Rotación de factores. En el modelo factorial se cumple que $\mathbf{R} = LL^\top + \Psi$.

Sin embargo, la matriz de cargas no es única. Sea Q una matriz ortogonal, es decir, una matriz que satisface $Q^\top Q = I$, entonces

$$LL^\top = (LQ)(LQ)^\top.$$

[123]ANDERSON. Op. cit.
[124]LAWLEY. Op. cit.
[125]JOHNSON, Richard A. Op. cit.

Por lo tanto, distintas matrices de cargas pueden reproducir exactamente la misma matriz de correlación. En la práctica, al estimar el modelo a partir de los datos se obtiene una matriz de cargas estimada $\hat{L}$, pero esta no es la única posible.

La rotación de factores consiste en transformar la matriz estimada $\hat{L}$ mediante una matriz ortogonal Q,

$$\tilde{L} = \hat{L}Q,$$

con el objetivo de obtener una estructura de cargas en la que cada variable esté fuertemente asociada a un solo factor y débilmente a los demás. Cuando esto ocurre, es posible identificar qué grupo de variables comparte cada factor y asignarle un significado concreto, lo que hace la solución factorial más comprensible. Esta transformación no modifica el ajuste del modelo ni la matriz de correlación reproducida; únicamente cambia la forma en que las variables quedan asociadas a cada factor [126] [127].

Rotación ortogonal. En la rotación ortogonal se mantiene la condición $Q^\top Q = I$, lo que implica que los factores permanecen incorrelacionados. La nueva matriz de cargas se obtiene como

$$\tilde{L} = \hat{L}Q.$$

Uno de los métodos más utilizados es la rotación Varimax, propuesta por Henry F. Kaiser en 1958 [128]. Sea $\hat{l}_{ij}$ el elemento ubicado en la fila i y columna j de la matriz $\hat{L}$. El criterio Varimax consiste en maximizar

$$V = \sum_{j=1}^{m} \left[\frac{1}{p} \sum_{i=1}^{p} \hat{l}_{ij}^4 - \left(\frac{1}{p} \sum_{i=1}^{p} \hat{l}_{ij}^2 \right)^2 \right].$$

Este criterio favorece que, dentro de cada factor, algunas cargas sean grandes y las demás pequeñas. En consecuencia, cada factor queda definido principalmente por un

[126] HARMAN. Op. cit.
[127] JOHNSON, Richard A. Op. cit.
[128] KAISER, Henry F. The varimax criterion for analytic rotation in factor analysis. En: *Psychometrika*. 1958, vol. 23, pp. 187-200.

conjunto reducido de variables, lo que facilita su interpretación [129] [130].

Rotación oblicua. En la rotación oblicua no se exige que Q sea ortogonal. En este caso,

$$\tilde{L} = \hat{L}Q,$$

pero $Q^\top Q \neq I$. La matriz de correlación puede escribirse como

$$\mathbf{R} = \tilde{L}\Phi\tilde{L}^\top + \Psi,$$

donde

$$\Phi = Q^{-1}(Q^{-1})^\top.$$

Aquí los factores pueden estar correlacionados, ya que

$$\mathrm{Cov}(\mathbf{F}) = \Phi \neq I.$$

Permitir esta correlación puede ser conveniente cuando se considera razonable que los factores estén relacionados entre sí. Entre los métodos oblicuos más utilizados se encuentran Promax y Oblimin [131] [132] [133]. La rotación factorial no cambia la calidad del ajuste del modelo ni la matriz de correlación reproducida. Su función es proporcionar una representación más clara de la estructura factorial estimada. La elección entre rotación ortogonal u oblicua depende de si se desea mantener factores independientes o permitir que estén correlacionados.

Método de rotación seleccionado: En este trabajo se utilizó la rotación Varimax. Esta elección se debe a que, a diferencia de las rotaciones oblicuas que permiten que los factores estén relacionados entre sí, la rotación Varimax mantiene los factores independientes unos de otros. Esto es apropiado en este caso, ya que el modelo

[129] ANDERSON. Op. cit.
[130] HARMAN. Op. cit.
[131] ANDERSON. Op. cit.
[132] HARMAN. Op. cit.
[133] LAWLEY. Op. cit.

factorial utilizado asume desde el principio que los factores no están relacionados entre sí [134]. Además, al mantener esta independencia, las cargas factoriales resultantes son más fáciles de interpretar, pues cada variable queda asociada de forma clara a un solo factor sin que la presencia de otros factores interfiera en esa asociación [135] [136].

3.5. MODELIZACIÓN CON ANÁLISIS FACTORIAL

Así, como en el análisis de componentes principales, la elección del número de factores influye directamente en la calidad de la reducción dimensional. Un número demasiado pequeño puede implicar la pérdida de información relevante, mientras que un número excesivo puede introducir complejidad innecesaria y dificultar la interpretación de la estructura de los datos. Antes de determinar este número, es necesario verificar que las variables presentan correlaciones suficientes para que la aplicación del AF esté justificada.

3.5.1. Índice KMO. Para verificar que las variables presentan correlaciones suficientes entre sí, condición necesaria para que la aplicación del AF esté justificada, se utiliza el índice Kaiser-Meyer-Olkin (KMO) [137], el cual compara, para cada par de variables, la correlación observada con su correlación parcial. Las correlaciones observadas corresponden a las entradas r_{ij} de la matriz de correlación muestral $\hat{\mathbf{R}}$, definida en la Sección 3.1. Las correlaciones parciales, por su parte, se obtienen a partir de la inversa de la matriz de correlación muestral $\hat{\mathbf{R}}^{-1}$. Si se denota por q_{ij} al elemento ubicado en la fila i y columna j de $\hat{\mathbf{R}}^{-1}$, la correlación parcial entre X_i y X_j se define como

$$u_{ij} = -\frac{q_{ij}}{\sqrt{q_{ii}\, q_{jj}}}.$$

Esta cantidad mide la relación entre X_i y X_j descartando la influencia que las demás variables puedan tener sobre ellas. Si esta correlación parcial es pequeña, significa

[134] ANDERSON. Op. cit.
[135] HARMAN. Op. cit.
[136] KAISER, Henry F. Op. cit.
[137] KAISER, Henry F. A second generation little jiffy. En: *Psychometrika*. 1970, vol. 35, nro. 4, pp. 401-415.

que la relación entre X_i y X_j está bien explicada por las demás variables del conjunto, lo cual es precisamente lo que se espera cuando existen factores comunes que estructuran la información compartida.

Formalmente, el índice KMO se define como

$$\text{KMO} = \frac{\sum_{i \neq j} r_{ij}^2}{\sum_{i \neq j} r_{ij}^2 + \sum_{i \neq j} u_{ij}^2},$$

donde r_{ij} y u_{ij} son las correlaciones observadas y parciales definidas anteriormente [138].

El índice toma valores en el intervalo $[0, 1]$. Cuando las correlaciones parciales son pequeñas respecto a las correlaciones observadas, el índice se acerca a 1, indicando que las variables comparten suficiente variabilidad común y que el AF es apropiado. Valores cercanos a 0, en cambio, sugieren que las variables no están suficientemente relacionadas entre sí y que la aplicación del AF no estaría justificada. Como referencia, valores superiores a $0{,}7$ se consideran aceptables para aplicar el análisis factorial [139].

3.5.2. Criterios de selección del número de factores. Una vez confirmado que las variables presentan correlaciones suficientes, es necesario determinar el número de factores m que se utilizará en el modelo. Este valor debe fijarse antes de estimar los parámetros por máxima verosimilitud, ya que la estructura del modelo depende directamente de él. Para determinarlo, se emplearon los mismos criterios estadísticos utilizados en la modelización mediante PCA: el criterio de Yeomans-Golder y el análisis paralelo de Horn, como fueron descritos en la Subsección 3.3.1. Ambos criterios se aplican sobre los autovalores de la matriz de correlación muestral $\hat{\mathbf{R}}$, y el número de factores m queda determinado por el número de autovalores que superan el umbral establecido por cada criterio. Sus definiciones formales se encuentran en la Subsección 3.3.1.

[138] Ibid.
[139] HAIR, J. et al. *Multivariate Data Analysis*. Cengage, 2019.

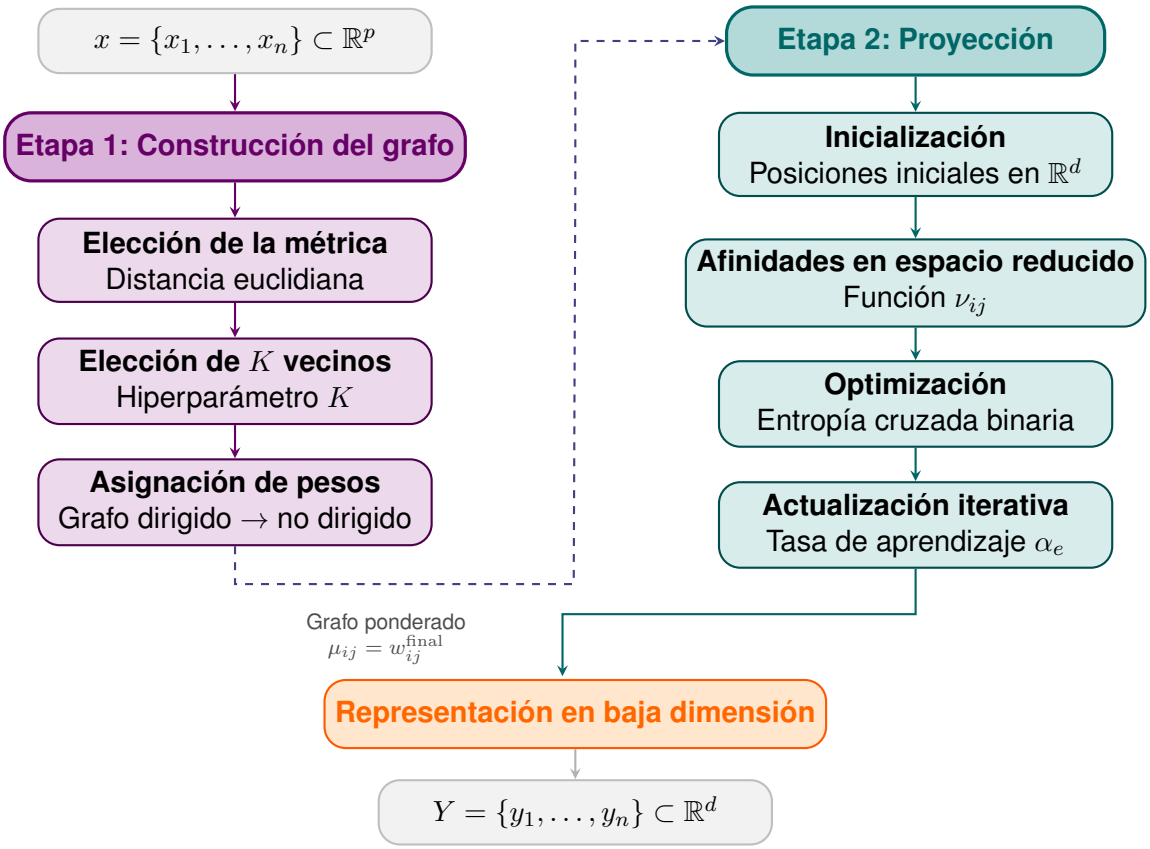
3.6. APROXIMACIÓN Y PROYECCIÓN UNIFORME DE VARIEDADES

El método de aproximación y proyección uniforme de variedades (Uniform Manifold Approximation and Projection (UMAP)) es una técnica no lineal cuyo objetivo es construir una representación de baja dimensión que preserve la estructura local de un conjunto de datos de alta dimensión [140]. UMAP no reproduce exactamente las distancias entre todos los puntos, sino que prioriza las relaciones de vecindad más significativas [141]. Esto significa que, si dos observaciones presentan una fuerte relación en el espacio-original, por ejemplo, porque pertenecen al mismo entorno cercano, el método busca que dicha relación se refleje también en el espacio reducido. Para formalizar estas relaciones, UMAP utiliza un grafo no dirigido, es decir, una estructura compuesta por un conjunto de vértices (que representan las observaciones) y un conjunto de aristas (que representan relaciones entre ellas). A cada arista se le asigna un peso que cuantifica la intensidad de la relación de vecindad entre dos puntos, obteniéndose así un grafo ponderado [142]. El procedimiento puede describirse en dos etapas principales: la construcción de un grafo de vecindades en el espacio original y la obtención de una representación de baja dimensión que preserve, en términos de dichos pesos la estructura local capturada inicialmente [143].

Etapas para la reducción de dimensionalidad con UMAP

[140] McINNES. Op. cit.
[141] Ibid.
[142] Ibid.
[143] Ibid.

Figura 1. Etapas del método UMAP.



Fuente: Elaboración propia .

3.6.1. Construcción del grafo de vecindades en el espacio original. Sea $\mathbf{x} = \{x_1, \ldots, x_n\}$ el conjunto de datos de entrada, donde cada x_i representa una imagen previamente vectorizada y estandarizada. Cada imagen está compuesta por píxeles, que son las unidades más pequeñas en las que se divide una imagen digital y a cada uno de los cuales se le asigna un valor numérico que representa su intensidad de color. Desde el punto de vista geométrico, cada observación se modela como un punto del espacio euclidiano de dimensión p, es decir, $x_i \in \mathbb{R}^p$, donde p representa el número de píxeles de la imagen.

La estandarización previa de las variables desempeña un papel fundamental, ya que las distancias entre observaciones dependen de la escala de las variables. Al estandarizar las variables de entrada, se evita que ciertas dimensiones tengan mayor influencia que otras en el cálculo de proximidades.

Para capturar la estructura local del conjunto de datos, UMAP transforma el conjunto

de puntos en $\mathbb{R}^p$ en un grafo ponderado cuya estructura se detalla posteriormente. En este contexto, la estructura local hace referencia a las relaciones de vecindad entre observaciones, entendidas como la proximidad entre puntos dentro del espacio original: dos observaciones son vecinas si la distancia entre ellas es pequeña en comparación con el resto del conjunto. El método no depende de la posición absoluta de los puntos en el espacio, sino únicamente de las distancias entre ellos.

En la representación mediante grafo, cada observación x_i se asocia a un vértice, y las aristas entre vértices representan las relaciones de vecindad determinadas en el espacio original. Por esta razón, a lo largo de este trabajo los términos observación y vértice se utilizarán para referirse al mismo elemento, indistintamente [144].

3.6.1.1. Elección de la métrica. Para cuantificar la cercanía entre dos observaciones, UMAP se formula en términos de una métrica general $d : \mathbb{R}^p \times \mathbb{R}^p \longrightarrow \mathbb{R}^+$, la cual asigna a cada par de puntos un valor numérico que representa la distancia entre ellos. A partir de esta distancia se definen las vecindades: dos observaciones se consideran vecinas si la distancia entre ellas es suficientemente pequeña [145]. Esta formulación no restringe la elección de la métrica, permitiendo adoptar distintas métricas para la distancia según la naturaleza de los datos y el tipo de estructura que se desee preservar.

Entre las métricas más utilizadas, se encuentran:

- **Distancia euclidiana** que mide la distancia geométrica directa entre dos puntos en un espacio vectorial. Es apropiada cuando las variables son numéricas continuas y están medidas en escalas similares, de modo que ninguna componente domine artificialmente el cálculo de la distancia. Se utiliza cuando interesa cuantificar la separación global entre puntos, considerando simultáneamente todas sus coordenadas [146].
- **Distancia Manhattan** definida como la suma de las diferencias absolutas entre

[144] Ibid.
[145] Ibid.
[146] Ibid.

las componentes de los vectores. Puede ser más adecuada cuando los datos presentan valores atípicos o cuando las diferencias grandes en una sola componente no deberían influir de manera desproporcionada en la distancia total. También es útil en espacios de alta dimensión o con datos dispersos [147].

- **Distancia del coseno** evalúa la similitud entre dos vectores a partir del ángulo que forman, independientemente de su magnitud. Es especialmente conveniente cuando interesa comparar la dirección o patrón de los vectores más que su tamaño, es decir, cuando dos observaciones se consideran similares si sus valores crecen y decrecen de forma proporcional, aunque sus magnitudes sean distintas [148].

- **Distancia de Mahalanobis** incorpora la estructura de correlación entre las variables mediante la inversa de la matriz de covarianzas. Es adecuada cuando las variables están correlacionadas, ya que ajusta la distancia teniendo en cuenta cómo varían las variables juntas, evitando sobreestimar diferencias en direcciones donde los datos presentan alta dispersión [149].

En este trabajo se adopta como métrica la distancia Euclidiana, ya que cada observación x_i se representa como un vector de valores numéricos continuos previamente estandarizados, lo que permite cuantificar de manera directa la proximidad entre pares de observaciones.

En consecuencia, para dos puntos $x_i, x_j \in \mathbb{R}^p$, con

$$x_i = (x_{i1}, x_{i2}, \ldots, x_{ip})^T \text{ y } x_j = (x_{j1}, x_{j2}, \ldots, x_{jp})^T$$

se define como:

$$d(x_i, x_j) = \sqrt{\sum_{l=1}^{p} (x_{il} - x_{jl})^2}.$$

[147] Ibid.
[148] McINNES. Op. cit.
[149] ANDERSON. Op. cit.

3.6.1.2. Elección de los K vecinos. Con la métrica $d(\cdot, \cdot)$ en $\mathbb{R}^p$ definida, el siguiente paso es identificar, para cada observación x_i, los puntos que se encuentran más cercanos a ella en el espacio original. Para ello se fija un entero positivo K, que corresponde a un hiperparámetro del método, es decir, un valor que debe establecerse antes de ejecutar el algoritmo y que no se aprende de los datos. Este hiperparámetro indica cuántos vecinos se considerarán alrededor de cada punto y controla el equilibrio entre la preservación de la estructura local y global de los datos: valores pequeños de K capturan relaciones muy cercanas, mientras que valores grandes incluyen información de puntos más distantes. Así, K controla el grado de localidad con el que se modela la estructura del conjunto de datos [150].

Dado x_i, se ordenan las restantes observaciones $\{x_j : j \neq i\}$ de manera no decreciente según la distancia $d(x_i, x_j)$.

Se define entonces el conjunto de los K vecinos más cercanos como

$$\mathcal{N}^{(K)}(x_i) = \{x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(K)}\},$$

donde $x_i^{(k)}$ denota la k-ésima observación más próxima a x_i según dicho ordenamiento, para $k = 1, \ldots, K$.

En caso de que varios puntos presenten la misma distancia, el ordenamiento previo permite seleccionar exactamente los primeros K elementos. Es decir, una vez ordenadas las observaciones de manera no decreciente según la distancia a x_i, el conjunto de vecinos queda definido como los primeros K puntos de dicho orden.

3.6.1.3. Asignación de pesos y creación del grafo. Una vez identificado el conjunto de vecinos $\mathcal{N}^{(K)}(x_i)$ para cada observación x_i, el siguiente paso es asignar un valor numérico a cada arista, denominado peso, que cuantifica la intensidad de la relación entre dos vértices. Valores altos se interpretan como una fuerte afinidad correspondiente a puntos muy próximos en el espacio original, mientras que valores bajos se interpretan como una relación más débil asociada a mayores distancias.

[150] McINNES. Op. cit.

El proceso se da en dos etapas. En la primera etapa, se construye un grafo dirigido en el cual cada vértice x_i se conecta mediante aristas orientadas hacia cada uno de sus vecinos. En la segunda etapa, este grafo dirigido se transforma en un grafo no dirigido, combinando las afinidades en ambas direcciones con el fin de obtener una representación simétrica de las relaciones.

Grafo dirigido En este contexto, la afinidad entre dos observaciones es un valor en el intervalo $(0, 1]$ que cuantifica qué tan cerca están una de otra: un valor cercano a 1 indica que las observaciones son muy próximas, mientras que un valor cercano a 0 indica que están muy alejadas.

Para cada observación x_i y cada vecino $x_i^{(k)} \in \mathcal{N}^{(K)}(x_i)$, con $k = 1, \ldots, K$, se define el peso dirigido

$$w_{i,x_i^{(k)}} = \exp\left(-\frac{\max\{0,\, d(x_i,\, x_i^{(k)}) - \rho_i\}}{\sigma_i}\right).$$

Para observaciones que no pertenecen a la vecindad de x_i, se define $w_{i,x_i^{(k)}} = 0$.

La expresión anterior transforma la distancia geométrica en una afinidad acotada en $(0, 1]$. La función exponencial garantiza un decrecimiento suave del peso conforme aumenta la distancia, preservando la estructura local sin introducir discontinuidades [151].

El término $\max\{0,\, d(x_i, x_i^{(k)}) - \rho_i\}$ introduce un desplazamiento local que evita penalizar distancias extremadamente pequeñas, de modo que los vecinos más próximos reciben pesos cercanos a uno. En consecuencia, valores altos del peso indican una relación de vecindad fuerte, mientras que valores pequeños reflejan una contribución débil a la estructura local en torno a x_i [152].

Los parámetros que intervienen en la expresión se definen de la siguiente manera:

- El parámetro ρ_i corresponde a la distancia desde x_i hasta su vecino más cercano

[151] Ibid.
[152] Ibid.

con distancia estrictamente positiva. Se toma el mínimo para garantizar que la afinidad con el vecino más cercano sea siempre igual a 1, asegurando que cada punto tenga al menos una conexión fuerte en el grafo [153], se calcula de la siguiente manera:

$$\rho_i = \min\{d(x_i, x_i^{(k)}) \ : \ x_i^{(k)} \in \mathcal{N}^{(K)}(x_i),\ d(x_i, x_i^{(k)}) > 0\}$$

- El parámetro $\sigma_i > 0$ actúa como un factor de escala local y se determina numéricamente imponiendo la condición[154]

$$\sum_{k=1}^{K} \exp\left(-\frac{\max\{0,\ d(x_i, x_i^{(k)}) - \rho_i\}}{\sigma_i}\right) = \log_2(K).$$

  Esta condición garantiza que la suma de las afinidades dentro de la vecindad de x_i sea siempre igual a $\log_2(K)$, es decir,

$$\sum_{k=1}^{K} w_{ik} = \log_2(K),$$

  controlando así cuántos vecinos contribuyen de forma significativa a la representación local de cada punto. Si σ_i fuese demasiado pequeño, los pesos se concentrarían en muy pocos vecinos; si fuese demasiado grande, la afinidad se distribuiría casi uniformemente entre todos. De este modo, σ_i se calcula automáticamente según qué tan concentrados o dispersos están los puntos en el entorno de x_i: en zonas donde los puntos están muy juntos toma valores pequeños, mientras que en zonas más dispersas aumenta, permitiendo una representación equilibrada de la estructura local.

Para cada x_i, los pesos definidos anteriormente definen K aristas salientes hacia los elementos de su vecindad $\mathcal{N}^{(K)}(x_i)$. Cada vecino $x_i^{(k)}$ pertenece al conjunto global $\mathbf{x} = \{x_1, \ldots, x_n\}$, por lo que existe un índice j tal que $x_i^{(k)} = x_j$. En consecuencia, las aristas locales pueden interpretarse globalmente como pares ordenados (x_i, x_j), con $x_j \in \mathcal{N}^{(K)}(x_i)$. Al reunir las aristas generadas para todos los índices $i = 1, \ldots, n$, se

[153] Ibid.
[154] Ibid.

obtiene una colección global de relaciones entre elementos de $\mathbf{x}$, lo que da lugar a una estructura que puede modelarse como un grafo dirigido ponderado[155].

El peso w_{ij} depende de los parámetros locales ρ_i y σ_i, determinados a partir de la vecindad de x_i. De manera análoga, w_{ji} se calcula utilizando ρ_j y σ_j. Como, en general, $\rho_i \neq \rho_j$ y $\sigma_i \neq \sigma_j$, no se tiene necesariamente que $w_{ij} = w_{ji}$.

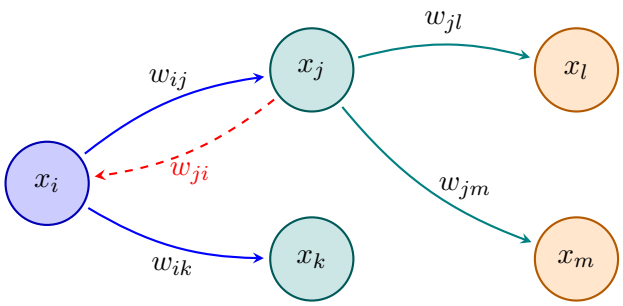
La estructura obtenida en esta etapa puede formalizarse como un grafo dirigido ponderado

$$\bar{G} = (\mathcal{V}, \mathcal{E}, \omega),$$

donde:

- $\mathcal{V} = \mathbf{x} = \{x_1, \ldots, x_n\}$ es el conjunto de vértices;
- $\mathcal{E} = \{(x_i, x_j) : x_j \in \mathcal{N}^{(K)}(x_i),\ i = 1, \ldots, n\}$ es el conjunto de aristas dirigidas;
- $\omega : V \times V \to [0, 1]$ es la función de pesos definida anteriormente, donde $w_{ij} > 0$ únicamente si $x_j \in \mathcal{N}^{(K)}(x_i)$.

Figura 2. Grafo dirigido ponderado.



Fuente: Elaboración propia.

Cada punto genera K aristas salientes hacia su vecindad local. La arista punteada ilustra la asimetría de las afinidades, donde $w_{ji} \neq w_{ij}$ en general.

[155] Ibid.

Transformación a grafo no dirigido El grafo dirigido describe la estructura local desde la perspectiva de cada observación, por lo que las afinidades no son necesariamente recíprocas; es decir, la afinidad de x_i hacia x_j puede ser distinta a la de x_j hacia x_i.

Sin embargo, en el espacio reducido la cercanía entre dos puntos se mide con una única distancia, que por definición de espacio métrico es la misma sin importar desde cuál de los dos puntos se calcule [156] [157]. Para que los pesos del grafo sean comparables con esas distancias, es necesario que también exista un único valor por par de puntos. Por ello, UMAP transforma el grafo dirigido en un grafo no dirigido mediante una combinación simétrica de las afinidades en ambas direcciones, obteniendo un único peso por par de puntos [158].

Finalmente, el grafo dirigido se transforma en un grafo no dirigido mediante una regla de combinación simétrica de las afinidades en ambas direcciones [159].

Formalmente, dicha combinación se define como sigue.

Proposición 5. Sea w_{ij}^{final} el peso final entre x_i y x_j, definido por

$$w_{ij}^{\text{final}} = w_{ij} + w_{ji} - w_{ij}w_{ji},$$

donde $0 \leq w_{ij}, w_{ji} \leq 1$. Entonces

$$0 \leq w_{ij}^{\text{final}} \leq 1.$$

Demostración. Sea $0 \leq w_{ij}, w_{ji} \leq 1$. Entonces

$$0 \leq 1 - w_{ij} \leq 1 \quad \text{y} \quad 0 \leq 1 - w_{ji} \leq 1.$$

Como el producto de dos números en el intervalo $[0, 1]$ también pertenece a $[0, 1]$, se

[156] MUNKRES, James R. *Topología*. 2 ed. Prentice Hall, 2002.
[157] RUDIN, Walter. *Principios de Análisis Matemático*. 3 ed. McGraw-Hill, 1976.
[158] McINNES. Op. cit.
[159] Ibid.

sigue que

$$0 \leq (1 - w_{ij})(1 - w_{ji}) \leq 1.$$

Por definición,

$$w_{ij}^{\text{final}} = 1 - (1 - w_{ij})(1 - w_{ji}).$$

Restar a 1 un número que pertenece al intervalo $[0, 1]$ produce nuevamente un número en $[0, 1]$. Por lo tanto,

$$0 \leq w_{ij}^{\text{final}} \leq 1.$$

□

El resultado es un grafo no dirigido ponderado que resume la estructura local del conjunto de datos. Los pesos w_{ij}^{final} de este grafo servirán como afinidades μ_{ij} en la etapa siguiente, donde UMAP buscará posiciones en el espacio reducido que reproduzcan fielmente esas relaciones de vecindad, garantizando que puntos con alta afinidad en el grafo queden próximos en la representación final[160].

3.6.2. Proyección en el espacio reducido. Una vez construido el grafo ponderado en el espacio original, cuya afinidad entre los vértices i y j está dada por $\mu_{ij} = w_{ij}^{\text{final}}$, el siguiente objetivo de UMAP es encontrar una representación de baja dimensión

$$Y = \{y_1, \ldots, y_n\} \subset \mathbb{R}^d,$$

donde cada y_i representa la imagen del punto x_i en el espacio reducido, es decir, ambos representan la misma observación pero en espacios de distinta dimensión: x_i en el espacio original y y_i en el espacio reducido[161]. El problema consiste en determinar las posiciones $\{y_i\}_{i=1}^n$ de manera que las relaciones de afinidad del grafo original queden fielmente representadas en el espacio reducido.

3.6.2.1. Inicialización. Dado que UMAP trabaja ajustando las posiciones de los puntos en el espacio reducido, cada uno de los n puntos debe ocupar algún lugar en dicho

[160] Ibid.
[161] Ibid.

espacio antes de que el proceso de ajuste pueda comenzar.

Esto puede hacerse de forma aleatoria o aprovechando la estructura del grafo construido en la fase anterior mediante un método basado en las propiedades algebraicas de dicho grafo, con el fin de obtener una configuración de partida que ya refleje parcialmente la organización global de los datos. Los detalles de este procedimiento pueden consultarse en Uniform Manifold Approximation and Projection for Dimension Reduction [162]. A partir de esta configuración inicial, las posiciones se refinan iterativamente mediante un proceso de optimización.

3.6.2.2. Afinidades en el espacio reducido. En el espacio original, las afinidades del grafo son los pesos μ_{ij}, valores entre 0 y 1 que indican qué tan cercanos son dos puntos: cercanos a 1 si están próximos y cercanos a 0 si están alejados. En el espacio reducido, en cambio, los puntos $y_i \in \mathbb{R}^d$ solo tienen posiciones, y una posición por sí sola no permite expresar la cercanía entre dos puntos en esos mismos términos. Por ello, es necesario definir afinidades también en el espacio reducido, de modo que las relaciones de cercanía en ambos espacios puedan compararse directamente y evaluar si la representación reducida es fiel a la estructura original.

UMAP modela esta afinidad mediante la función [163]:

$$\nu_{ij} = \frac{1}{1 + a\,\|y_i - y_j\|_2^{2b}},$$

donde los parámetros $a, b > 0$ se determinan buscando los valores que hacen que ν_{ij}, vista como función de la distancia entre y_i e y_j, se aproxime lo mejor posible a la curva de referencia

$$\Phi(r) = \begin{cases} 1 & \text{si } r \leq \texttt{min-dist}, \\ \exp\big(-(r - \texttt{min-dist})\big) & \text{si } r > \texttt{min-dist}. \end{cases}$$

donde $r = \|y_i - y_j\|_2$.

Concretamente, dado el valor de `min-dist` fijado por el usuario, los parámetros a y b

[162] Ibid.
[163] Ibid.

se determinan automáticamente minimizando la suma de diferencias cuadráticas entre ν_{ij} y $\Phi(r)$, procedimiento conocido como ajuste por mínimos cuadrados no lineales. El hiperparámetro `min-dist` es establecido por el usuario antes de ejecutar el algoritmo y controla la separación mínima permitida entre puntos en el espacio reducido: valores pequeños producen representaciones más compactas, mientras que valores grandes generan una distribución más dispersa de los puntos. La función ν_{ij} es decreciente respecto a la distancia: toma valores próximos a 1 cuando y_i e y_j están cerca, y valores próximos a 0 cuando están alejados, constituyendo así un indicador natural de proximidad en el espacio reducido.

3.6.2.3. Optimización mediante entropía cruzada. El algoritmo busca posiciones $\{y_i\}$ tales que las afinidades ν_{ij} en el espacio reducido se asemejen lo más posible a las afinidades μ_{ij} del grafo original, donde $\mu_{ij} = w_{ij}^{\text{final}}$ son los pesos del grafo no dirigido construido en la etapa anterior y ν_{ij} es la función de afinidad en el espacio reducido. Para cuantificar la diferencia entre ambas afinidades se hace uso de la entropía cruzada binaria[164]:

$$C = \sum_{i \neq j} \left[\mu_{ij} \log \frac{\mu_{ij}}{\nu_{ij}} + (1 - \mu_{ij}) \log \frac{1 - \mu_{ij}}{1 - \nu_{ij}} \right].$$

Dado que los términos $\mu_{ij} \log \mu_{ij}$ y $(1 - \mu_{ij}) \log(1 - \mu_{ij})$ son constantes respecto a las posiciones $\{y_i\}$, minimizar C es equivalente a minimizar

$$\widetilde{C} = -\sum_{i \neq j} \Big(\mu_{ij} \log \nu_{ij} + (1 - \mu_{ij}) \log(1 - \nu_{ij}) \Big).$$

que corresponde precisamente a la entropía cruzada binaria entre las afinidades μ_{ij} y ν_{ij}. La función $\widetilde{C}$ penaliza dos situaciones: pares de puntos con alta afinidad en el grafo original (μ_{ij} grande) que quedan alejados en la representación final (ν_{ij} pequeño), y pares con baja afinidad que aparecen artificialmente próximos. Minimizar $\widetilde{C}$ equivale a aproximar las relaciones de vecindad del grafo original y evitar proximidades no justificadas.

[164] Ibid.

3.6.2.4. Actualización iterativa de las posiciones. La minimización de $\widetilde{C}$ se lleva a cabo de forma iterativa: en cada paso, cada punto y_i se desplaza en la dirección que reduce la discrepancia entre ν_{ij} y μ_{ij}. Este desplazamiento tiene dos efectos simultáneos: acerca los puntos que presentan alta afinidad en el grafo original pero que aún están lejos en la representación, y aleja los que tienen baja afinidad pero aparecen artificialmente próximos[165].

Para controlar la magnitud de estos ajustes, se utiliza una tasa de aprendizaje α_e que decrece linealmente con el número de épocas e:

$$\alpha_e = 1 - \frac{e}{E},$$

donde E denota el número total de épocas, valor que es determinado automáticamente por el algoritmo en función del tamaño del conjunto de datos. Las correcciones son mayores al inicio del proceso y se reducen progresivamente a medida que avanza el número de épocas. El proceso finaliza al completar las E épocas, obteniendo así la representación final en baja dimensión.

3.7. MODELIZACIÓN CON UMAP

La reducción de dimensionalidad mediante UMAP se realizó exclusivamente sobre el conjunto de datos de entrenamiento. Esta decisión metodológica evita la introducción de sesgos y fugas de información (*data leakage*), garantizando que la estructura del espacio reducido se aprenda únicamente a partir de la información disponible durante la fase de ajuste.

A diferencia de PCA, cuyo criterio de reducción se basa en la maximización de la varianza explicada, y del AF, que modela la estructura de correlación entre las variables mediante la estimación de factores comunes por máxima verosimilitud, UMAP construye la representación de baja dimensión preservando las relaciones de afinidad

[165] Ibid.

y vecindad entre las observaciones[166]. Dado que la dimensión del espacio reducido constituye un hiperparámetro libre del método, su elección no está sujeta a ningún criterio intrínseco del algoritmo. Por esta razón, la dimensión del espacio reducido se determinará de manera que sea comparable con el número de componentes y factores seleccionados mediante PCA y AF, para mantener la consistencia experimental y facilitar la comparación directa del desempeño de los modelos de clasificación entre los tres métodos.

Para evaluar cuantitativamente la calidad de la reducción de dimensionalidad obtenida mediante UMAP, se emplearon las métricas Confiabilidad (Trustworthiness) y continuidad (Continuity). Ambas métricas miden en qué medida se preservan las relaciones de vecindad del conjunto de datos original tras la proyección al espacio reducido.

3.7.1. Evaluación de la reducción de dimensionalidad. A lo largo de esta sección, $\{x_1, \ldots, x_n\} \subset \mathbb{R}^p$ denota el conjunto de observaciones en el espacio original y $\{y_1, \ldots, y_n\} \subset \mathbb{R}^d$, con $d \ll p$, su representación reducida obtenida mediante UMAP. Para un entero positivo k, se definen los conjuntos de vecinos más cercanos:

- $\mathcal{N}^{(k)}(x_i)$: el conjunto de los k vecinos más cercanos de x_i en el espacio original $\mathbb{R}^p$, según la distancia $\|x_i - x_j\|_2$.
- $\mathcal{N}^{(k)}(y_i)$: el conjunto de los k vecinos más cercanos de y_i en el espacio reducido $\mathbb{R}^d$, según la distancia $\|y_i - y_j\|_2$.

Asimismo, $r_X(i,j)$ denota el rango del punto x_j en el ordenamiento de los demás puntos según su distancia a x_i en $\mathbb{R}^p$, de menor a mayor, y $r_Y(i,j)$ denota el rango análogo de y_j respecto a y_i en $\mathbb{R}^d$. Dado que cada observación x_j tiene una imagen correspondiente y_j, las operaciones de conjuntos entre $\mathcal{N}^{(k)}(x_i)$ y $\mathcal{N}^{(k)}(y_i)$ se entienden a través de los índices: se escribe $x_j \in \mathcal{N}^{(k)}(y_i)$ para indicar que la imagen y_j figura entre los k vecinos más cercanos de y_i en el espacio reducido.

[166] Ibid.

3.7.1.1. Confiabilidad. La métrica de Confiabilidad evalúa en qué medida los vecinos de un punto en el espacio reducido corresponden a vecinos reales en el espacio original. Formalmente, se define como [167]

$$T(k) = 1 - \frac{2}{nk(2n - 3k - 1)} \sum_{i=1}^{n} \sum_{x_j \in \mathcal{N}^{(k)}(y_i) \setminus \mathcal{N}^{(k)}(x_i)} \big(r_X(i,j) - k\big).$$

El conjunto $\mathcal{N}^{(k)}(y_i) \setminus \mathcal{N}^{(k)}(x_i)$ contiene los puntos que aparecen entre los k vecinos más cercanos de y_i en el espacio reducido, pero que no figuraban entre los k vecinos más cercanos de x_i en el espacio original. El término $r_X(i,j) - k$ penaliza de manera proporcional qué tan lejos se encontraba originalmente x_j de x_i: cuanto mayor era esa distancia en el espacio original, mayor es la penalización. Un valor de $T(k)$ cercano a 1 indica que las vecindades en el espacio reducido reflejan fielmente la estructura local original.

3.7.1.2. Continuidad. La métrica de Continuidad evalúa la dirección complementaria: cuantifica en qué medida los vecinos cercanos en el espacio original permanecen próximos tras la proyección. Se penaliza la pérdida de vecinos, es decir, puntos que eran cercanos en el espacio original pero que quedan alejados en el espacio reducido. Formalmente, se define como [168]

$$C(k) = 1 - \frac{2}{nk(2n - 3k - 1)} \sum_{i=1}^{n} \sum_{x_j \in \mathcal{N}^{(k)}(x_i) \setminus \mathcal{N}^{(k)}(y_i)} \big(r_Y(i,j) - k\big).$$

El conjunto $\mathcal{N}^{(k)}(x_i) \setminus \mathcal{N}^{(k)}(y_i)$ contiene los puntos que eran vecinos cercanos de x_i en el espacio original, pero cuya imagen y_j no figura entre los k vecinos más cercanos de y_i en el espacio reducido. El término $r_Y(i,j) - k$ penaliza de manera proporcional qué tan lejos quedó y_j del vecindario local de y_i tras la proyección. Un valor de $C(k)$ cercano a 1 indica que las relaciones de vecindad del espacio original se conservan satisfactoriamente en la representación reducida.

[167] STASIS, S.; STABLES, R. y HOCKMAN, J. Semantically Controlled Adaptive Equalisation in Reduced Dimensionality Parameter Space. En: *Applied Sciences*. 2016, vol. 6, nro. 4, p. 116.
[168] Ibid.

4. MÉTODOS DE CLASIFICACIÓN PARA IMÁGENES

En este capítulo se describen los modelos de clasificación empleados para la detección automática de enfermedades oculares a partir de imágenes, desde métodos estadísticos clásicos hasta arquitecturas de aprendizaje profundo. Estos modelos se seleccionron tras una fase de exploración en la que se evaluaron múltiples alternativas, eligiendo aquellos que representan perspectivas distintas frente al problema: desde modelos estadísticos interpretables hasta redes neuronales diseñadas para el procesamiento de imágenes. Para cada modelo se presentan: su base teórica, su descripción matemática y los aspectos relevantes para su implementación.

Un modelo de clasificación es un procedimiento que, a partir de un conjunto de datos etiquetados, aprende una regla de decisión capaz de asignar una clase a nuevas observaciones no vistas. En el contexto del análisis de imágenes médicas, esta tarea implica identificar patrones visuales que permitan distinguir entre diferentes categorías diagnósticas, lo que la convierte en una herramienta de apoyo valiosa para la detección automática de enfermedades[169] [170].

4.1. REGRESIÓN LOGÍSTICA PARA CLASIFICACIÓN

La regresión logística es un modelo supervisado cuyo objetivo es estimar la probabilidad de que una observación pertenezca a cada una de las C clases posibles, a partir de un vector de características $\mathbf{x} \in \mathbb{R}^d$. A diferencia de métodos que asignan directamente una etiqueta, la regresión logística produce una distribución de probabilidad sobre las clases, lo que permite cuantificar la incertidumbre de cada predicción [171] [172].

[169] GOODFELLOW. Op. cit.
[170] HASTIE, Trevor; TIBSHIRANI, Robert y FRIEDMAN, Jerome. *The Elements of Statistical Learning*. Springer, 2009.
[171] BISHOP, Christopher M. *Pattern Recognition and Machine Learning*. Springer, 2006.
[172] MURPHY, Kevin P. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

4.1.1. Marco probabilístico. Sea $(\Omega, \mathcal{F}, \mathbb{P})$ un espacio de probabilidad, donde Ω denota el espacio muestral (conjunto de todos los resultados posibles), $\mathcal{F}$ una σ-álgebra de eventos (colección de subconjuntos de Ω a los que se asigna una probabilidad) y $\mathbb{P}$ una medida de probabilidad (función que asigna a cada evento un valor en $[0, 1]$) [173]. Sobre este espacio se definen dos variables aleatorias:

- $X : \Omega \to \mathbb{R}^d$, que representa el vector de características de una observación, donde d es la dimensión del vector de entrada. En el contexto de este trabajo, d corresponde al número de componentes retenidos tras aplicar la reducción de dimensionalidad, o al número de características cuando se trabaja directamente con las imágenes.
- $Y : \Omega \to \{1, \ldots, C\}$, variable categórica que representa la etiqueta de clase asignada a cada observación, tomando valores enteros entre 1 y C.

Se asume que se dispone de una muestra de n observaciones,

$$\{(x_i, y_i)\}_{i=1}^{n},$$

correspondiente a n pares observados del par de variables aleatorias (X, Y). Se asume que estas observaciones son independientes e idénticamente distribuidas (i.i.d.), es decir, cada observación se genera de manera independiente de las demás y todas provienen de la misma población. Esta condición permite expresar la probabilidad conjunta de toda la muestra como el producto de las probabilidades individuales de cada observación, lo cual es fundamental para construir una función de verosimilitud que se utiliza para estimar los parámetros del modelo [174].

El objetivo del modelo es estimar los valores de Y:

$$\mathbb{P}(Y = c \mid X = \mathbf{x}), \quad c \in \mathcal{C} = \{1, \ldots, C\}$$

[173] CASELLA, George y BERGER, Roger L. *Statistical Inference*. 2 ed. Pacific Grove, CA: Duxbury Press, 2002. ISBN 978-0534243128.
[174] Ibid.

la cual cuantifica la probabilidad de que una observación con características $\mathbf{x} \in \mathbb{R}^d$ pertenezca a la clase c [175].

4.1.2. Construcción de un modelo.

4.1.2.1. Predictor lineal. Para modelar la relación entre las variables aleatorias $\mathbf{X}$ y la probabilidad de pertenecer a cada clase, la regresión logística asume que esta relación puede resumirse mediante una combinación lineal de las características. Este supuesto, denominado *hipótesis del predictor lineal* no implica que la probabilidad final sea lineal en $\mathbf{X}$, sino que la información relevante para discriminar entre clases puede capturarse mediante una función lineal de las características[176].

Formalmente, para cada clase $c \in \mathcal{C}$ se introduce un vector de parámetros $\mathbf{w}_c \in \mathbb{R}^d$ y un escalar $b_c \in \mathbb{R}$, denominado término de intercepto.

El predictor lineal asociado a la clase $c \in \mathcal{C}$ se define como

$$z_c(\mathbf{x}) = \mathbf{w}_c^\top \mathbf{X} + b_c,$$

donde cada componente de $\mathbf{w}_c$ cuantifica la influencia de la característica correspondiente sobre la pertenencia a la clase c, y b_c permite desplazar la frontera de decisión para que el modelo no quede forzado a pasar por el origen [177].

El valor $z_c(\mathbf{x})$ no es directamente una probabilidad, pues puede tomar cualquier valor real. Para transformarlo en una probabilidad es necesario un segundo paso, el cual corresponde a aplicar una función que garantice que las salidas sean no negativas y sumen uno. En el caso multiclase, la función estándar para lograr esto es la función softmax, que transforma el vector de predictores en una distribución de probabilidad

[175] BISHOP. Op. cit.
[176] HASTIE. Op. cit.
[177] Ibid.

sobre las C clases [178].

4.1.2.2. Función softmax. Agrupando los valores z_c de todas las clases en el vector $\mathbf{z} = (z_1(\mathbf{x}), \ldots, z_C(\mathbf{x})) \in \mathbb{R}^C$, la función softmax se define como:

$$\mathbb{P}(Y = c \mid X = \mathbf{x}) = \frac{\exp(z_c(\mathbf{x}))}{\sum_{r=1}^{C} \exp(z_r(\mathbf{x}))}, \quad c \in \mathcal{C}.$$

Esta función garantiza que las probabilidades estimadas sean no negativas y sumen uno, lo que permite interpretarlas como probabilidades válidas [179].

4.1.3. Estimación de parámetros.

4.1.3.1. Función de verosimilitud. Los parámetros $\{\mathbf{w}_c, b_c\}_{c=1}^{C}$ se estiman por máxima verosimilitud. Bajo la hipótesis i.i.d., la probabilidad conjunta de observar toda la muestra puede escribirse como el producto de las probabilidades individuales de cada observación, dando lugar a la función de verosimilitud:

$$\mathcal{L}(\{\mathbf{w}_c, b_c\}) = \prod_{i=1}^{n} \prod_{c=1}^{C} \mathbb{P}(Y = c \mid X = x_i)^{\mathbb{I}\{y_i = c\}},$$

donde $\mathbb{I}\{y_i = c\}$ es la función indicadora:

$$\mathbb{I}\{y_i = c\} = \begin{cases} 1, & \text{si } y_i = c, \\ 0, & \text{en otro caso.} \end{cases}$$

El exponente $\mathbb{I}\{y_i = c\}$ garantiza que solo contribuya al producto la clase verdadera de cada observación. Maximizar $\mathcal{L}$ equivale a encontrar los parámetros que hacen más probable la muestra observada bajo el modelo [180].

[178] BISHOP. Op. cit.
[179] Ibid.
[180] CASELLA. Op. cit.

4.1.3.2. Función de pérdida: entropía cruzada categórica. Al tomar logaritmos y cambiar el signo, maximizar $\mathcal{L}$ equivale a minimizar la función de pérdida conocida como entropía cruzada categórica:

$$\ell(y, \hat{y}) = -\sum_{i=1}^{n} \sum_{c=1}^{C} \mathbb{I}\{y_i = c\} \log \hat{y}_{i,c},$$

donde $\hat{y}_{i,c} = \mathbb{P}(Y = c \mid X = x_i)$ es la probabilidad estimada de que la observación i pertenezca a la clase c. Esta función asigna un valor numérico, denominado pérdida, que mide qué tan alejadas están las probabilidades estimadas de la realidad: una pérdida pequeña indica que las probabilidades asignadas a cada categoría están cerca de los valores reales, mientras que una pérdida grande indica que el modelo se equivoca con frecuencia o tiene poca confianza en sus predicciones. En el límite, si el modelo predice correctamente todas las observaciones con certeza total, la entropía cruzada vale cero [181] [182].

La minimización de ℓ se lleva a cabo mediante algoritmos de optimización basados en gradiente, que ajustan iterativamente los parámetros $\{\mathbf{w}_c, b_c\}$ hasta alcanzar un mínimo. Una vez entrenado el modelo, la clasificación de una nueva observación $\mathbf{x}$ se obtiene asignándola a la clase con mayor probabilidad estimada:

$$\hat{y} = \arg\max_{c \in \mathcal{C}}; \mathbb{P}(Y = c \mid X = \mathbf{x}).$$

4.2. MÁQUINAS DE SOPORTE VECTORIAL

Las Máquinas de Soporte Vectorial (SVM) son un conjunto de algoritmos de aprendizaje supervisado fundamentados en la Teoría del aprendizaje estadístico y el principio de minimización del riesgo estructural [183]. En su formulación básica, una SVM busca encontrar un hiperplano, es decir, una superficie plana que divide el espacio en dos regiones, de modo que cada región contenga los puntos de una clase distinta.

[181] BISHOP. Op. cit.
[182] MURPHY. Op. cit.
[183] VAPNIK, Vladimir N. *Statistical Learning Theory*. New York: Wiley, 1998.

A diferencia de los modelos que buscan únicamente reducir el error sobre los datos de entrenamiento, las SVM incorporan en su función objetivo un criterio explícito de complejidad: maximizar la distancia entre el hiperplano que separa las clases y los puntos más cercanos a él, distancia que se denomina margen. Por tanto, el modelo busca que los puntos de cada clase queden lo más alejados posible del hiperplano separador, lo que le otorga una mayor capacidad de generalización a nuevas observaciones [184].

4.2.1. Geometría del hiperplano óptimo. La formulación matemática de las SVM se desarrolla inicialmente para el caso de clasificación binaria, es decir, cuando las observaciones pertenecen a una de dos clases posibles. Posteriormente, en la Sección 4.2.4 se describe cómo esta formulación se extiende al caso multiclase. Para aplicar este método, cada observación debe estar representada como un vector numérico de dimensión d, donde d corresponde al número de componentes retenidos después de la reducción dimensional o al número de características de la imagen cuando se trabaja directamente con ella. El problema central consiste en encontrar un hiperplano afín, es decir, un hiperplano que no necesariamente pasa por el origen, que divida el espacio en dos regiones, una por clase [185] [186].

Sea un conjunto de datos de entrenamiento

$$\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$$

donde cada vector de características $\mathbf{x}_i \in \mathbb{R}^d$ representa una observación, y la variable respuesta $y_i \in \{-1, +1\}$ indica la clase a la que pertenece dicha observación. El objetivo es encontrar un hiperplano que separe correctamente las muestras de ambas clases.

Desde un punto de vista analítico, un hiperplano afín en $\mathbb{R}^d$ puede describirse como el

[184] Ibid.
[185] CORTES, Corinna y VAPNIK, Vladimir. Support-vector networks. En: *Machine Learning*. 1995, vol. 20, nro. 3, pp. 273-297.
[186] VAPNIK. Op. cit.

conjunto de puntos que satisfacen la ecuación

$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b = 0,$$

donde el vector $\mathbf{w} \in \mathbb{R}^d$ es normal al hiperplano y determina su orientación, mientras que el escalar $b \in \mathbb{R}$ controla su desplazamiento respecto al origen. En ausencia del término b, el hiperplano estaría forzado a pasar por el origen, lo cual restringiría innecesariamente la flexibilidad del modelo.

La función $f(\mathbf{x})$ no solo define la frontera de decisión, sino que también permite clasificar nuevas observaciones mediante el signo de su evaluación: un punto $\mathbf{x}$ se asigna a la clase $+1$ si $f(\mathbf{x}) > 0$ y a la clase -1 si $f(\mathbf{x}) < 0$ [187].

4.2.1.1. Definición del Margen. Para que el hiperplano clasifique correctamente todas las observaciones del conjunto de entrenamiento, es necesario que el signo de $f(\mathbf{x}_i)$ coincida con el signo de la etiqueta y_i para cada $i = 1, \ldots, n$. Esta condición puede expresarse de forma compacta como

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) > 0.$$

Sin embargo, la ecuación del hiperplano no cambia si se multiplican $\mathbf{w}$ y b por una misma constante positiva. En efecto, si $(\mathbf{w}, b)$ define un hiperplano separador, entonces $(\beta\mathbf{w}, \beta b)$, para cualquier $\beta > 0$, define el mismo hiperplano. Para eliminar esta ambigüedad y permitir una formulación bien definida del problema de optimización, se introduce una normalización canónica, la cual consiste en imponer que las observaciones más cercanas al hiperplano satisfagan

$$y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1, \quad \forall i = 1, \ldots, n.$$

Las observaciones para las que esta restricción se cumple con igualdad, es decir, $y_i(\mathbf{w}^\top \mathbf{x}_i + b) = 1$, son las más cercanas al hiperplano separador $\mathbf{w}^\top \mathbf{x} + b = 0$ y se

[187] CORTES, Corinna y VAPNIK, Vladimir. Support-vector networks. En: *Machine Learning*. 1995, vol. 20, nro. 3, pp. 273-297.

ubican sobre los dos hiperplanos paralelos

$$\mathbf{w}^\top \mathbf{x} + b = \pm 1,$$

los cuales delimitan el margen. Las observaciones ubicadas sobre dichos hiperplanos reciben el nombre de vectores de soporte, ya que son las únicas que determinan la posición y orientación del hiperplano óptimo [188].

Dado que para los vectores de soporte la normalización canónica garantiza que $|f(\mathbf{x}_i)| = 1$, la distancia euclidiana desde un punto $\mathbf{x}_i$ al hiperplano está dada por

$$r_i = \frac{|f(\mathbf{x}_i)|}{\|\mathbf{w}\|},$$

Dado que la normalización canónica garantiza $|f(\mathbf{x}_i)| = 1$ para todos los vectores de soporte, al sustituir en r_i todos obtienen la misma distancia al hiperplano. Como los vectores de soporte son por definición los puntos más cercanos al hiperplano, esta distancia es la mínima posible [189] [190]:

$$r_{\text{mín}} = \frac{1}{\|\mathbf{w}\|}.$$

El margen total se obtiene sumando las distancias mínimas de cada clase al hiperplano separador, una por cada lado:

$$\text{Margen} = \frac{1}{\|\mathbf{w}\|} + \frac{1}{\|\mathbf{w}\|} = \frac{2}{\|\mathbf{w}\|}.$$

Esta expresión revela una relación fundamental: maximizar el margen equivale a minimizar la norma del vector $\mathbf{w}$. Esta equivalencia permite transformar el problema geométrico de separación óptima en un problema de optimización convexa, que es la base del enfoque de las SVM.

[188] Ibid.

[189] BOYD, Stephen y VANDENBERGHE, Lieven. *Convex Optimization*. Cambridge University Press, 2004.

[190] CORTES. Op. cit.

4.2.2. El problema de optimización. El objetivo de la SVM es maximizar este margen $\frac{2}{\|\mathbf{w}\|}$. Matemáticamente, maximizar $\frac{2}{\|\mathbf{w}\|}$ es equivalente a minimizar $\|\mathbf{w}\|$, o por conveniencia analítica, minimizar $\frac{1}{2}\|\mathbf{w}\|^2$. Lo cual conduce al problema de optimización en su forma primal, es decir, expresado directamente en términos de las variables originales $\mathbf{w}$ y b [191]:

$$\begin{aligned} &\min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}\|^2 \\ &\text{sujeto a} \quad y_i(\mathbf{w}^\top \mathbf{x}_i + b) \geq 1, \quad \forall i = 1, \ldots, n. \end{aligned}$$

4.2.2.1. Resolución mediante el enfoque dual. Para resolver el problema primal se utiliza el método de los multiplicadores de Lagrange, el cual transforma el problema original en uno equivalente, denominado problema dual, cuya solución permite obtener los valores α_i que identifican los vectores de soporte. A partir de estos valores se recupera el vector $\mathbf{w}$ y posteriormente el sesgo b.

Se trata de un problema de optimización convexa con restricciones. Para resolverlo se utiliza el método de los multiplicadores de Lagrange. Se construye la función lagrangiana:

$$L_P(\mathbf{w}, b, \boldsymbol{\alpha}) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{n} \alpha_i [y_i(\mathbf{w}^\top \mathbf{x}_i + b) - 1],$$

donde $\alpha_i \geq 0$ son los multiplicadores de Lagrange asociados a cada restricción. Derivando L_P respecto a $\mathbf{w}$ y b e igualando a cero se obtienen las condiciones de Karush-Kuhn-Tucker (KKT) [192]:

$$\frac{\partial L_P}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = 0 \implies \mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i, \tag{4.1}$$

$$\frac{\partial L_P}{\partial b} = -\sum_{i=1}^{n} \alpha_i y_i = 0 \implies \sum_{i=1}^{n} \alpha_i y_i = 0. \tag{4.2}$$

La ecuación (4.1) revela que el vector $\mathbf{w}$ óptimo es una combinación lineal de los datos de entrenamiento. Sin embargo, debido a las condiciones KKT, $\alpha_i = 0$ para la mayoría de las observaciones; solo aquellas ubicadas sobre el margen, es decir, las

[191] Ibid.
[192] BOYD. Op. cit.

que satisfacen $\alpha_i > 0$, son los vectores de soporte.

Sustituyendo estas condiciones en la función lagrangiana se obtiene el problema dual [193]:

$$\max_{\boldsymbol{\alpha}} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j \, \mathbf{x}_i^\top \mathbf{x}_j$$

$$\text{sujeto a} \quad \sum_{i=1}^{n} \alpha_i y_i = 0, \quad \alpha_i \geq 0, \quad \forall\, i, \ldots, n.$$

Conceptualmente, este problema asigna a cada observación un peso $\alpha_i \geq 0$. La mayoría de las observaciones reciben $\alpha_i = 0$, lo que significa que no participan en la construcción del hiperplano separador. Solo aquellas observaciones ubicadas sobre el margen reciben $\alpha_i > 0$ y son las que determinan la posición del hiperplano; estos son los vectores de soporte. La restricción $\sum_{i=1}^{n} \alpha_i y_i = 0$ garantiza que las contribuciones de ambas clases estén balanceadas.

Este problema depende de los datos únicamente a través de productos escalares $\mathbf{x}_i^\top \mathbf{x}_j$, lo cual es fundamental para la extensión no lineal mediante kernels descrita en la siguiente subsección. Una vez resuelto este problema y obtenidos los valores α_i, el vector $\mathbf{w}$ se obtiene directamente sustituyendo los α_i en $\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$. Solo queda determinar el sesgo b, lo cual se describe a continuación.

4.2.2.2. Obtención del sesgo b. El sesgo b se obtiene tomando cualquier vector de soporte $\mathbf{x}_s$, donde $y_s \in \{-1, +1\}$ es su etiqueta de clase, para el cual la restricción se cumple con igualdad:

$$y_s(\mathbf{w}^\top \mathbf{x}_s + b) = 1.$$

Multiplicando por y_s y usando que $y_s^2 = 1$, se despeja b:

$$b = y_s - \mathbf{w}^\top \mathbf{x}_s.$$

En la práctica, se calcula b para cada vector de soporte y se toma el promedio como valor final del sesgo, ya que los valores individuales pueden presentar pequeñas

[193] CORTES. Op. cit.

variaciones numéricas [194] [195].

4.2.3. Extensión no lineal mediante kernels. En muchos problemas de clasificación, como el diagnóstico a partir de imágenes médicas, los datos no son separables linealmente en el espacio original $\mathbb{R}^d$. Una solución consiste en transformar los datos mediante una función $\phi : \mathbb{R}^d \to \mathcal{H}$, donde $\mathcal{H}$ es un espacio de mayor dimensión en el que los datos sí sean separables.

Sin embargo, calcular $\phi(\mathbf{x})$ explícitamente puede ser computacionalmente costoso o incluso inviable cuando $\mathcal{H}$ tiene dimensión muy alta. Dado que el problema dual depende de los datos únicamente a través de productos escalares, es posible reemplazar $\mathbf{x}_i^\top \mathbf{x}_j$ por una función $K(\mathbf{x}_i, \mathbf{x}_j)$ que calcula el producto escalar en $\mathcal{H}$ sin necesidad de calcular $\phi(\mathbf{x})$ explícitamente. Este recurso se denomina el *truco del kernel* [196].

El kernel es una función que mide qué tan similares son dos observaciones entre sí. Entre más parecidas, mayor es el valor del kernel, y en caso contrario este es menor. Esta medida de similitud le permite al modelo detectar patrones y estructuras en los datos que no serían visibles si solo se compararan las distancias directas entre observaciones [197].

En este trabajo se utilizó el kernel de base radial (RBF):

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\big(-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|^2\big)\,,$$

donde $\gamma > 0$ es un hiperparámetro que controla qué tan sensible es el modelo a las diferencias entre observaciones. Valores grandes de γ hacen que el modelo solo considere similares a las observaciones muy cercanas entre sí, lo que produce una separación entre clases más ajustada a los datos de entrenamiento. Valores pequeños de γ consideran similares a observaciones más lejanas, produciendo una separación más grande entre las clases[198].

[194] BOYD. Op. cit.
[195] CORTES. Op. cit.
[196] VAPNIK. Op. cit.
[197] Ibid.
[198] Ibid.

4.2.4. Extensión a clasificación multiclase. Dado que la formulación matemática de SVM es binaria, para clasificar las $K > 2$ clases se implementó la estrategia Uno-contra-Uno (One-vs-One) [199] [200]. Se entrenan $K(K-1)/2$ clasificadores binarios, uno por cada par de clases, y la decisión final se obtiene mediante votación mayoritaria: cada clasificador emite un voto por una de las dos clases que distingue, y la observación se asigna a la clase que recibe más votos[201].

Una vez estimados $\mathbf{w}$ y b, la clasificación de una nueva observación $\mathbf{x}$ se obtiene mediante

$$\hat{y} = \text{signo}\left(\sum_{i \in SV} \alpha_i y_i \, K(\mathbf{x}_i, \mathbf{x}) + b\right),$$

donde SV denota el conjunto de vectores de soporte, es decir, aquellos con $\alpha_i > 0$. La función signo devuelve $+1$ si el valor dentro del paréntesis es positivo, asignando la observación a una clase, y -1 si es negativo, asignándola a la otra. Este proceso se repite independientemente para cada par de clases posible y al final la clase que acumuló más votos es la predicción final [202] [203].

4.3. REDES NEURONALES PARA CLASIFICACIÓN

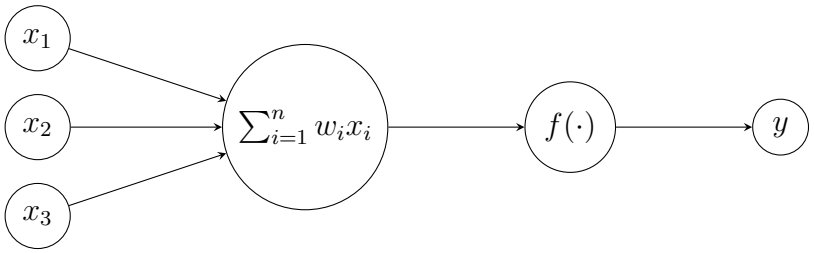
Las Redes Neuronales Artificiales (RNAs) son algoritmos ampliamente utilizados en el campo de la Inteligencia Artificial, debido a su capacidad para modelar relaciones complejas entre variables a partir de datos observados. Una RNA puede entenderse como un conjunto de elementos simples, denominados neuronas artificiales, que se encuentran conectados entre sí y actúan de forma conjunta para producir una respuesta del sistema frente a una entrada determinada [204]. Estas arquitecturas se inspiran conceptualmente en la forma en que el sistema nervioso humano procesa la información: las señales de entrada provienen de otras neuronas a través de conexiones denomina-

[199] BISHOP. Op. cit.
[200] HSU, Chih-Wei y LIN, Chih-Jen. A comparison of methods for multiclass support vector machines. En: *IEEE Transactions on Neural Networks*. 2002, vol. 13, nro. 2, pp. 415-425.
[201] BISHOP. Op. cit.
[202] Ibid.
[203] HSU. Op. cit.
[204] LÓPEZ, Juan Carlos. *Redes neuronales artificiales: fundamentos, modelos y aplicaciones*. Madrid: Editorial Alfaomega, 2008.

das sinapsis, cada una con una influencia distinta, y su efecto conjunto determina si la neurona genera una señal de salida. Esta señal puede variar en intensidad dependiendo de la fuerza de las conexiones[205] [206]. De forma análoga, en una neurona artificial cada unidad recibe un conjunto de valores de entrada y asigna a cada uno una importancia relativa mediante parámetros numéricos conocidos como *pesos*. A partir de esta información, la neurona produce una salida a través de una función matemática denominada *función de activación*. Aunque esta analogía es únicamente conceptual y no pretende reproducir fielmente los mecanismos biológicos del cerebro, resulta útil para construir modelos capaces de aprender patrones a partir de los datos de manera automática [207].

Desde un punto de vista matemático, una neurona artificial recibe un vector de entradas observadas $\mathbf{x} = (x_1, x_2, \ldots, x_n)^\top$ y salida y. A cada entrada se le asigna un peso mediante un vector $\mathbf{w} = (w_1, w_2, \ldots, w_n)^\top$, que refleja la contribución relativa de cada entrada. Estas cantidades se combinan a través de una expresión lineal $\sum_{i=1}^{n} w_i x_i$, que resume la información de entrada en un único valor escalar. Posteriormente, este valor se transforma mediante una función de activación, que introduce no linealidad en el modelo y permitir la representación de relaciones más complejas entre los datos. Esta estructura es la base fundamental para la construcción de redes neuronales de mayor complejidad.

Figura 3. Esquema de una neurona artificial o perceptrón simple.



Fuente: Elaboración propia.

La Figura 3 corresponde al perceptrón simple, que representa la forma más básica de una neurona artificial: una sola unidad que recibe entradas, las combina mediante

[205] GOODFELLOW. Op. cit.
[206] HAYKIN, Simon. *Neural Networks and Learning Machines*. 3 ed. Upper Saddle River, NJ: Prentice Hall, 2009.
[207] Ibid.

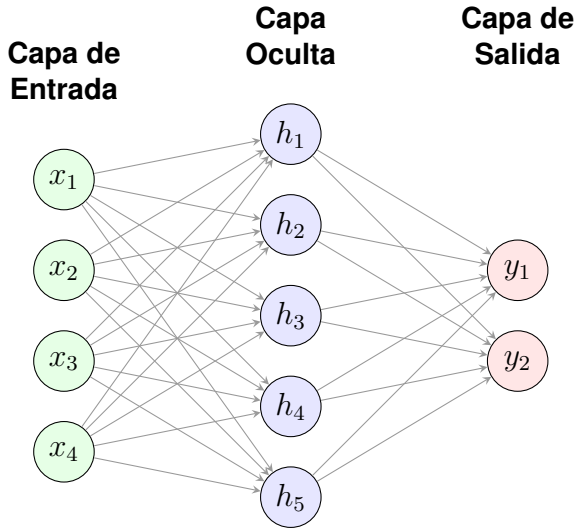
pesos y produce una salida a través de una función de activación [208]. Existen diversos tipos de arquitecturas de redes neuronales artificiales, diseñadas para abordar distintos tipos de datos y problemas. Entre ellas se encuentran las redes totalmente conectadas (densas), las redes neuronales convolucionales (CNN) y arquitecturas más recientes que incorporan principios de diseño eficiente, tales como el equilibrio entre profundidad, anchura y resolución de la red[209] [210].

Estas arquitecturas eficientes, entre las que se encuentra la familia EfficientNet, buscan alcanzar un alto desempeño manteniendo un número reducido y bien distribuido de parámetros entrenables. Esto permite disminuir el consumo de memoria, reducir el tiempo de entrenamiento y facilitar la implementación del modelo en entornos con recursos computacionales limitados, como unidades de procesamiento gráfico con memoria restringida o sistemas de cómputo compartido. Dichos enfoques resultan especialmente relevantes en tareas de visión por computador, donde los modelos suelen manejar grandes volúmenes de datos.

4.3.1. Red neuronal densa o perceptrón multicapa. Una red neuronal densa, también conocida como red totalmente conectada o perceptrón multicapa (MLP por sus siglas en inglés), es una arquitectura en la que las neuronas se organizan en capas secuenciales: una capa de entrada que recibe los datos, una o más capas ocultas que transforman la información, y una capa de salida que produce la predicción final. En esta arquitectura, cada neurona de una capa se conecta con todas las neuronas de la capa siguiente, de ahí el nombre de red totalmente conectada o densa[211]. La Figura 4 ilustra esta estructura.

[208] Ibid.
[209] GOODFELLOW. Op. cit.
[210] TAN, Mingxing y LE, Quoc V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. En: *Proceedings of the 36th International Conference on Machine Learning (ICML)*. 2019, pp. 6105-6114.
[211] GOODFELLOW. Op. cit.

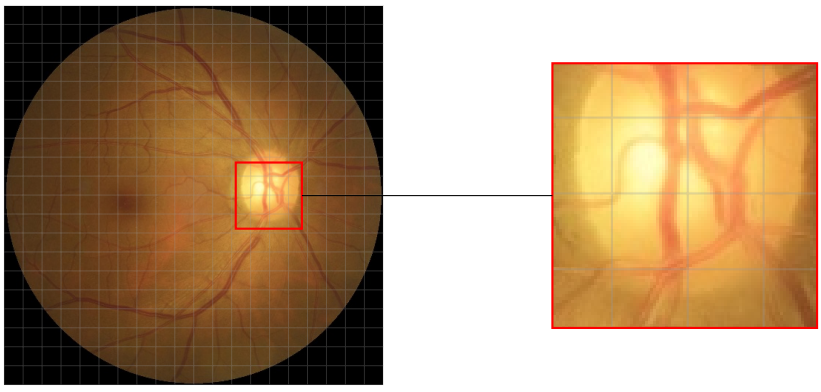
Figura 4. Esquema de una Red Neuronal Densa (Multicapa).



Fuente: Elaboración propia.

4.3.1.1. Limitaciones de las redes neuronales densas. Las imágenes poseen una estructura espacial, es decir, sus valores no son simplemente una lista de números independientes sino que están organizados en una cuadrícula bidimensional de píxeles donde la posición relativa de cada uno respecto a sus vecinos es relevante. Los píxeles cercanos tienden a estar más correlacionados que los píxeles alejados, lo que permite identificar patrones locales como bordes, regiones homogéneas o texturas, como se ilustra en la Figura 5.

Figura 5. Visualización de la estructura espacial. La rejilla representa los píxeles.



(a) Imagen Vectorizada $\mathbf{X}$ **(b) Vecindad local ampliada**

Fuente: ODIR-5K

En una red neuronal densa, esta información espacial se pierde, ya que la imagen

debe transformarse en un vector unidimensional antes de ser procesada. Como consecuencia, se ignoran las relaciones locales entre píxeles y se incrementa el número de parámetros del modelo, lo que eleva el riesgo de sobreajuste[212] [213]. Estas limitaciones motivan el desarrollo de arquitecturas capaces de explotar la estructura espacial de los datos.

4.3.2. Fundamentos de las Redes Neuronales Convolucionales. Las Redes Neuronales Convolucionales (CNN, por sus siglas en inglés) surgen como una extensión natural de las RNAs tradicionales, diseñadas específicamente para el procesamiento de datos con estructura espacial, como imágenes bidimensionales. La idea central consiste en reemplazar las conexiones densas por operaciones locales, denominadas *convoluciones*, que actúan sobre pequeñas regiones de la imagen, procesando únicamente grupos reducidos de píxeles vecinos en cada paso. Esto permite extraer características relevantes, como bordes, transiciones de intensidad o patrones simples, preservando al mismo tiempo la organización espacial de la imagen [214]. Este enfoque se inspira conceptualmente en estudios del sistema visual biológico, donde se ha observado que ciertas neuronas responden únicamente a estímulos localizados dentro de regiones específicas del campo visual, conocidas como campos receptivos[215].

4.3.3. Operación de convolución. La convolución es la operación fundamental mediante la cual una CNN analiza una imagen de forma estructurada. Su objetivo es identificar patrones locales, es decir, características que aparecen en regiones pequeñas de la imagen, tales como bordes, transiciones entre zonas claras y oscuras, o variaciones de textura. En lugar de procesar la imagen como un todo, examina pequeñas regiones locales realizando un cálculo numérico sencillo sobre cada una de ellas. Este cálculo se repite de forma ordenada mientras la región se desplaza a lo largo de toda la imagen, produciendo nuevos valores que reflejan la presencia de dichos patrones en cada ubicación.

[212] Ibid.
[213] LeCUN, Yann; BENGIO, Yoshua y HINTON, Geoffrey. Deep Learning. En: *Nature*. 2015, vol. 521, pp. 436-444.
[214] LeCUN, Yann; BOTTOU, Léon; BENGIO, Yoshua y HAFFNER, Patrick. Gradient-based learning applied to document recognition. En: *Proceedings of the IEEE*. 1998, vol. 86, nro. 11, pp. 2278-2324.
[215] Ibid.

Desde un punto de vista matemático, una imagen digital en color puede representarse como una colección de matrices bidimensionales, una por cada canal de color. En particular, una imagen RGB se describe mediante un arreglo de valores reales de la forma

$$\mathbf{X} \in \mathbb{R}^{H \times W \times C},$$

donde H y W representan la altura y el ancho de la imagen, respectivamente, y C denota el número de canales de color. En el caso de imágenes RGB, se tiene usualmente $C = 3$, correspondientes a los canales rojo, verde y azul[216].

Para extraer información local de la imagen, las redes neuronales convolucionales introducen el concepto de *filtro* o *núcleo convolucional*. Un filtro no corresponde a una subregión de la imagen ni se obtiene directamente a partir de los datos observados. En su lugar, consiste en un conjunto de parámetros numéricos libres del modelo, organizados como una colección de matrices

$$\mathbf{K} = \{K_c\}_{c=1}^{C}, \quad K_c \in \mathbb{R}^{k \times k},$$

donde el tamaño espacial $k \times k$ es fijo y considerablemente menor que el de la imagen de entrada.

El filtro es necesario porque analizar todos los píxeles de la imagen a la vez sería ineficiente y no permitiría identificar patrones locales. En cambio, el filtro actúa como una lupa pequeña que se desliza por la imagen buscando un patrón específico: produce un valor alto cuando encuentra lo que busca y un valor bajo cuando no. Por ejemplo, si se quiere detectar bordes en la imagen, se usa un filtro cuyos coeficientes reaccionan ante cambios bruscos de intensidad entre píxeles vecinos, como el límite entre una zona clara y una oscura. En zonas uniformes donde no hay cambios, ese mismo filtro producirá valores cercanos a cero.

Inicialmente, los valores del filtro se eligen de forma arbitraria mediante una inicialización aleatoria de pequeña magnitud. Durante el entrenamiento, estos valores se ajustan automáticamente: el modelo calcula qué tan equivocadas están sus predicciones,

[216]GOODFELLOW. Op. cit.

y con base en eso modifica cada coeficiente del filtro en la dirección que reduce ese error. Este proceso se repite iterativamente hasta que el filtro aprende qué patrones locales son relevantes para la tarea de clasificación[217] [218].

La operación de convolución consiste en desplazar el filtro sobre la imagen y, en cada posición válida, calcular una combinación ponderada de los valores de los píxeles locales en todos los canales de color. Formalmente, el valor del mapa de salida en la posición (i, j) se define como

$$(\mathbf{X} * \mathbf{K})(i, j) = \sum_{c=1}^{C} \sum_{u=0}^{k-1} \sum_{v=0}^{k-1} X_c(i + u, j + v)\, K_c(u, v).$$

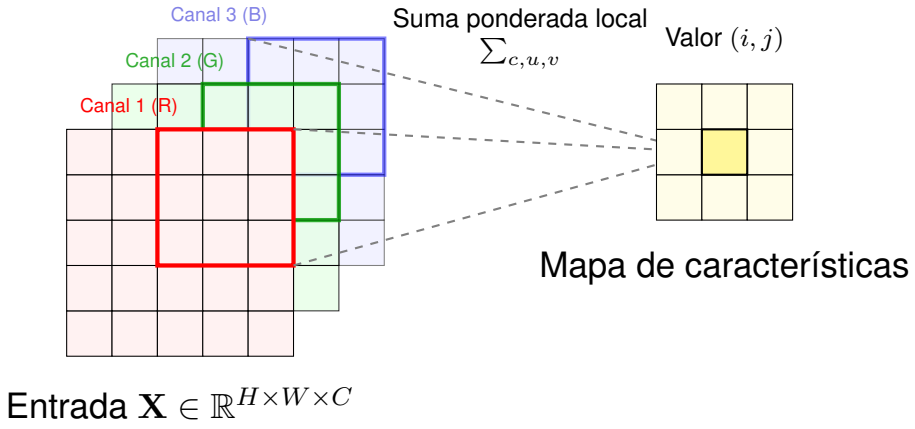
Este cálculo produce un valor numérico que mide el grado de coincidencia entre el patrón codificado en el filtro y la región local de la imagen centrada en la posición (i, j). Al repetir esta operación sobre todas las posiciones válidas de la imagen, se obtiene una nueva matriz bidimensional denominada *mapa de características*. Dicho mapa indica en qué zonas de la imagen el patrón aprendido por el filtro aparece con mayor intensidad.

En este contexto, se denominan *posiciones válidas* aquellas ubicaciones del filtro para las cuales éste se superpone completamente sobre la imagen de entrada, de modo que cada uno de sus coeficientes puede asociarse a un píxel existente. Las posiciones cercanas a los bordes de la imagen que no cumplen esta condición quedan excluidas del cálculo. Una consecuencia de excluir los bordes es que el mapa de características resultante es más pequeño que la imagen de entrada. Para evitar esta reducción, existe una técnica denominada relleno (*padding*), que consiste en agregar píxeles adicionales alrededor de los bordes de la imagen antes de aplicar el filtro, permitiendo que este se centre también en los píxeles del borde y produciendo un mapa de salida del mismo tamaño que la entrada. Sin embargo, en esta sección se considera únicamente la convolución sin relleno para mantener la explicación más simple.

Una representación gráfica de este procedimiento se detalla en la Figura 6

[217] Ibid.
[218] LeCUN, Yann et al. Op. cit., 1998.

Figura 6. Visualización de la operación de convolución sobre una imagen RGB.



$$(\mathbf{X} * \mathbf{K})(i, j) = \sum_{c=1}^{3} \sum_{u=0}^{k-1} \sum_{v=0}^{k-1} X_c(i + u, j + v)\, K_c(u, v)$$

Fuente: Elaboración propia.

4.3.4. Capas convolucionales y parámetros del modelo. Una capa convolucional se define como un conjunto de filtros convolucionales que actúan en paralelo sobre una misma imagen de entrada. Cada filtro aplica la operación de convolución descrita en la subsección anterior y produce un mapa de características que resalta la presencia de un patrón específico en distintas regiones de la imagen.

Sea $\mathbf{X} \in \mathbb{R}^{H \times W \times C}$ la imagen de entrada a la capa. Al usar múltiples filtros en paralelo, la capa convolucional puede capturar simultáneamente distintos tipos de patrones presentes en la imagen, por ejemplo, un filtro puede especializarse en bordes horizontales mientras otro lo hace en bordes verticales. Si la capa contiene M filtros, estos se denotan por:

$$\mathbf{K}^{(1)}, \mathbf{K}^{(2)}, \ldots, \mathbf{K}^{(M)},$$

donde cada filtro $\mathbf{K}^{(m)}$ está formado por una colección de matrices

$$\mathbf{K}^{(m)} = \{K_c^{(m)}\}_{c=1}^{C}, \quad K_c^{(m)} \in \mathbb{R}^{k \times k}.$$

Al aplicar cada filtro sobre la imagen, se obtiene un mapa de características bidimen-

sional. En particular, el mapa de salida asociado al filtro $\mathbf{K}^{(m)}$ se define como

$$\mathbf{Y}^{(m)}(i,j) = (\mathbf{X} * \mathbf{K}^{(m)})(i,j), \qquad m = 1, \ldots, M.$$

Por tanto, la salida completa de la capa convolucional está formada por M mapas de características,

$$\mathbf{Y}^{(1)}, \mathbf{Y}^{(2)}, \ldots, \mathbf{Y}^{(M)},$$

que pueden organizarse como un arreglo tridimensional

$$\mathbf{Y} \in \mathbb{R}^{H' \times W' \times M}.$$

Desde esta perspectiva, cada posición espacial (i,j) de la salida queda asociada a un vector de M valores, donde cada componente cuantifica la respuesta de un filtro distinto en dicha región de la imagen.

Los *parámetros entrenables* de una capa convolucional están constituidos exclusivamente por los coeficientes numéricos de sus filtros. En consecuencia, si la capa contiene M filtros de tamaño espacial $k \times k$ aplicados sobre imágenes con C canales, el número total de parámetros es

$$M \cdot k \cdot k \cdot C.$$

Estos parámetros desempeñan el papel de pesos del modelo y se ajustan automáticamente durante el proceso de entrenamiento mediante algoritmos de optimización basados en gradiente.

Una característica esencial de las capas convolucionales es que los coeficientes de cada filtro se utilizan de manera idéntica en todas las posiciones de la imagen. Es decir, el mismo conjunto de parámetros se emplea para evaluar cada región local, sin introducir pesos distintos para ubicaciones espaciales diferentes. Matemáticamente, esto implica que la transformación definida por un filtro es independiente de la posición y responde de la misma forma ante un mismo patrón, con independencia de dónde aparezca en la imagen.

Esta reutilización sistemática de parámetros permite reducir de forma significativa la

complejidad del modelo y favorece su capacidad de generalización, ya que la red aprende a detectar características visuales relevantes sin depender de su localización exacta dentro de la imagen[219] [220].

4.3.5. Capas de activación y submuestreo. La salida de una capa convolucional consiste en uno o varios mapas de características cuyos valores se obtienen mediante combinaciones lineales de los píxeles de la imagen de entrada. Si estas salidas se utilizaran directamente, el modelo completo sería esencialmente una composición de transformaciones lineales, lo cual limitaría severamente su capacidad para modelar relaciones complejas.

Para superar esta limitación, las redes neuronales convolucionales incorporan *capas de activación*, que aplican funciones no lineales de manera puntual a los valores de los mapas de características. Formalmente, si

$$\mathbf{Y} \in \mathbb{R}^{H' \times W' \times M}$$

denota la salida de una capa convolucional, una capa de activación define una nueva salida

$$\mathbf{Z}(i, j, m) = \varphi\big(\mathbf{Y}(i, j, m)\big),$$

donde $\varphi : \mathbb{R} \to \mathbb{R}$ es una función no lineal aplicada elemento a elemento.

Estas funciones, denominadas *funciones de activación*, no introducen nuevos parámetros entrenables, pero desempeñan un papel esencial al permitir que la red represente estructuras no lineales presentes en los datos.

Entre las funciones de activación más utilizadas se encuentran las siguientes:

[219] GOODFELLOW. Op. cit.
[220] LeCUN, Yann et al. Op. cit., 1998.

Tabla 1. Funciones de activación comunes en redes neuronales.

Función	**Expresión**	**Descripción**
Lineal	$\varphi(x) = x$	No introduce no linealidad. Su uso se limita a capas de salida en problemas de regresión[221].
Sigmoide o logística	$\varphi(x) = \frac{1}{1+e^{-x}}$	Transforma cualquier valor real en un número del intervalo $(0, 1)$, lo que permite interpretaciones probabilísticas. Sin embargo, puede presentar problemas de saturación del gradiente para valores grandes de $\|x\|$ [222].
Tangente hiperbólica	$\varphi(x) = \tanh(x)$	Produce valores en $(-1, 1)$. Versión centrada en cero de la sigmoide, comparte sus limitaciones en redes profundas [223].
ReLU	$\varphi(x) = \max(0, x)$	Computacionalmente sencilla y evita el desvanecimiento del gradiente. Es la función estándar en arquitecturas modernas [224].

Fuente: Elaboración propia.

Desde un punto de vista geométrico, estas funciones modifican la forma en que los valores de los mapas de características se propagan a través de la red, permitiendo que diferentes patrones visuales interactúen de manera no lineal en capas posteriores. Además de las capas de activación, las CNN aplican una etapa adicional denominada *submuestreo*, cuyo objetivo es reducir el tamaño espacial del mapa conservando la información más relevante. Para ello, el mapa de características $\mathbf{Z} \in \mathbb{R}^{H' \times W'}$ se divide en regiones locales disjuntas de tamaño fijo, por ejemplo 2×2, y a cada región se le aplica una función que condensa sus valores en un único número representativo.

En el caso del *max pooling*, el valor asignado a una región $\mathcal{R}$ corresponde al máximo de sus elementos:

$$\max_{(i,j)\in\mathcal{R}} \mathbf{Z}(i, j).$$

Al aplicar este procedimiento sobre todas las regiones se obtiene un nuevo mapa de

menor tamaño espacial, manteniendo las activaciones más significativas y reduciendo la complejidad computacional del modelo.

4.3.6. Operación aplanamiento y transición a capas densas. Luego de aplicar sucesivamente capas convolucionales, funciones de activación y operaciones de submuestreo, la información de la imagen ya no se encuentra en su forma original, sino que está representada mediante un conjunto de mapas de características de menor dimensión espacial, descritos como un arreglo tridimensional $\mathbf{Z} \in \mathbb{R}^{H' \times W' \times F}$, donde H' y W' son las dimensiones espaciales y F el número de filtros. Cada uno de los F mapas recoge una característica distinta extraída de la imagen de entrada. Sin embargo, las capas densas que realizan la clasificación final no admiten estructuras con organización espacial como entrada, por lo que es necesario transformar el conjunto de mapas de características en un vector unidimensional.

Esta transformación se realiza mediante la operación de aplanamiento (*flatten*), que reorganiza todos los valores de $\mathbf{Z}$ en un vector $\mathbf{z} \in \mathbb{R}^{H' \times W' \times F}$ sin introducir nuevos parámetros ni modificar la información contenida. Este vector representa el punto de transición entre la fase de extracción de características, basada en operaciones locales y estructura espacial, y la fase de clasificación, en la que la información se analiza mediante capas densas[225].

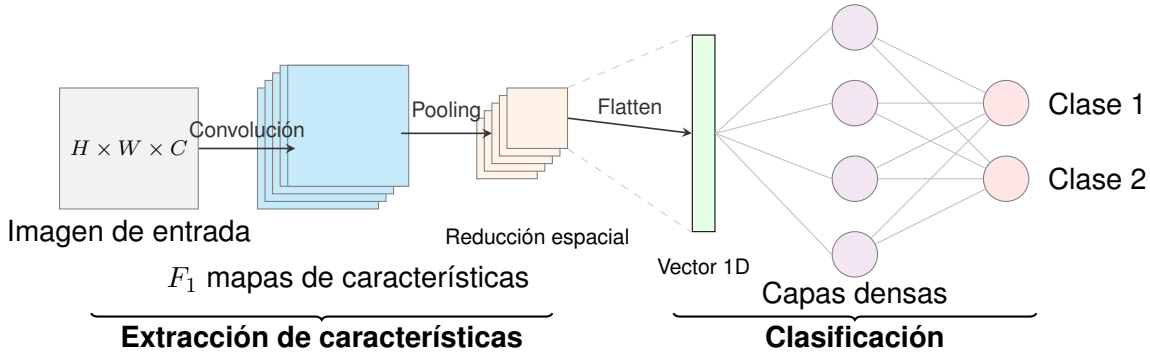
4.3.7. Arquitectura general de una red neuronal convolucional. Una red neuronal convolucional se construye a partir de la combinación ordenada de las operaciones descritas en las secciones anteriores. Una CNN se organiza como una sucesión de bloques compuestos por una capa convolucional, una función de activación no lineal y, opcionalmente, una capa de submuestreo. Estos bloques se apilan de forma jerárquica, seguidos por una o más capas totalmente conectadas encargadas de producir la salida final del modelo[226] [227].

Las primeras capas convolucionales capturan patrones locales simples como bordes o

[225] GOODFELLOW. Op. cit.
[226] Ibid.
[227] LeCUN, Yann et al. Op. cit., 1998.

variaciones de intensidad, y a medida que se avanza en profundidad, los mapas de características combinan esa información para representar estructuras más complejas. Finalmente, las capas totalmente conectadas reciben una representación vectorial de alto nivel y realizan la clasificación final [228].

Figura 7. Arquitectura general de una red neuronal convolucional.



Fuente: Elaboración propia.

Las capas convolucionales generan múltiples mapas de características a partir de la imagen de entrada. Posteriormente, las capas de submuestreo reducen su dimensión espacial. La operación de aplanamiento transforma los mapas resultantes en un vector unidimensional, que sirve como entrada para las capas densas encargadas de realizar la clasificación final.

4.3.8. Arquitecturas convolucionales profundas. El diseño de arquitecturas convolucionales profundas ha estado marcado por el incremento de la profundidad, el número de filtros o la resolución de las imágenes, pero hacerlo de forma aislada conduce a modelos costosos y poco eficientes. Para abordar esto, Mingxing Tan y Quoc Le propusieron la familia EfficientNet, un conjunto de arquitecturas que escalan simultáneamente las tres dimensiones de forma balanceada, partiendo de la hipótesis de que el desempeño mejora cuando profundidad, ancho y resolución crecen de forma conjunta y controlada [229]. Esta familia incluye variantes desde EfficientNet-B0, la arquitectura base y más simple, hasta versiones más complejas como B1, B2, B3 y superiores, cada una obtenida aplicando un mayor grado de escalamiento sobre B0. En este trabajo se utilizó EfficientNetB3 por ofrecer un equilibrio adecuado entre capa-

[228] LeCUN. Op. cit.
[229] TAN. Op. cit.

cidad de representación y costo computacional, siendo suficientemente potente para analizar imágenes médicas de alta resolución sin requerir los recursos de las variantes más grandes [230].

4.3.8.1. Bloque constructivo: MBConv. Antes de describir la arquitectura completa, es necesario presentar el bloque fundamental de EfficientNet, denominado *MBConv* [231] [232]. Este bloque es la unidad básica de procesamiento y se repite múltiples veces a lo largo de la red.

Cada bloque recibe como entrada un mapa de características $\mathbf{Z} \in \mathbb{R}^{H \times W \times C}$, donde H y W representan la altura y el ancho del mapa respectivamente, y C denota el número de canales, es decir, la cantidad de mapas de características producidos por la capa anterior, cada uno de los cuales describe una característica visual distinta de la imagen. Por ejemplo, en la primera capa de la red, los tres canales corresponden a las intensidades de los colores rojo, verde y azul de la imagen original. A partir de esta entrada, el bloque realiza las siguientes operaciones en secuencia:

1. **Expansión de canales.** Un canal es cada una de las capas de información que componen un mapa de características, en la imagen original son los tres canales de color RGB, pero en capas intermedias cada canal representa un tipo distinto de patrón detectado por un filtro. El primer paso consiste en aumentar el número de canales de C a $t \cdot C$, donde $t \geq 1$ es un factor de expansión fijo, típicamente $t = 6$. Cada canal nuevo se obtiene como una combinación lineal de los C canales de entrada:

$$Z'_m(i,j) = \sum_{c=1}^{C} w_{mc} \cdot Z_c(i,j), \quad m = 1, \ldots, tC,$$

donde tC denota el producto del factor de expansión t por el número original de canales C, es decir, el nuevo número de canales tras la expansión. Los coeficientes w_{mc} son los parámetros del filtro 1×1, que se aprenden automáticamente durante el entrenamiento. Cada canal nuevo representa una perspectiva distinta

[230] Ibid.
[231] SANDLER, Mark et al. MobileNetV2: Inverted Residuals and Linear Bottlenecks. En: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 4510-4520.
[232] TAN. Op. cit.

de la información original, determinada por los pesos que el modelo aprende a asignarle. A diferencia de los canales RGB, que tienen un significado físico fijo, estos canales nuevos no tienen una interpretación visual directa: son combinaciones de los canales originales que el modelo encontró útiles para la tarea de clasificación.

Por ejemplo, si la entrada tiene dimensiones $10 \times 10 \times 3$ y $t = 2$, la salida de este paso tiene dimensiones $10 \times 10 \times 6$: las dimensiones espaciales no cambian, solo aumenta el número de canales: $\mathbf{Z} \in \mathbb{R}^{H \times W \times C} \longrightarrow \mathbf{Z}' \in \mathbb{R}^{H \times W \times tC}$. Aumentar el número de canales permite que el bloque analice los datos desde más perspectivas antes de aplicar la convolución principal, lo que incrementa su capacidad para detectar distintos tipos de patrones visuales.

2. **Convolución separable en profundidad.** En una convolución estándar, un único filtro de tamaño $k \times k$ opera sobre todos los tC canales a la vez para producir un solo valor de salida. En cambio, la convolución separable aplica un filtro independiente de tamaño $k \times k$ a cada canal por separado [233]. Siguiendo el ejemplo anterior, si tras la expansión se tienen 6 canales, se aplican 6 filtros distintos, uno por canal.

   Si $Z'_c \in \mathbb{R}^{H \times W}$ denota el canal c del mapa expandido, el filtro $K_c \in \mathbb{R}^{k \times k}$ asociado a ese canal produce un nuevo mapa $\mathbf{A}_c$ cuyos valores se calculan como

$$\mathbf{A}_c(i,j) = \sum_{u=0}^{k-1} \sum_{v=0}^{k-1} Z'_c(i+u,\, j+v)\, K_c(u,v),$$

   donde $\mathbf{A}_c(i,j)$ indica qué tan presente está el patrón aprendido por el filtro K_c en la región de tamaño $k \times k$ del canal c centrada en la posición (i,j). El resultado $\mathbf{A}_c$ es una matriz de números del mismo tamaño espacial que el canal de entrada, donde cada valor refleja la respuesta del filtro en esa posición. Al aplicar este procedimiento a cada uno de los tC canales, se obtienen tC mapas $\mathbf{A}_1, \ldots, \mathbf{A}_{tC}$, cada uno de dimensiones $H' \times W'$, donde H' y W' son ligeramente menores que H y W debido a que el filtro no puede centrarse en los píxeles del borde de la imagen.

   Esta estrategia reduce considerablemente el número de parámetros del modelo.

[233] HOWARD, Andrew G. et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. En: *arXiv preprint arXiv:1704.04861*. 2017.

En una convolución estándar, cada filtro tiene tamaño $k \times k \times tC$ porque debe operar sobre todos los canales a la vez, y como se necesitan tC filtros distintos para producir tC canales de salida, el total de parámetros es $tC \cdot k^2 \cdot tC$. En la convolución separable, en cambio, cada filtro solo tiene tamaño $k \times k$ porque opera sobre un único canal. Aunque sigue habiendo tC filtros, el total de parámetros se reduce a $tC \cdot k^2$, un factor tC veces menor [234]. Por ejemplo, con $tC = 18$ y $k = 3$, la convolución estándar requeriría $18 \cdot 9 \cdot 18 = 2916$ parámetros, mientras que la separable solo requiere $18 \cdot 9 = 162$.

3. **Proyección de canales.** Tras la convolución separable, el resultado es un conjunto de tC mapas de características de dimensiones $H' \times W'$. Sin embargo, el número de canales necesario para la siguiente etapa de la red, denotado C', es generalmente distinto de tC. El valor de C' fue determinado por los autores durante el diseño de la arquitectura y varía según la etapa de la red, como se detalla en la Tabla 2. Para ajustar el número de canales al valor C' requerido, se aplica nuevamente una convolución 1×1 que en cada píxel combina los tC valores disponibles y produce C' valores de salida:

$$\mathbf{A} \in \mathbb{R}^{H' \times W' \times tC} \longrightarrow \mathbf{Z}_{\text{salida}} \in \mathbb{R}^{H' \times W' \times C'}.$$

La lógica del bloque completo es entonces la siguiente: primero se amplía el número de canales para aumentar la capacidad de análisis, luego se aplica la convolución espacial sobre ese espacio expandido, y finalmente se reduce el número de canales al valor necesario para la siguiente etapa de la red.

Finalmente, cuando las dimensiones de entrada y salida coinciden ($C = C'$ y $H = H'$), se añade una conexión residual que suma directamente la entrada del bloque a su salida:

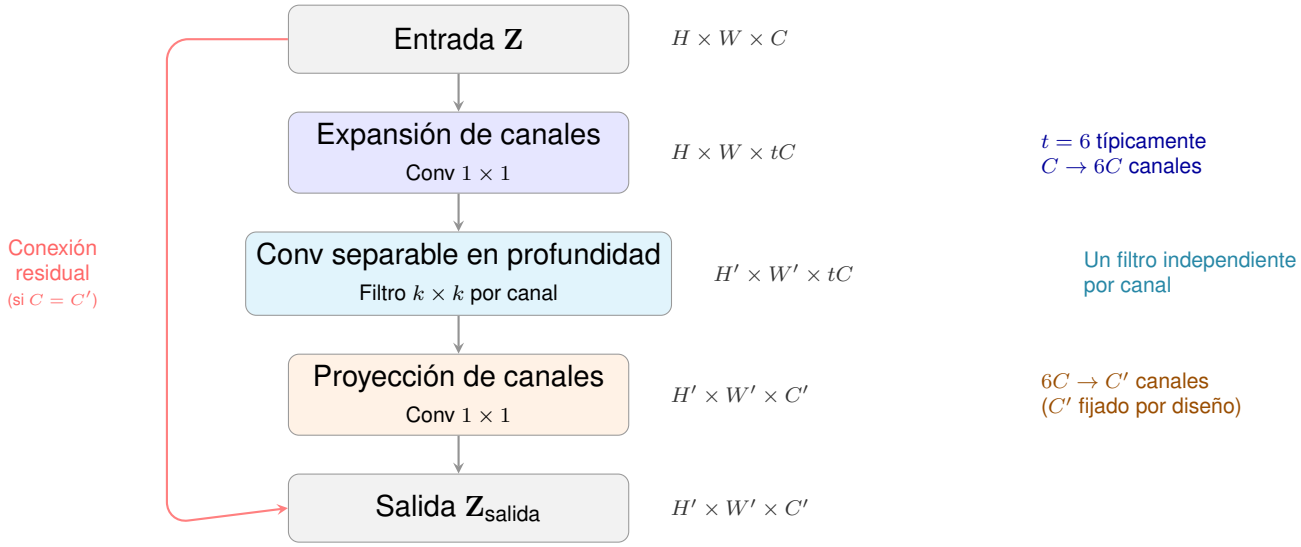
$$\mathbf{Z}_{\text{salida}} \leftarrow \mathbf{Z}_{\text{salida}} + \mathbf{Z}.$$

Esta suma tiene un propósito: si las transformaciones aplicadas dentro del bloque modifican la información de forma incorrecta o innecesaria, la entrada original sigue estando disponible y se suma al resultado, actuando como una corrección. Así, el bloque solo necesita aprender qué debe añadir o ajustar respecto a la entrada, en

[234] Ibid.

lugar de aprender toda la transformación desde cero. Esto hace que el entrenamiento sea más estable cuando la red tiene muchas capas apiladas [235] [236].

Figura 8. Estructura interna de un bloque MBConv.



Fuente: Elaboración propia.

La entrada pasa por tres operaciones secuenciales: expansión de canales, convolución separable en profundidad y proyección de canales. La conexión residual (en rojo) suma la entrada directamente a la salida cuando las dimensiones de entrada y salida coinciden.

4.3.8.2. Escalamiento compuesto. El escalamiento compuesto es el principio de diseño que da origen a toda la familia EfficientNet y explica por qué EfficientNetB3 tiene mayor capacidad que la arquitectura base EfficientNetB0. El escalamiento compuesto ya viene incorporado en la arquitectura: el usuario no lo aplica manualmente. Al diseñar una red más potente, las opciones evidentes son aumentar la profundidad agregando más etapas, ampliar su ancho aumentando el número de canales por etapa, o incrementar la resolución de las imágenes de entrada. Sin embargo, hacerlo de forma aislada produce mejoras cada vez menores: una red muy profunda pero estrecha, o muy ancha pero poco profunda, no aprovecha bien sus recursos, ya que las dimensiones que no se escalan terminan limitando el desempeño general del modelo

[235] HE, Kaiming et al. Deep Residual Learning for Image Recognition. En: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770-778.
[236] TAN. Op. cit.

sin importar cuánto se haya aumentado la otra[237] [238].

Para resolver este problema, Mingxing Tan y Quoc V. Le [239] propusieron el escalamiento compuesto, un método que escala las tres dimensiones de forma simultánea y coordinada a través de un único parámetro global denominado coeficiente de escalamiento compuesto, $\phi \geq 0$:

$$d = \alpha^{\phi}, \quad w = \beta^{\phi}, \quad r = \gamma^{\phi},$$

donde d es el factor por el que se multiplica el número de repeticiones de cada etapa, w es el factor por el que se multiplica el número de canales C' en cada etapa, y r es el factor por el que se escala la resolución de la imagen de entrada [240]. En particular, cuando $\phi = 0$ se tiene $d = w = r = 1$, es decir, no se aplica ningún escalamiento y se obtiene la arquitectura base EfficientNet-B0, la versión más simple de la familia EfficientNet, sobre la cual se construyen todas las demás variantes aplicando distintos valores de ϕ.

Los valores α, β y γ son constantes fijas determinadas por los autores, estas controlan cuánto crece cada dimensión por unidad de escalamiento. Estos valores deben satisfacer la restricción:

$$\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2.$$

Esta restricción tiene una justificación computacional concreta: el costo de una convolución crece linealmente con la profundidad, pero de forma cuadrática con el ancho y con la resolución, pues al aumentar el número de canales o el tamaño de la imagen, las operaciones crecen en ambas dimensiones simultáneamente [241] [242]. Por ello β y γ aparecen elevados al cuadrado. Bajo esta restricción, al incrementar ϕ en una unidad el costo computacional total crece de forma aproximadamente proporcional a 2^{ϕ}, lo que permite un control predecible del balance entre capacidad del modelo y recursos computacionales requeridos.

[237] GOODFELLOW. Op. cit.
[238] TAN. Op. cit.
[239] Ibid.
[240] Ibid.
[241] LeCUN, Yann et al. Op. cit., 1998.
[242] TAN. Op. cit.

Los autores determinaron los valores $\alpha = 1{,}2$, $\beta = 1{,}1$ y $\gamma = 1{,}15$ mediante una búsqueda sistemática sobre una cuadrícula de combinaciones posibles, seleccionando aquella que maximiza el desempeño de la red base bajo la restricción anterior [243]. Estos valores indican que, por unidad de escalamiento, la profundidad crece un $20\,\%$, el ancho un $10\,\%$ y la resolución un $15\,\%$. Estos valores son los mismos para toda la familia EfficientNet, lo que varía entre cada variante es únicamente el valor de ϕ. En particular, EfficientNetB3 corresponde al caso $\phi = 3$, cuya configuración resultante se detalla más adelante.

4.3.8.3. EfficientNet-B0 como arquitectura base. Aunque EfficientNet-B0 no es la arquitectura utilizada en este trabajo, su descripción es necesaria porque EfficientNetB3 se obtiene directamente a partir de ella. Su estructura se determinó mediante un proceso automatizado que explora sistemáticamente distintas combinaciones de número de bloques, tamaños de filtro y números de canales, seleccionando aquella que maximiza el desempeño bajo una restricción fija de costo computacional [244].

El resultado es una red organizada en nueve etapas secuenciales, donde cada etapa agrupa un conjunto de capas que comparten la misma configuración. La columna de repeticiones en la Tabla 2 indica cuántas veces se apila el mismo bloque dentro de cada etapa: la salida de cada bloque es la entrada del siguiente, de modo que más repeticiones implican mayor profundidad y mayor capacidad para aprender patrones complejos. Estos valores fueron fijados por los autores durante el diseño de la arquitectura.

La primera etapa consiste en una convolución estándar de tamaño 3×3 que procesa directamente la imagen de entrada, cuya resolución en EfficientNet-B0 es de 224×224 píxeles [245]. A diferencia de la convolución 1×1, que opera sobre un único píxel a la vez, una convolución 3×3 examina una región de 3×3 píxeles vecinos en cada paso, lo que permite extraer información espacial de la imagen desde el inicio.

[243] Ibid.
[244] Ibid.
[245] Ibid.

Las etapas intermedias (etapas 2 a 8) están compuestas por bloques MBConv con distintos factores de expansión t, tamaños de filtro k y números de canales de salida C', todos determinados por los autores. La columna k indica el tamaño espacial del filtro en la convolución separable en profundidad: un valor $k = 3$ significa que el filtro examina regiones de 3×3 píxeles, mientras que $k = 5$ examina regiones de 5×5 píxeles, capturando así patrones de mayor tamaño.

La etapa final combina una convolución 1×1, una operación de promedio global espacial y una capa densa de clasificación. El promedio global espacial es una operación de submuestreo que toma todos los valores de cada canal y los resume en un único número, produciendo un vector de longitud igual al número de canales. Este vector es luego procesado por la capa densa para producir la clasificación final.

La Tabla 2 resume la configuración completa de cada etapa.

Tabla 2. Configuración de las etapas de EfficientNet-B0

Etapa	**Bloque**	**Expansión t**	**Filtro k**	**Canales C'**	**Repeticiones**
1	Conv 3×3	—	3	32	1
2	MBConv	1	3	16	1
3	MBConv	6	3	24	2
4	MBConv	6	5	40	2
5	MBConv	6	3	80	3
6	MBConv	6	5	112	3
7	MBConv	6	5	192	4
8	MBConv	6	3	320	1
9	Conv 1×1 + Pool + Red Densa	—	—	—	1

Fuente: Elaboración propia.

La estructura de EfficientNet-B0 presentada en la tabla es fija: cualquier red B0 tendrá siempre esta misma configuración, independientemente del conjunto de datos sobre el que se entrene. Lo que cambia durante el entrenamiento no es la arquitectura sino los valores numéricos de los parámetros internos de cada bloque, es decir, los coeficientes

de los filtros.

4.3.8.4. EfficientNet-B3. EfficientNet-B3 se obtiene aplicando el escalamiento compuesto con $\phi = 3$ sobre la arquitectura base EfficientNet-B0. Sustituyendo en las fórmulas de escalamiento:

$$d = 1{,}2^3 \approx 1{,}73, \quad w = 1{,}1^3 \approx 1{,}33, \quad r = 1{,}15^3 \approx 1{,}52.$$

Estos factores modifican la arquitectura base de la siguiente manera: el número de repeticiones de cada etapa se multiplica por $1{,}73$, el número de canales por etapa se multiplica por $1{,}33$, y la resolución de la imagen de entrada se escala por $1{,}52$. Dado que EfficientNet-B0 fue diseñada para procesar imágenes de 224×224 píxeles, al aplicar este factor de escala los autores fijaron la resolución de entrada de EfficientNet-B3 en 300×300 píxeles [246].

La Tabla 3 muestra la configuración resultante de cada etapa, que puede compararse directamente con la Tabla 2 para apreciar el efecto concreto del escalamiento.

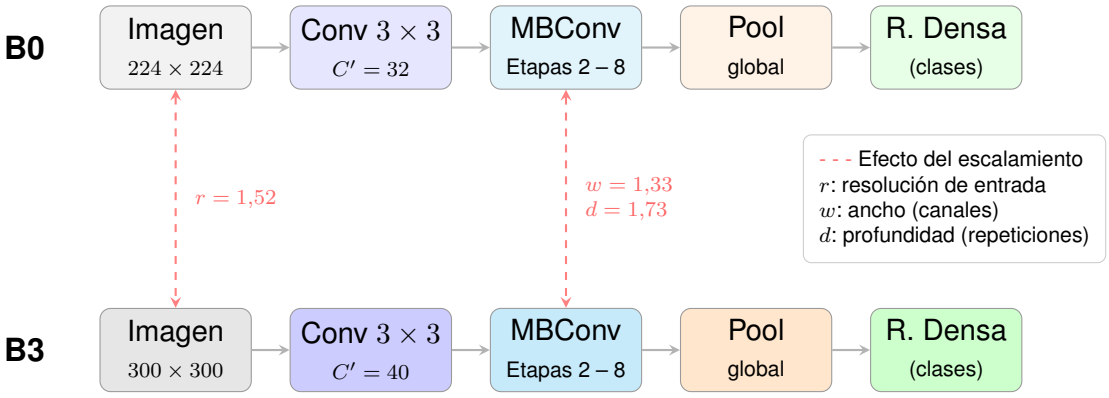
Tabla 3. Configuración de las etapas de EfficientNet-B3 obtenida aplicando el escalamiento compuesto con $\phi = 3$ sobre EfficientNet-B0.

Etapa	**Bloque**	**Expansión** t	**Filtro** k	**Canales** C'	**Repeticiones**
1	Conv 3×3	—	3	40	1
2	MBConv	1	3	24	2
3	MBConv	6	3	32	3
4	MBConv	6	5	48	3
5	MBConv	6	3	96	5
6	MBConv	6	5	136	5
7	MBConv	6	5	232	6
8	MBConv	6	3	384	2
9	Conv 1×1 + Pool + Red Densa	—	—	—	1

Fuente: Elaboración propia.

[246] Ibid.

Figura 9. Comparación del flujo de datos en EfficientNet-B0 y EfficientNet-B3.



Fuente: Elaboración propia .

Las flechas rojas indican el efecto del escalamiento compuesto con $\phi = 3$ sobre la resolución de entrada ($r = 1{,}52$), el número de canales por etapa ($w = 1{,}33$) y el número de repeticiones de cada bloque ($d = 1{,}73$).

Al comparar ambas tablas se observa que EfficientNet-B3 tiene más canales por etapa, más repeticiones de cada bloque y procesa imágenes de mayor resolución que EfficientNet-B0. Este incremento coordinado permite que EfficientNet-B3 detecte patrones visuales más complejos sin un crecimiento desproporcionado del número de parámetros, lo que la hace especialmente adecuada para tareas de clasificación de imágenes médicas, donde es necesario analizar estructuras visuales detalladas manteniendo un equilibrio entre precisión y consumo de recursos computacionales[247].

[247] Ibid.

Tabla 4. Comparación de variantes de la familia EfficientNet.

Variante	**Resolución**	**Parámetros**	ϕ	**Adecuación para imágenes médicas**
B0	224×224	5.3M	0	Resolución insuficiente para detectar estructuras finas
B1	240×240	7.8M	1	Mejora leve respecto a B0, aún limitada para patrones complejos
B2	260×260	9.2M	2	Capacidad moderada, puede perder detalles diagnósticos
B3	$\mathbf{300 \times 300}$	**12M**	**3**	**Equilibrio óptimo: resolución suficiente para estructuras diagnósticas y costo manejable**
B4	380×380	19M	4	Mayor capacidad pero costo computacional elevado
B5	456×456	30M	5	Costo muy alto para el conjunto de datos disponible
B6	528×528	43M	6	Requiere recursos computacionales muy grandes
B7	600×600	66M	7	Inviable para conjuntos de datos de tamaño moderado

Fuente: Elaboración propia.

Tabla 5. Comparación teórica de los modelos empleados en el trabajo para clasificación de imágenes médicas.

Modelo	Fortalezas	Limitaciones
EfficientNetB3	Preentrenado, preserva estructura espacial, escalamiento balanceado de profundidad, ancho y resolución.	Mayor número de parámetros, menos interpretable.
CNN	Explota estructura espacial mediante convoluciones locales, detecta patrones jerárquicos.	Sin preentrenamiento, capacidad limitada con conjuntos pequeños.
ViT	Captura dependencias globales entre regiones distantes de la imagen.	Requiere grandes volúmenes de datos, bajo desempeño con representaciones comprimidas.
Regresión Logística	Interpretable, eficiente, coeficientes explicables.	Solo aprende relaciones lineales, limitado para patrones visuales complejos.
SVM-RBF	Kernel RBF separa clases en alta dimensión, robusto ante muchas características.	No aprovecha la organización espacial de los píxeles.

Fuente: Elaboración propia.

4.4. REDES TIPO TRANSFORMER

Las redes neuronales convolucionales analizan la imagen mediante filtros locales, lo que les dificulta relacionar regiones distantes sin pasar por múltiples capas intermedias. Las redes de tipo Transformer abordan este problema de forma distinta: permiten que cualquier región de la imagen se relacione directamente con cualquier otra, independientemente de la distancia que las separe. Esto se logra mediante el denominado mecanismo de atención [248].

[248] VASWANI, Ashish et al. Attention Is All You Need. En: *Advances in Neural Information Processing Systems*. 2017, vol. 30.

4.4.1. Mecanismo de atención. El mecanismo de atención opera sobre una secuencia de N vectores $\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_N \in \mathbb{R}^D$, donde D es la dimensión de proyección. Cada vector representa un parche de la imagen si se trabaja directamente con imágenes, o una característica extraída si se trabaja con reducción de dimensionalidad. El objetivo del mecanismo es determinar, para cada vector, qué tan relevante es cada uno de los demás vectores respecto a él, de modo que cada vector pueda incorporar información de los que más le aportan.

Para ello, a partir de cada vector $\mathbf{z}_i$ se derivan tres vectores mediante transformaciones lineales cuyos parámetros son aprendidos por el modelo durante el entrenamiento:

$$\mathbf{Q}_i = \mathbf{z}_i \mathbf{W}_Q, \quad \mathbf{K}_i = \mathbf{z}_i \mathbf{W}_K, \quad \mathbf{V}_i = \mathbf{z}_i \mathbf{W}_V,$$

donde $\mathbf{W}_Q, \mathbf{W}_K, \mathbf{W}_V \in \mathbb{R}^{D \times d_k}$ son matrices de parámetros y d_k es la dimensión de los vectores resultantes. El vector $\mathbf{Q}_i$ representa la pregunta que hace el vector i al resto de la secuencia sobre qué información le resulta útil; $\mathbf{K}_j$ representa lo que el vector j ofrece como respuesta a esa pregunta; y $\mathbf{V}_j$ representa el contenido concreto que el vector j aportará si es considerado relevante [249].

4.4.1.1. Cálculo de relevancias y pesos de atención. La relevancia del vector j para el vector i se calcula como:

$$r_{ij} = \frac{\mathbf{Q}_i^\top \mathbf{K}_j}{\sqrt{d_k}},$$

donde d_k es la dimensión de cada cabeza de atención, definida en 4.4.1.2. La división por $\sqrt{d_k}$ evita que los valores crezcan demasiado [250]. Una vez calculadas todas las relevancias, estos valores se normalizan mediante la función softmax para que los pesos asociados a cada vector i sumen 1:

$$\alpha_{ij} = \frac{\exp(r_{ij})}{\sum_{l=1}^{N} \exp(r_{il})},$$

[249] Ibid.
[250] Ibid.

donde α_{ij} representa cuánto peso le da el vector i al vector j: los pesos de todos los vectores suman 1, de modo que el vector i distribuye su atención entre todos los demás según su relevancia. Para una secuencia de N vectores, se calculan en total $N \times N$ valores de relevancia.

Con estos pesos calculados, la salida del mecanismo de atención para el vector i es un nuevo vector de longitud d_k, obtenido como la suma ponderada de todos los vectores V_j:

$$\text{Atención}(\mathbf{z}_i) = \sum_{j=1}^{N} \alpha_{ij} \mathbf{V}_j.$$

Este nuevo vector incorpora información de todos los demás vectores de la secuencia, con mayor contribución de aquellos que resultaron más relevantes.

Este mecanismo es el componente central del Transformador de Visión (ViT). Al aplicarlo sobre imágenes de fondo de ojo, cada región de la imagen puede consultar directamente a cualquier otra para determinar qué información le es útil, lo que permite al modelo considerar la imagen como un todo en lugar de analizarla región por región como lo haría una CNN.

4.4.1.2. Atención multi-cabeza. En la práctica, el proceso descrito anteriormente se repite h veces en paralelo, donde h es un hiperparámetro fijado por el usuario. Cada una de estas h repeticiones se denomina cabeza de atención y dispone de sus propias matrices $\mathbf{W}_Q^{(l)}$, $\mathbf{W}_K^{(l)}$ y $\mathbf{W}_V^{(l)}$ para $l = 1, \ldots, h$, independientes entre sí y aprendidas por separado durante el entrenamiento. Esto permite que cada cabeza se especialice en detectar un tipo diferente de relación entre los vectores. La dimensión de cada cabeza se fija como $d_k = D/h$, de modo que la información total procesada se mantiene proporcional a D. Las salidas de las h cabezas se concatenan en un único vector y se combinan mediante una transformación lineal:

$$\text{Multi-cabeza}(\mathbf{z}_i) = \text{Concatenar}(\text{cabeza}_1, \ldots, \text{cabeza}_h)\, \mathbf{W}_O,$$

donde $\mathbf{W}_O \in \mathbb{R}^{hd_k \times D}$ es una matriz de parámetros aprendida que combina y reduce la información de todas las cabezas en un único vector de dimensión D, el mismo tamaño

que la entrada original. Esto permite que la salida del mecanismo multi-cabeza pueda seguir siendo procesada por las capas siguientes sin cambiar la dimensión de los datos [251].

Tras la atención multi-cabeza, cada vector pasa por una red neuronal densa de dos capas, aplicada de forma independiente a cada vector de la secuencia. Esta red no intercambia información entre vectores sino que procesa cada uno por separado, y está compuesta por una primera transformación lineal seguida de una función de activación ReLU y una segunda transformación lineal:

$$\text{Red densa}(\mathbf{z}_i) = \max(0,\, \mathbf{z}_i\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2.$$

Su función es procesar y refinar la información incorporada por el mecanismo de atención. Tanto la atención multi-cabeza como esta red densa utilizan conexiones residuales y normalización de capas, siguiendo el mismo principio descrito para los bloques MBConv [252] [253].

El conjunto formado por estos dos componentes forma un bloque Transformer, que se repite L veces en secuencia, de modo que la salida de un bloque es la entrada del siguiente y cada bloque dispone de sus propios parámetros aprendidos de forma independiente. Los primeros bloques identifican relaciones simples entre vectores y los siguientes detectan relaciones más complejas, de forma análoga a las capas de una CNN [254]. Los hiperparámetros que el usuario debe fijar son la dimensión de proyección D, el número de cabezas h y el número de bloques L.

4.4.2. Transformer sobre imágenes: ViT-B/16. El Transformador de Visión (ViT) es una arquitectura que aplica el mecanismo de atención directamente sobre imágenes, adaptando el Transformer original para procesar datos visuales [255]. La variante utilizada en este trabajo es ViT-B/16, donde B indica que se trata de la variante base y 16

[251] Ibid.
[252] HE, Kaiming. Op. cit.
[253] VASWANI. Op. cit.
[254] Ibid.
[255] DOSOVITSKIY, Alexey et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. En: *International Conference on Learning Representations (ICLR)*. 2021.

indica que cada parche mide 16×16 píxeles.

4.4.2.1. División en parches y proyección. Dado que el mecanismo de atención opera sobre secuencias de vectores, el primer paso consiste en transformar la imagen en una secuencia de vectores. Para ello, la imagen de entrada de dimensiones $224 \times 224 \times 3$ se divide en parches de 16×16 píxeles. Esto produce

$$P = \frac{224}{16} \times \frac{224}{16} = 196$$

parches, cada uno con $16 \times 16 \times 3 = 768$ valores [256]. Cada parche se aplana en un vector de longitud 768 y se proyecta mediante una transformación lineal aprendida durante el entrenamiento:

$$\mathbf{z}_i = \text{Aplanar}(\text{Parche}_i)\,\mathbf{W}_E, \quad i = 1, \ldots, P,$$

donde $\mathbf{W}_E \in \mathbb{R}^{768 \times D}$ es la matriz de proyección y $D = 768$ es la dimensión de proyección de ViT-B/16. El resultado es una secuencia de $P = 196$ vectores, uno por parche.

4.4.2.2. Token de clasificación. Para realizar la clasificación final, se añade al inicio de la secuencia un vector adicional denominado token de clasificación, denotado $\mathbf{z}_0 \in \mathbb{R}^D$. Este vector no proviene de ningún parche de la imagen, sino que es un parámetro aprendido durante el entrenamiento. Al participar en el mecanismo de atención junto con los $P = 196$ vectores de parches, este token acumula información de toda la imagen y es el que se utiliza al final para producir la clasificación. Al incluir el nuevo vector, la secuencia pasa de $P = 196$ a $N = P + 1 = 197$ vectores, que es el valor de N para esta variante [257]. El token de clasificación es apropiado para imágenes de fondo de ojo porque permite que el modelo acumule información global de toda la imagen antes de tomar una decisión diagnóstica, sin depender de una región específica. Esto es especialmente útil en el diagnóstico de enfermedades oculares, donde los indicadores visuales pueden estar distribuidos en diferentes zonas de la imagen y el modelo

[256] Ibid.
[257] Ibid.

necesita considerar la imagen como un todo para tomar una decisión correcta [258][259].

4.4.2.3. Codificación de posición. Al convertir los parches en vectores se pierde la información sobre la ubicación original de cada parche en la imagen. Para recuperarla, se suma a cada vector de parche $\mathbf{z}_i \in \mathbb{R}^D$, obtenido en la subsubsección 4.4.2.1, un *vector de posición* $\mathbf{p}_i \in \mathbb{R}^D$, el cual es aprendido durante el entrenamiento:

$$\mathbf{z}_i \leftarrow \mathbf{z}_i + \mathbf{p}_i, \quad i = 0, \ldots, P.$$

Estos vectores de posición no son fijos: el modelo aprende automáticamente qué valores asignarles para que la información de posición sea útil para la clasificación. El orden de los parches sigue la dirección natural de lectura: de izquierda a derecha en cada fila, avanzando fila por fila de arriba hacia abajo, de modo que el parche de la esquina superior izquierda ocupa la posición 1 y el de la esquina inferior derecha ocupa la posición 196 [260].

4.4.2.4. Aplicación del mecanismo de atención. Una vez construida la secuencia de $N = 197$ vectores con su información de posición, se aplica el procedimiento descrito en la subsección 4.4.1. Para cada vector se calculan las relevancias respecto a todos los demás vectores de la secuencia, se obtienen los pesos de atención mediante la función softmax, y cada vector produce una nueva versión de sí mismo tomando información de los demás. Este proceso se repite $h = 12$ veces en paralelo mediante la atención multi-cabeza, y la salida se refina mediante la red densa, formando así un bloque Transformer. En ViT-B/16, este bloque se repite $L = 12$ veces en secuencia: los primeros bloques identifican relaciones simples entre parches cercanos, y los bloques más profundos combinan esa información para detectar relaciones más complejas entre regiones distantes de la imagen. Los valores $h = 12$ y $L = 12$ son hiperparámetros independientes fijados por los autores y coinciden numéricamente por diseño [261]. La configuración completa se resume en la Tabla 6.

[258] Ibid.
[259] LITJENS. Op. cit.
[260] DOSOVITSKIY. Op. cit.
[261] Ibid.

Tabla 6. Configuración de ViT-B/16 .

Parámetro	**Valor**
Tamaño del parche	16×16 píxeles
Resolución de entrada	224×224 píxeles
Número de parches P	196
Número de vectores N	197
Dimensión de proyección D	768
Número de bloques L	12
Cabezas de atención h	12
Dimensión por cabeza d_k	64

Fuente: Elaboración propia.

4.4.2.5. Clasificación final. Tras pasar por los $L = 12$ bloques Transformer, cada uno de los 197 vectores contiene información de todos los parches de la imagen. En particular, el token de clasificación $\mathbf{z}_0$, al haber participado en el mecanismo de atención en cada uno de los 12 bloques, ha acumulado información importante de toda la imagen. Por esta razón, es este vector el que se extrae al final y se pasa a una capa densa, que produce la clasificación final asignando la imagen a una de las categorías posibles.

4.4.3. Transformer sobre vectores de características. Esta variante aplica el mecanismo de atención no sobre parches de una imagen, sino sobre las componentes obtenidas tras aplicar una técnica de reducción de dimensionalidad. A diferencia de ViT-B/16, la entrada no es una imagen cruda sino un vector $\mathbf{x} \in \mathbb{R}^n$, donde n es el número de componentes retenidas, que en esta variante corresponde al valor concreto de N definido en la subsección 4.4.1 [262].

4.4.3.1. Construcción de la secuencia. Cada componente x_i del vector de entrada se trata como un elemento independiente de la secuencia. Como cada componente

[262] VASWANI. Op. cit.

es un valor escalar y el mecanismo de atención requiere vectores, cada x_i se proyecta a un vector de dimensión $D = 64$ mediante una capa densa:

$$\mathbf{z}_i = x_i \cdot \mathbf{w}_i + \mathbf{b}_i, \quad i = 1, \ldots, n,$$

donde $\mathbf{w}_i \in \mathbb{R}^D$ y $\mathbf{b}_i \in \mathbb{R}^D$ son parámetros aprendidos durante el entrenamiento. El resultado es una secuencia de $N = n$ vectores de longitud $D = 64$, uno por cada componente de la imagen reducida, sobre la cual se aplica el mecanismo de atención [263].

4.4.3.2. Ausencia de codificación de posición. A diferencia de ViT-B/16, en esta variante no se añade codificación de posición. Esto se debe a que las componentes producidas por los métodos de reducción de dimensionalidad no tienen un orden espacial natural: a diferencia de los parches de una imagen, donde la posición relativa entre parches es informativa, el orden en que se listan las componentes reducidas no aporta información relevante para la clasificación [264].

4.4.3.3. Aplicación del mecanismo de atención. Una vez construida la secuencia de $N = n$ vectores, se aplica el mismo procedimiento descrito en la subsección 4.4.1: para cada vector se calculan las relevancias respecto a todos los demás, se obtienen los pesos de atención mediante softmax, y cada vector produce una nueva versión de sí mismo incorporando información de los demás. Este proceso se realiza con $h = 4$ cabezas de atención en paralelo, con $d_k = D/h = 16$ por cabeza, seguido de la red densa, formando un bloque Transformer. En esta variante se utiliza un único bloque Transformer ($L = 1$) [265].

4.4.3.4. Clasificación final.. En lugar del token de clasificación utilizado en ViT-B/16, esta variante emplea un promedio global. Tras el bloque Transformer, los n vectores actualizados $\mathbf{z}_1, \ldots, \mathbf{z}_n \in \mathbb{R}^D$ se resumen en un único vector tomando el promedio

[263] Ibid.
[264] Ibid.
[265] VASWANI. Op. cit.

componente a componente:

$$\bar{\mathbf{z}} = \frac{1}{n} \sum_{i=1}^{n} \mathbf{z}_i \in \mathbb{R}^D.$$

Este vector resume la información de todas las componentes y se pasa a una capa densa que produce la clasificación final. La configuración completa de esta variante se resume en la Tabla 7.

Tabla 7. Configuración del Transformer sobre vectores de características.

Parámetro	**Valor**
Número de componentes n	Variable
Número de vectores N	n
Dimensión de proyección D	64
Número de bloques L	1
Cabezas de atención h	4
Dimensión por cabeza d_k	16
Codificación de posición	No
Agregación final	Promedio global

Fuente: Elaboración propia.

5. RESULTADOS

5.1. PREPARACIÓN Y PREPROCESAMIENTO DE LOS DATOS

5.1.1. Descripción del conjunto de datos. El conjunto de datos utilizado en este trabajo es **Ocular Disease Intelligent Recognition (ODIR-5K)** [266] [267], una base de datos oftalmológica estructurada recopilada por Shanggong Medical Technology Co., Ltd. a partir de diferentes hospitales y centros médicos en China. Contiene fotografías del fondo de ojo de 5.000 pacientes, capturadas con distintas cámaras comerciales (Canon, Zeiss y Kowa), lo que introduce variabilidad natural en la resolución. Las anotaciones fueron realizadas por lectores humanos capacitados bajo un sistema de control de calidad.

Para la mayoría de los pacientes se dispone de fotografías del ojo izquierdo y del ojo derecho, cada una con su propio diagnóstico independiente. Dado que los diagnósticos de ambos ojos pueden coincidir o diferir entre sí, el conjunto de datos cuenta con un total de 6.392 imágenes, en lugar de 5.000, distribuidas en ocho categorías diagnósticas. Esta característica refleja la naturaleza real del examen oftalmológico, donde cada ojo es evaluado de forma independiente.

[266] LARXEL. Op. cit.

[267] SHANGGONG MEDICAL TECHNOLOGY CO., LTD. ODIR-2019: Ocular Disease Intelligent Recognition Dataset [en línea]. Grand Challenge, 2019. [Consultado: 2024]. Disponible en: https://odir2019.grand-challenge.org/dataset/

Tabla 8. Categorías diagnósticas del conjunto de datos ODIR-5K

Etiqueta	Enfermedad	Total de imágenes
N	Normal	2.873
D	Diabetes	1.608
O	Otras anomalías	708
C	Catarata	293
G	Glaucoma	284
A	Degeneración macular relacionada con la edad	266
M	Miopía patológica	232
H	Hipertensión	128
Total		**6.392**

Fuente: Elaboración propia.

La Tabla 8, muestra que el conjunto de datos presenta un marcado desbalanceo de clases: la categoría Normal (N) concentra el 44,9 % de las imágenes, mientras que Hipertensión (H) representa apenas el $2\,\%$. Este desbalanceo refleja la naturaleza de los datos clínicos reales y representa uno de los principales desafíos metodológicos abordados en este trabajo.

Para este problema de clasificación, las **variables explicativas** corresponden a los valores de intensidad de cada píxel de la imagen de fondo de ojo, representadas como un vector numérico $\mathbf{X}$. Dependiendo del escenario, este vector puede contener los píxeles originales de la imagen o las componentes obtenidas tras aplicar una técnica de reducción dimensional, como se detalla más adelante. La **variable respuesta** Y es la categoría diagnóstica asignada a cada imagen, una variable categórica que toma uno de los ocho valores posibles: $Y \in \{N, D, O, C, G, A, M, H\}$, según la clasificación presentada en la Tabla 8.

5.1.2. División del conjunto de datos. Para evaluar el desempeño de los modelos, el conjunto de datos fue dividido en dos subconjuntos: **entrenamiento** (80 %) y **prueba** (20 %). El subconjunto de entrenamiento es el que el modelo utiliza para aprender

los patrones de cada enfermedad, mientras que el subconjunto de prueba contiene imágenes que el modelo nunca ha visto durante el entrenamiento, lo que permite evaluar de forma objetiva su capacidad de generalización ante datos nuevos [268].

La división se realizó de forma proporcional por clase: del total de imágenes de cada categoría diagnóstica, el $80\,\%$ fue asignado al subconjunto de entrenamiento y el $20\,\%$ restante al subconjunto de prueba. Por ejemplo, si una enfermedad cuenta con 100 imágenes en total, 80 se destinan al entrenamiento y 20 a la prueba, preservando la distribución original del conjunto de datos en ambos subconjuntos. Este procedimiento es importante en conjuntos de datos desbalanceados, pues una división puramente aleatoria podría dejar clases minoritarias sin suficiente representación en el subconjunto de prueba, comprometiendo la evaluación del modelo en esas categorías [269]. La Tabla 9 muestra la distribución resultante.

Tabla 9. Distribución de imágenes por clase en los subconjuntos de entrenamiento y prueba

Etiqueta	Enfermedad	Entrenamiento	Prueba
N	Normal	2.298	575
D	Diabetes	1.286	322
O	Otras anomalías	566	142
C	Catarata	234	59
G	Glaucoma	227	57
A	Degeneración macular	213	53
M	Miopía patológica	186	46
H	Hipertensión	103	25
Total		**5.113**	**1.279**

Fuente: Elaboración propia.

5.1.3. Preprocesamiento para modelos con reducción dimensional.

[268] HASTIE. Op. cit.

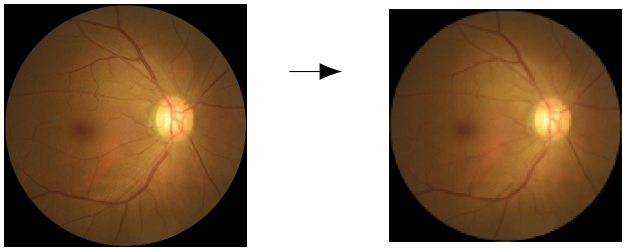
[269] JOHNSON, Justin M. y KHOSHGOFTAAR, Taghi M. Survey on deep learning with class imbalance. En: *Journal of Big Data*. 2019, vol. 6, nro. 1, pp. 1-54.

5.1.3.1. Redimensionamiento de las imágenes. El primer paso del preprocesamiento consistió en el redimensionamiento de las imágenes originales. Las imágenes del conjunto de datos ODIR-5K presentan resoluciones variables según la cámara utilizada en su captura, y fue necesario estandarizarlas a un tamaño uniforme. El objetivo es reducir la cantidad de información por procesar, lo cual se traduce en una disminución significativa tanto del uso de memoria como del tiempo de cómputo requerido durante el entrenamiento de los modelos[270].

Al disminuir la resolución de una imagen se sacrifica en cierta medida la calidad visual, ya que se pierde parte del detalle presente en la versión original. Sin embargo, diversos estudios han demostrado que existe un equilibrio entre la resolución de entrada y el costo computacional: resoluciones más altas incrementan el número de parámetros a optimizar y el uso de memoria, lo que a su vez limita la cantidad de imágenes que pueden procesarse simultáneamente durante el entrenamiento, es decir, obliga a trabajar con lotes más pequeños, lo que puede ralentizar el proceso de aprendizaje [271]. En este trabajo se seleccionó un tamaño de 128×128 píxeles, una resolución ampliamente utilizada en tareas de clasificación de imágenes médicas [272] [273], que conserva las estructuras visuales relevantes del fondo de ojo con un costo computacional manejable.

La Figura 10 ilustra la diferencia visual entre la imagen original y la imagen redimensionada a 128×128 píxeles.

Figura 10. Imagen original del conjunto de datos ODIR-5K[274] redimensionada a 128×128 píxeles.



Fuente: Elaboración propia.

[270] GOODFELLOW. Op. cit.
[271] SABOTTKE, Carl F. y SPIELER, Bradley M. The Effect of Image Resolution on Deep Learning in Radiography. En: *Radiology: Artificial Intelligence*. 2020, vol. 2, nro. 1, p. e190015.
[272] RUKUNDO, Olivier. Effects of Image Size on Deep Learning. En: *arXiv preprint arXiv:2101.11508*. 2021.
[273] SABOTTKE. Op. cit.

5.1.3.2. Aplanamiento y construcción de la matriz de datos. Una vez redimensionadas, cada imagen queda representada como una matriz de 128×128 píxeles, donde cada píxel contiene tres valores de intensidad correspondientes a los canales Rojo (R), Verde (G) y Azul (B):

$$\text{Imagen} = \begin{bmatrix} [R_{1,1},\ G_{1,1},\ B_{1,1}] & [R_{1,2},\ G_{1,2},\ B_{1,2}] & \cdots & [R_{1,128},\ G_{1,128},\ B_{1,128}] \\ [R_{2,1},\ G_{2,1},\ B_{2,1}] & [R_{2,2},\ G_{2,2},\ B_{2,2}] & \cdots & [R_{2,128},\ G_{2,128},\ B_{2,128}] \\ \vdots & \vdots & \ddots & \vdots \\ [R_{128,1},\ G_{128,1},\ B_{128,1}] & \cdots & \cdots & [R_{128,128},\ G_{128,128},\ B_{128,128}] \end{bmatrix}$$

donde $R_{r,c}$, $G_{r,c}$ y $B_{r,c}$ denotan los valores de intensidad del píxel en la fila r y columna c, con valores entre 0 y 255.

La Regresión Logística, el SVM y las técnicas de reducción dimensional requieren que cada observación sea un vector de números individuales, por lo que se aplica el proceso de aplanamiento: recorrer la matriz fila por fila, separando cada triplete $[R_{r,c},\ G_{r,c},\ B_{r,c}]$ en tres valores independientes y concatenándolos en una única fila continua [275] [276].

El vector resultante para cada imagen queda:

$$[\underbrace{R_{1,1}\ G_{1,1}\ B_{1,1}}_{\text{(r=1,c=1)}}, \underbrace{R_{1,2}\ G_{1,2}\ B_{1,2}}_{\text{(r=1,c=2)}}, \ldots, \underbrace{R_{1,128}\ G_{1,128}\ B_{1,128}}_{\text{(r=1, c=128)}}, \underbrace{R_{2,1}\ G_{2,1}\ B_{2,1}}_{\text{(r=2, c=1)}}, \ldots, \underbrace{R_{128,128}\ G_{128,128}\ B_{128,128}}_{\text{(r=128, c=128)}}]$$

Con tres canales de color por imagen, el número total de características es $128 \times 128 \times 3 = 49.152$ características.

Reuniendo las 6.392 imágenes aplanadas se construye la matriz de datos de tamaño $\mathbb{R}^{6392 \times 49152}$, donde cada fila corresponde a una imagen y cada columna corresponde a la intensidad de un canal de color en un píxel específico de la imagen.

[274] `https://www.kaggle.com/datasets/andrewmvd/ocular-disease-recognition-odir5k`
[275] GONZALEZ, Rafael C. y WOODS, Richard E. *Digital Image Processing*. 4 ed. Pearson, 2018.
[276] GOODFELLOW. Op. cit.

5.1.3.3. Estandarización de los datos. Los datos fueron estandarizados mediante `StandardScaler` [277], para que cada característica tenga media 0 y desviación estándar 1, de modo que ningún píxel tenga mayor peso que otro por encontrarse en una escala de valores más alta.

El proceso opera de forma independiente sobre cada una de las 49.152 columnas de la matriz de datos, donde cada columna representa un píxel específico a través de todas las imágenes de entrenamiento. Para cada columna j, se calcula su media μ_j y su desviación estándar σ_j, y se aplica la transformación:

$$z_{i,j} = \frac{x_{i,j} - \mu_j}{\sigma_j}, \qquad \mu_j = \frac{1}{n}\sum_{i=1}^{n} x_{i,j}, \qquad \sigma_j = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x_{i,j} - \mu_j)^2}$$

donde $x_{i,j}$ es el valor original del píxel j en la imagen i y $z_{i,j}$ es su valor estandarizado. Esta estandarización es necesaria para PCA y AF, cuyo funcionamiento se basa en la matriz de covarianza de los datos[278]: si los píxeles no estuvieran estandarizados, aquellos con mayor varianza dominarían los primeros componentes o factores, independientemente de su relevancia informativa.

Los parámetros μ_j y σ_j se estimaron exclusivamente sobre el subconjunto de entrenamiento y se aplicaron al subconjunto de prueba [279].

5.1.4. Preprocesamiento para modelos sin reducción dimensional. A diferencia del enfoque con reducción dimensional, en esta sección las imágenes se utilizan en su forma completa, sin pasar por ninguna técnica de compresión de información. Sin embargo, el preprocesamiento varía según el tipo de modelo, por lo que se distinguen dos flujos de trabajo.

[277] PEDREGOSA, Fabian et al. Scikit-learn: Machine Learning in Python. En: *Journal of Machine Learning Research*. 2011, vol. 12, pp. 2825-2830.
[278] JOLLIFFE. Op. cit.
[279] HASTIE. Op. cit.

5.1.4.1. Modelos clásicos. Para la Regresión Logística y SVM-RBF, las imágenes fueron redimensionadas a 128×128 píxeles, manteniendo el mismo tamaño utilizado en la sección de reducción dimensional para facilitar la comparación entre ambos enfoques. Al igual que en la sección anterior (véase 5.1.3.2), cada imagen fue aplanada en un vector de $128 \times 128 \times 3 = 49.152$ características. Sin embargo, en lugar de aplicar StandardScaler, los valores fueron normalizados al rango $[0, 1]$ dividiendo entre 255, dado que en este caso no se aplican técnicas de reducción dimensional[280].

5.1.4.2. Modelos de aprendizaje profundo. Para las arquitecturas de aprendizaje profundo CNN y ViT las imágenes fueron redimensionadas a 224×224 píxeles, mientras que EfficientNetB3 requiere imágenes de 300×300 píxeles, de acuerdo con las especificaciones de su arquitectura original [281].

Carga y generación de datos para modelos de aprendizaje profundo: A diferencia de los modelos con reducción dimensional, donde fue posible cargar la totalidad de las imágenes en memoria de forma simultánea gracias a su representación como vectores aplanados, las arquitecturas de aprendizaje profundo trabajan con imágenes completas de dimensiones considerablemente mayores ($224 \times 224 \times 3$ y $300 \times 300 \times 3$). Cargar todas las imágenes del conjunto de datos en memoria RAM de forma simultánea resultaría computacionalmente inviable[282]. Por tanto, se implementaron generadores de datos mediante la clase `ImageDataGenerator` de Keras [283], que cargan las imágenes en lotes (*batches*) durante el entrenamiento en lugar de hacerlo todas a la vez.

Procesamiento por lotes: El procesamiento por lotes consiste en dividir el conjunto de entrenamiento en grupos de imágenes de tamaño fijo denominados *batches* que el modelo procesa uno a uno durante cada época: primero aprende del lote 1, actualiza sus parámetros, luego procesa el lote 2, vuelve a actualizarse, y así hasta recorrer

[280] GOODFELLOW. Op. cit.
[281] TAN. Op. cit.
[282] GOODFELLOW. Op. cit.
[283] CHOLLET, François et al. Keras [en línea]. 2015. Disponible en: https://keras.io

todas las imágenes. En este trabajo se utilizó un tamaño de lote de 32 imágenes para CNN y ViT, y de 16 imágenes para EfficientNetB3, dado su mayor tamaño de entrada. Este enfoque permite entrenar modelos con conjuntos de datos que no caben completamente en memoria y, además, como cada lote contiene imágenes distintas, las actualizaciones de los parámetros del modelo varían en cada paso, lo que le ayuda a explorar mejor el espacio de soluciones y evitar que el modelo aprenda patrones demasiado específicos del conjunto de entrenamiento sin poder generalizarlos a datos nuevos [284].

Aumentación de datos: Adicionalmente, los generadores implementan aumentación de datos (*data augmentation*) durante el entrenamiento, técnica que consiste en aplicar transformaciones aleatorias a las imágenes originales para generar versiones artificialmente modificadas de las mismas [285].

Las transformaciones aplicadas durante el entrenamiento fueron: Rotación aleatoria de hasta $20°$, simulando variaciones en la orientación de la cámara durante la captura del fondo de ojo; Desplazamiento horizontal y vertical de hasta el $10\,\%$ del tamaño de la imagen, simulando pequeñas variaciones en el posicionamiento del paciente; y Volteo horizontal aleatorio, válido en imágenes de fondo de ojo dado que las estructuras oculares son aproximadamente simétricas [286].

La aumentación de datos se aplicó exclusivamente al conjunto de entrenamiento. El conjunto de prueba fue procesado únicamente con la normalización correspondiente, para que la evaluación se realice sobre imágenes en su estado original [287].

Estandarización de datos: Para los modelos CNN y ViT se aplicó la normalización estándar, es decir,

$$x_{norm} = \frac{x}{255}$$

[284] GOODFELLOW. Op. cit.
[285] SHORTEN, Connor y KHOSHGOFTAAR, Taghi M. A survey on Image Data Augmentation for Deep Learning. En: *Journal of Big Data*. 2019, vol. 6, nro. 1, pp. 1-48.
[286] Ibid.
[287] Ibid.

Esta normalización es la convención estándar para redes neuronales profundas[288], y es necesaria cuando se utilizan modelos preentrenados, cuyos pesos fueron optimizados con imágenes en esta escala. A diferencia de la estandarización aplicada en la subsección anterior, la normalización por 255 preserva los valores no negativos de los píxeles, condición necesaria para el correcto funcionamiento de funciones de activación como ReLU[289].

Las imágenes mantienen su estructura tridimensional $H \times W \times 3$, donde H y W corresponden al alto y ancho respectivamente sin ser aplanadas, ya que las redes neuronales convolucionales y los transformers están diseñados para explotar precisamente la estructura espacial de las imágenes [290] [291].

Para el caso puntual de EfficientNetB3, se utilizó la función de preprocesamiento específica de su arquitectura (`preprocess_input`). Esto se debe a que EfficientNetB3 fue preentrenado sobre **ImageNet** [292], un conjunto de datos masivo de más de 14 millones de imágenes utilizado para entrenar modelos de visión por computador y sus pesos internos fueron optimizados esperando imágenes con una escala y centrado específicos. Aplicar una normalización diferente alteraría la distribución de los valores de entrada y degradaría el rendimiento del modelo preentrenado [293].

5.1.5. Etiquetado de las imágenes. Tanto para los modelos con reducción dimensional como para los modelos sin reducción, cada imagen debe estar asociada a su categoría diagnóstica para que el modelo pueda aprender a distinguir entre enfermedades. Esta asociación se construyó a partir del archivo `full_df.csv` [294], que contiene el nombre de cada imagen y su diagnóstico correspondiente. Mediante un diccionario, cada nombre de archivo fue emparejado con su etiqueta, de forma que al cargar una imagen, el modelo identifica inmediatamente a qué categoría pertenece.

[288] GOODFELLOW. Op. cit.
[289] Ibid.
[290] LeCUN, Yann et al. Op. cit., 1998.
[291] VASWANI. Op. cit.
[292] DENG, Jia et al. ImageNet: A large-scale hierarchical image database. En: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009, pp. 248-255.
[293] TAN. Op. cit.
[294] LARXEL. Op. cit.

Sin embargo, los algoritmos de clasificación requieren entradas numéricas para realizar sus operaciones internas, como el cálculo de distancias, productos punto y gradientes, por lo que no pueden operar directamente sobre cadenas de texto [295]. Por esta razón, las etiquetas originales `N`, `D`, `G`, `C`, `A`, `H`, `M` y `O` fueron convertidas a valores numéricos enteros mediante un `LabelEncoder` [296], que asigna un número único a cada categoría de forma automática siguiendo orden alfabético. La Tabla 10 muestra la correspondencia resultante.

Tabla 10. Codificación numérica de las categorías diagnósticas

Código numérico	Etiqueta	Enfermedad
0	A	Degeneración macular relacionada con la edad
1	C	Catarata
2	D	Diabetes
3	G	Glaucoma
4	H	Hipertensión
5	M	Miopía patológica
6	N	Normal
7	O	Otras anomalías

Fuente: Elaboración propia.

Así, a cada imagen le corresponde un número entre 0 y 7 que indica la enfermedad que presenta. Esta codificación se aplica de igual forma en ambas secciones del trabajo: modelos con reducción dimensional y modelos sin reducción dimensional [297].

5.1.6. Manejo del desbalanceo de clases. Como se mostró en la Tabla 8, el conjunto de datos ODIR-5K presenta un marcado desbalance entre clases, lo que puede llevar a los modelos a ignorar las categorías minoritarias durante el entrenamiento [298]. Para abordar este problema, se adoptó la estrategia de **pesos de clase** (*class weight*), que asigna una penalización mayor al error cometido en las clases minoritarias durante el entrenamiento. En términos prácticos, cuando el modelo clasifica incorrectamente

[295] HASTIE. Op. cit.
[296] PEDREGOSA. Op. cit.
[297] HASTIE. Op. cit.
[298] JOHNSON, Justin M. Op. cit.

una imagen de una clase poco frecuente, ese error cuenta más que equivocarse en una clase mayoritaria, obligando al modelo a prestar más atención a las enfermedades menos representadas. La penalización asignada a cada clase c se calcula como:

$$w_c = \frac{n}{k \cdot n_c}$$

donde n es el número total de imágenes de entrenamiento, k es el número de clases y n_c es el número de imágenes de la clase c. Así, clases con pocos ejemplos (n_c pequeño) reciben un peso w_c mayor, mientras que clases frecuentes (n_c grande) reciben un peso menor. Por ejemplo, Hipertensión (H) con 103 imágenes recibirá un peso considerablemente mayor que Normal (N) con 2.298 imágenes.

Esta estrategia fue aplicada tanto en los modelos con reducción dimensional como en los modelos sin reducción dimensional sin modificar la distribución original de los datos ni generar imágenes artificiales, con lo que se preservan las proporciones reales del conjunto de datos clínico [299]. Previamente se exploró la técnica SMOTE (*Synthetic Minority Oversampling Technique*), que genera muestras sintéticas de las clases minoritarias para equilibrar el conjunto de datos [300]. Sin embargo, los resultados no mostraron mejoras significativas respecto a los pesos de clase, por lo que se optó por pesos de clase como estrategia principal.

5.2. MÉTRICAS DE EVALUACIÓN

Para evaluar el desempeño de los modelos de clasificación se utilizaron las siguientes métricas [301]:

Exactitud (*Accuracy*): proporción de predicciones correctas sobre el total de predicciones realizadas [302]. Es decir, de todas las imágenes evaluadas, ¿cuántas clasificó

[299] Ibid.
[300] CHAWLA, N. V.; BOWYER, K. W.; HALL, L. O. y KEGELMEYER, W. P. SMOTE: Synthetic Minority Over-sampling Technique. En: *Journal of Artificial Intelligence Research*. 2002, vol. 16, pp. 321-357.
[301] HASTIE. Op. cit.
[302] Ibid.

correctamente el modelo?

$$\text{Exactitud} = \frac{\text{Verdaderos Positivos} + \text{Verdaderos Negativos}}{\text{Total de observaciones}}$$

Precisión (*Precision*): proporción de predicciones positivas del modelo para una clase que fueron realmente correctas [303]. Es decir, de todas las imágenes que el modelo clasificó como pertenecientes a una clase, ¿cuántas realmente lo eran?

$$\text{Precisión} = \frac{\text{Verdaderos Positivos}}{\text{Verdaderos Positivos} + \text{Falsos Positivos}}$$

Sensibilidad (*Recall*): proporción de los casos reales de una clase que el modelo logró identificar correctamente [304]. Es decir, de todas las imágenes que realmente pertenecen a una clase, ¿cuántas logró detectar el modelo?

$$\text{Recall} = \frac{\text{Verdaderos Positivos}}{\text{Verdaderos Positivos} + \text{Falsos Negativos}}$$

F1-score: media armónica entre la Precisión y el Recall, que pone en un solo valor el equilibrio entre ambas métricas. Un F1-score alto indica que el modelo tiene tanto una buena precisión como un buen recall para esa clase. Es especialmente útil en conjuntos de datos desbalanceados, donde la exactitud global puede ser engañosa [305].

$$\text{F1-score} = 2 \times \frac{\text{Precisión} \times \text{Recall}}{\text{Precisión} + \text{Recall}}$$

Support (tamaño real de cada clase): número real de imágenes de cada clase en el conjunto de prueba. Este valor no cambia entre modelos y permite contextualizar los resultados, dado que las clases con support bajo son más difíciles de clasificar correctamente [306].

[303] Ibid.
[304] Ibid.
[305] JOHNSON, Justin M. Op. cit.
[306] HASTIE. Op. cit.

Macro avg (promedio macro): promedio simple de las métricas entre todas las clases, sin considerar el número de imágenes por clase. Refleja el desempeño del modelo de forma equitativa entre clases y penaliza cuando el modelo ignora las categorías minoritarias [307].

Matriz de confusión: tabla que muestra cómo el modelo clasifica cada imagen. Las filas representan las clases reales y las columnas representan las clases predichas por el modelo. Los valores en la diagonal principal corresponden a las predicciones correctas y los valores fuera de la diagonal indican los errores de clasificación. Por ejemplo, si en la fila D (Diabetes) y columna N (Normal) aparece el valor 126, significa que 126 imágenes de pacientes con Diabetes fueron incorrectamente clasificadas como Normal por el modelo [308].

Curva de pérdida y exactitud (*Loss and accuracy curves*): muestran cómo evolucionan el error y la exactitud del modelo a lo largo de las épocas de entrenamiento, tanto para el conjunto de entrenamiento como para el de validación. Cuando la curva de entrenamiento mejora pero la de validación se estanca o empeora, se produce el **sobreajuste** (*overfitting*), indicando que el modelo memoriza los datos de entrenamiento en lugar de aprender patrones generalizables. Valores bajos en ambas curvas indican **subajuste** (*underfitting*), es decir, que el modelo no captura suficientemente bien los patrones del conjunto de datos[309].

5.3. RESULTADOS DE MODELOS CON REDUCCIÓN DIMENSIONAL

En esta sección se presentan los resultados de clasificación sobre representaciones reducidas, organizados en tres etapas: verificación de la adecuación de los datos mediante el criterio KMO para aplicar el AF; selección del número óptimo de componentes para los métodos de reducción dimensional; y evaluación de los modelos bajo dos condiciones: sin balanceo de clases y con pesos de clase.

[307] Ibid.
[308] Ibid.
[309] GOODFELLOW. Op. cit.

5.3.1. Exploración de modelos y criterios de selección. Como punto de partida, se evaluaron 20 modelos de clasificación sobre las representaciones reducidas, con el objetivo de identificar cuáles mostraban mayor potencial sobre este conjunto de datos antes de realizar un análisis más profundo. Esta exploración incluyó desde modelos clásicos como Regresión Logística, SVM y KNN, hasta métodos de ensamble como Random Forest y Gradient Boosting, y modelos de aprendizaje profundo como CNN, Transformer y MLP. Los resultados completos se presentan en las Tablas 11, 12 y 13

Tabla 11. Exactitud de modelos de aprendizaje profundo con reducción dimensional, sin balanceo

Modelo	**PCA**	**AF**	**UMAP**
CNN 1D	0,4832	0,4738	0,4605
Transformer	0,4535	0,4519	0,4535
MLP (TensorFlow)	0,5059	0,4980	0,4582
MLP (Sklearn)	0,4535	0,4894	0,4535

Fuente: Elaboración propia.

Tabla 12. Exactitud de modelos clásicos con reducción dimensional, sin balanceo

Modelo	**PCA**	**AF**	**UMAP**
Regresión Logística	0,4441	0,4558	0,4511
SVM Lineal	0,2729	0,2721	0,4511
SVM RBF	0,4855	0,4754	0,4527
SVM Polinómico	0,4613	0,4394	0,4550
SVM Sigmoide	0,3573	0,3229	0,3057
LDA	0,4738	0,4629	0,4378
QDA	0,2158	0,2275	0,4253
KNN Uniforme	0,3823	0,4159	0,3925
KNN Distancia	0,4058	0,4269	0,3839
Centroide	0,1493	0,2455	0,1439
Gaussian NB	0,3041	0,2502	0,2776
Complement NB	0,3964	0,3659	0,3753
Árbol de Decisión	0,4339	0,4113	0,3909
Random Forest	0,4824	0,4816	0,4355
Gradient Boosting	0,4722	0,4910	0,4128
AdaBoost	0,4339	0,4246	0,4277

Fuente: Elaboración propia.

Tabla 13. Exactitud de modelos con reducción dimensional, con balanceo

Modelo	**Condición**	**PCA**	**AF**	**UMAP**
Regresión Logística	Class Weight	0.1548	0.2103	0.1509
SVM Lineal	Class Weight	0.2854	0.2869	0.1126
SVM RBF	Class Weight	0.2752	0.3534	0.1274
SVM Polinómico	Class Weight	0.2877	0.3354	0.1282
SVM Sigmoide	Class Weight	0.0516	0.0837	0.0727
CNN 1D	Class Weight	0.3346	0.3053	0.2518
Transformer	Class Weight	0.2111	0.2260	0.3761
MLP (TensorFlow)	Class Weight	0.2048	0.2166	0.1611
LDA*	Priors Iguales	0.2565	0.2471	0.1517
Árbol de Decisión	Class Weight	0.1470	0.1470	0.1446
Random Forest	Class Weight	0.4738	0.4793	0.4019
Gradient Boosting	Class Weight	0.4347	0.4425	0.3471
QDA*	Priors Iguales	0.1587	0.1392	0.0938

* LDA y QDA no permiten asignar pesos de clase. Como alternativa, se asignó una probabilidad de $\frac{1}{8}$ a cada clase, de modo que todas las enfermedades tuvieran la misma importancia al momento de clasificar.

Fuente: Elaboración propia.

Con base en estos resultados se seleccionaron cuatro modelos para un análisis detallado: Regresión Logística, SVM-RBF, CNN y ViT. Esta selección busca cubrir enfoques muy distintos de clasificación, desde modelos lineales simples hasta arquitecturas de aprendizaje profundo, lo que permite obtener una visión más completa de cómo diferentes tipos de modelos se comportan sobre el mismo problema[310] [311]. Los mismos cuatro modelos fueron evaluados en la sección sin reducción dimensional, incorporando adicionalmente EfficientNetB3 para explorar el efecto del conocimiento previo sobre los resultados [312]. Los resultados detallados de todos los modelos explorados están disponibles en el repositorio del proyecto[313].

[310] Ibid.
[311] HASTIE. Op. cit.
[312] TAN. Op. cit.
[313] `https://bit.ly/notebook-ODIR5K`

5.3.2. Justificación estadística para la aplicación del análisis factorial. Como se describió en el capítulo anterior 3.5.1, antes de aplicar el AF es necesario verificar que los datos presenten una estructura de correlación adecuada mediante la medida KMO. El valor obtenido para el conjunto de datos fue:

$$\text{KMO} = 0{,}9796$$

Este valor se clasifica como **Excelente** según los criterios de Kaiser (KMO $> 0{,}90$) [314], lo que indica que todas las variables presentan correlaciones adecuadas para el análisis. Estos resultados confirman que los datos son estadísticamente adecuados para la aplicación del AF, resultado esperado dado que en las imágenes de fondo de ojo los píxeles cercanos entre sí tienden a compartir información visual similar, lo que genera naturalmente altas correlaciones entre variables [315].

5.3.3. Selección del número de componentes y factores. Para determinar el número de componentes o factores a retener en cada técnica de reducción, se aplicaron dos criterios: el **Análisis Paralelo de Horn** y el criterio de **Yeomans-Golder**. Adicionalmente, se estableció como requisito metodológico conservar al menos el $94\,\%$ de la varianza acumulada, para garantizar que la reducción preservara la información esencial de las imágenes originales. Cuando ambos criterios arrojaron valores distintos, se adoptó el valor máximo entre ellos como criterio conservador.

5.3.3.1. Análisis de Componentes Principales (PCA). El Análisis Paralelo de Horn se ejecutó en tres repeticiones, sugiriendo retener entre 29 y 30 componentes para superar el umbral del $94\,\%$ de varianza acumulada. El criterio de Yeomans-Golder, por su parte, identificó 53 componentes con autovalor $\lambda > 1$; sin embargo, al aplicar el requisito del $94\,\%$ de varianza, se determinó que entre 24 y 25 componentes son suficientes. Adoptando el valor máximo entre ambos criterios, se fijó el número de componentes en $k = 30$.

[314] KAISER, Henry F. An index of factorial simplicity. En: *Psychometrika*. 1974, vol. 39, nro. 1, pp. 31-36.
[315] HAIR. Op. cit.

Los primeros componentes concentran la mayor parte de la varianza: el primer componente explica aproximadamente el $41\,\%$ de la varianza total y la curva acumulada alcanza el $94,74\,\%$ con los 30 componentes retenidos. La reducción lograda es significativa: se pasa de 49.152 características originales a únicamente 30 componentes, representando una compresión del $99{,}9\,\%$. Para verificar si un mayor número de componentes mejoraba el desempeño de los modelos, se realizaron pruebas exploratorias con 50 y 100 componentes; sin embargo, los resultados no mostraron mejoras significativas respecto a los 30 componentes, por lo que se mantuvo este valor para los tres métodos de reducción.

5.3.3.2. Análisis Factorial (AF). Dado que PCA y AF parten de la misma matriz de datos, los criterios arrojaron resultados casi idénticos, fijando el número de factores en $k = 30$.

La calidad de la reducción factorial se evaluó mediante las comunalidades, que indican qué proporción de la varianza de cada variable es explicada por los factores retenidos [316]. La gran mayoría de las variables presenta comunalidades superiores a $0{,}90$, con un promedio de $0{,}9427$, una comunalidad máxima de $1{,}0000$ y una mínima de $0{,}2589$. Estos valores indican que los 30 factores retenidos capturan adecuadamente la estructura de varianza de los datos originales. Al igual que en PCA, la reducción lograda es del $99{,}9\,\%$, pasando de 49.152 variables a 30 factores.

5.3.3.3. Aproximación y proyección uniforme de variedades (UMAP). Para UMAP no existe una regla matemática formal para determinar el número óptimo de componentes[317]. Se adoptaron 30 componentes para mantener consistencia metodológica con PCA y AF, facilitando la comparación directa entre los tres métodos. Los parámetros de configuración utilizados fueron: número de vecinos $= 30$ (aproximadamente el $1\,\%$ de las muestras de entrenamiento, garantizando un balance entre estructura local y global), distancia mínima $= 0{,}0$ (favoreciendo agrupamiento compacto) y métrica euclidiana. Al igual que en PCA y AF, se evaluaron configuraciones con 50 y 100

[316] HAIR, J. et al. *Multivariate Data Analysis*. Cengage, 2019.
[317] McINNES. Op. cit.

componentes sin obtener mejoras significativas.

La calidad de la reducción fue evaluada mediante las métricas de **Confiabilidad** (*Trustworthiness*) y **Continuidad** (*Continuity*), descritas en 3.7.1 y evaluadas para distintos valores de k vecinos. Los resultados se presentan en la Tabla 14.

Tabla 14. Confiabilidad y Continuidad de la reducción UMAP para distintos valores de vecinos

k **vecinos**	**Confiabilidad**	**Continuidad**
5	0,9801	0,9861
10	0,9737	0,9810
15	0,9701	0,9777
20	0,9671	0,9751
30	0,9623	0,9707
50	0,9530	0,9628
Promedio	**0,9677**	**0,9755**

Fuente: Elaboración propia.

Los resultados obtenidos muestran valores promedio de $0{,}9677$ para la Confiabilidad y $0{,}9755$ para la Continuidad, ambos muy por encima del umbral de referencia de $94\,\%$. Esto indica que la reducción de 49.152 dimensiones a 30 componentes no distorsiona significativamente la estructura de los datos: los puntos que eran similares entre sí en el espacio original siguen siendo similares en el espacio reducido, y viceversa[318].

5.3.4. Entropía de Shannon. La entropía de Shannon cuantifica qué tan distribuida está la información entre los componentes de la representación reducida [319]: una entropía alta indica que ningún componente domina sobre los demás, mientras que una entropía baja señala concentración de información en pocos componentes.

[318] Ibid.

[319] SHANNON, Claude E. A Mathematical Theory of Communication. En: *Bell System Technical Journal*. 1948, vol. 27, nro. 3, pp. 379-423.

5.3.4.1. Fundamento matemático. Dado un vector $\mathbf{x} \in \mathbb{R}^n$ asociado a la representación reducida de las imágenes, la entropía de Shannon se calcula como:

$$H(\mathbf{x}) = -\sum_{i=1}^{n} p_i \log(p_i)$$

donde p_i representa el peso relativo de cada componente del vector, calculado como:

$$p_i = \frac{|x_i|}{\sum_{j=1}^{n} |x_j|}$$

La entropía alcanza su valor máximo cuando todos los componentes tienen el mismo peso, es decir, cuando $p_i = \frac{1}{n}$ para todo i. En ese caso:

$$H_{\text{máx}} = -\sum_{i=1}^{n} \frac{1}{n} \log\left(\frac{1}{n}\right) = \log(n)$$

Este valor teórico sirve como referencia: cuanto más cercana sea la entropía obtenida a $H_{\text{máx}}$, mejor preserva el método la diversidad informativa de los datos originales [320].

Cálculo sobre el conjunto de prueba. El cálculo se realizó sobre el conjunto de prueba ($n = 1{,}279$ imágenes) por dos razones: estos datos no participaron en ninguna etapa del entrenamiento ni de la reducción dimensional, lo que permite una evaluación objetiva; y calcular la entropía sobre los datos de entrenamiento podría sobreestimar la calidad de la reducción, dado que los métodos fueron ajustados precisamente sobre esos datos [321].

Para n = 1.279 muestras de prueba, la entropía máxima teórica es:

$$H_{\text{máx}} = \log(1.279) = 7{,}1538 \text{ nats}$$

[320] Ibid.
[321] HASTIE. Op. cit.

donde los resultados se expresan en **nats**, unidad de medida de la entropía cuando se emplea el logaritmo natural en su cálculo.

Los resultados obtenidos se presentan en la Tabla 15

Tabla 15. Entropía de Shannon por método de reducción dimensional

Método	**Entropía**	**% de H_{max}**
AF	6,6922	93,55 %
PCA	6,7654	94,57 %
UMAP	6,8846	96,24 %
H_{max} (referencia)	7,1538	100 %

Fuente: Elaboración propia.

La Tabla 15 muestra que los tres métodos preservan más del $93\,\%$ de la diversidad informativa máxima posible, lo que indica que ninguno introduce una pérdida significativa de información al reducir las dimensiones. UMAP retiene la mayor diversidad informativa ($96{,}24\,\%$), seguido de PCA ($94{,}57\,\%$) y AF ($93{,}55\,\%$).

5.3.5. Resultados sin balanceo. En esta sección se presentan los resultados obtenidos al entrenar cada modelo utilizando la distribución original del conjunto de datos ODIR-5K, sin aplicar ninguna estrategia de peso de clases. Cada modelo fue evaluado sobre las tres representaciones reducidas, PCA, AF y UMAP, todas con 30 componentes, y los resultados se reportan mediante las métricas definidas en la sección 5.2.

5.3.5.1. Regresión Logística. La Regresión Logística, por su naturaleza lineal, traza hiperplanos de separación entre clases en el espacio reducido, lo que permite observar con claridad cómo cada representación organiza la información diagnóstica [322].

[322] Ibid.

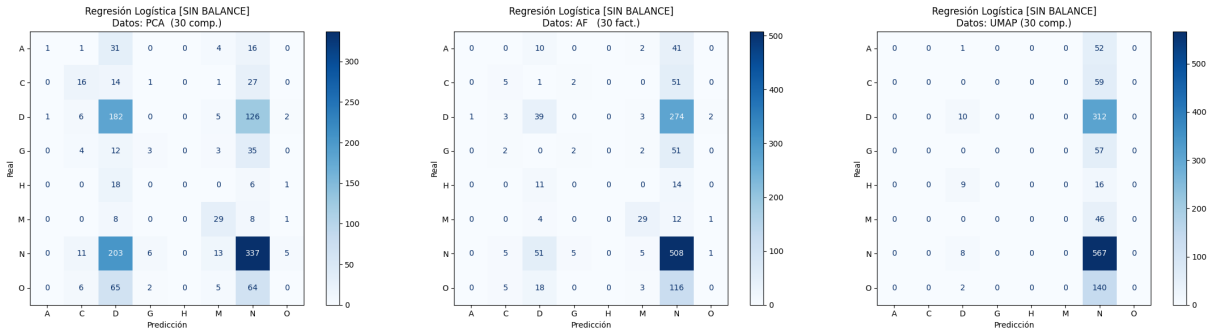
Cuadro 1. Reporte de clasificación — Regresión Logística sin balanceo

		PCA			AF			UMAP		
Clase	**Support**	**Precisión**	**Recall**	**F1-score**	**Precisión**	**Recall**	**F1-score**	**Precisión**	**Recall**	**F1-score**
A	53	0,50	0,02	0,04	0,00	0,00	0,00	0,00	0,00	0,00
C	59	0,36	0,27	0,31	0,25	0,08	0,13	0,00	0,00	0,00
D	322	0,34	0,57	0,43	0,29	0,12	0,17	0,33	0,03	0,06
G	57	0,25	0,05	0,09	0,22	0,04	0,06	0,00	0,00	0,00
H	25	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
M	46	0,48	0,63	0,55	0,66	0,63	0,64	0,00	0,00	0,00
N	575	0,54	0,59	0,56	0,48	0,88	0,62	0,45	0,99	0,62
O	142	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
macro avg	1279	0,31	0,27	0,25	0,24	0,22	0,20	0,10	0,13	0,08
Exactitud	–	0,4441			0,4558			0,4511		

Fuente: Elaboración propia.

El cuadro 1 revela diferencias importantes entre las tres representaciones. PCA genera el espacio reducido más informativo para este modelo: es la única representación donde el modelo logra distribuir sus predicciones entre varias categorías simultáneamente, incluyendo Diabetes con un F1-score de $0{,}43$ y Miopía patológica con $0{,}55$, lo que indica que PCA preserva características visuales que permiten separar estas clases de Normal. AF invierte este balance: concentra aún más las predicciones en Normal y mejora Miopía patológica hasta $0{,}64$, pero pierde casi por completo la capacidad de distinguir Diabetes, cuyo F1-score cae a $0{,}17$. Esto sugiere que AF organiza el espacio reducido de forma que Diabetes y Normal quedan demasiado cerca entre sí. Con UMAP, el modelo solo reconoce Normal de forma útil, lo que indica que esta representación no genera suficiente separación entre categorías para un clasificador lineal. En las tres representaciones, Hipertensión y Otras anomalías registran F1-score de $0{,}00$, clases que por su escasa representación y variabilidad visual resultan indetectables sin estrategia de balanceo.

Figura 11. Matrices de confusión — Regresión Logística sin balanceo.



PCA AF UMAP

Fuente: Elaboración propia .

La Figura 11 permite visualizar hacia dónde se dirigen los errores en cada caso. En PCA, los errores siguen dos direcciones principales: 126 casos de Diabetes son clasificados como Normal, mientras que Normal dispersa 203 de sus casos hacia Diabetes y otras categorías, lo que refleja que ambas clases comparten características visuales en este espacio; las clases minoritarias como Hipertensión y Otras anomalías terminan distribuidas entre Diabetes y Normal sin lograr predicciones propias. En AF, la atracción hacia Normal es más marcada: aunque la diagonal concentra 508 aciertos en esa clase, 274/322 casos de Diabetes son clasificados como Normal, lo que indica que que AF no genera suficiente separación entre estas dos categorías. En UMAP, prácticamente todas las clases son redirigidas hacia Normal, incluyendo casi en su totalidad algunas de las categorías, 52/53 imágenes de Degeneración macular, las 59 de Catarata, las 57 de Glaucoma, 16/25 de Hipertensión, las 46 de Miopía patológica y 140/142 de Otras anomalías.

La exactitud global de las tres representaciones, $0{,}4441$, $0{,}4558$ y $0{,}4511$, no refleja el desempeño real del modelo: se sostiene principalmente sobre los aciertos en Normal y Diabetes, ocultando el desempeño nulo en las clases minoritarias. El macro avg de F1-score, $0{,}25$, $0{,}20$ y $0{,}08$ respectivamente, es el indicador más informativo del desempeño real bajo condiciones de desbalanceo.

5.3.5.2. Máquinas de Soporte Vectorial (SVM-RBF). Las Máquinas de Soporte Vectorial buscan encontrar el hiperplano de máximo margen que separa las clases

en el espacio reducido, por lo que su desempeño depende en gran medida del kernel utilizado [323]. Durante la fase de exploración se evaluaron cuatro tipos de kernel (Lineal, RBF, Polinómico y Sigmoide) sobre cada representación, los resultados se pueden observar en la Tabla 16:

Tabla 16. Comparativa de exactitud entre kernels

Representación	**Kernel**	**Sin balance**	**Class Weight**
PCA	Lineal	0,2729	0,2854
PCA	RBF	0,4855	0,2752
PCA	Polinómico	0,4613	0,2877
PCA	Sigmoide	0,3573	0,0516
AF	Lineal	0,2721	0,2869
AF	RBF	0,4754	0,3534
AF	Polinómico	0,4394	0,3354
AF	Sigmoide	0,3229	0,0837
UMAP	Lineal	0,4511	0,1126
UMAP	RBF	0,4527	0,1274
UMAP	Polinómico	0,4550	0,1282
UMAP	Sigmoide	0,3057	0,0727

Fuente: Elaboración propia.

Finalmente se optó por el uso del kernel RBF ya que alcanza la exactitud más alta en PCA ($0{,}4855$) y AF ($0{,}4754$) bajo la condición sin balanceo, y mantiene resultados competitivos en UMAP ($0{,}4527$). El código completo de exploración está disponible en el repositorio del proyecto[324].

Una vez seleccionado el kernel RBF, se evaluó el modelo sobre las tres representaciones reducidas bajo la condición sin balanceo. Los resultados por clase se presentan en el Cuadro 2.

[323] Ibid.

[324] `https://bit.ly/notebook-ODIR5K`

Cuadro 2. Reporte de clasificación — SVM-RBF sin balanceo

		PCA			AF			UMAP		
Clase	**Support**	**Precisión**	**Recall**	**F1-score**	**Precisión**	**Recall**	**F1-score**	**Precisión**	**Recall**	**F1-score**
A	53	0,00	0,00	0,00	0,20	0,04	0,06	0,00	0,00	0,00
C	59	0,39	0,15	0,22	0,56	0,32	0,41	0,00	0,00	0,00
D	322	0,39	0,21	0,27	0,38	0,26	0,31	0,38	0,05	0,08
G	57	0,00	0,00	0,00	0,20	0,07	0,10	0,00	0,00	0,00
H	25	1,00	0,08	0,15	0,80	0,16	0,27	0,00	0,00	0,00
M	46	0,67	0,61	0,64	0,67	0,48	0,56	0,00	0,00	0,00
N	575	0,50	0,89	0,64	0,50	0,81	0,62	0,46	0,98	0,62
O	142	0,14	0,01	0,03	0,24	0,06	0,09	0,00	0,00	0,00
macro avg	1279	0,39	0,24	0,24	0,44	0,27	0,30	0,10	0,13	0,09
Exactitud	–	0,4855			0,4754			0,4527		

Fuente: Elaboración propia.

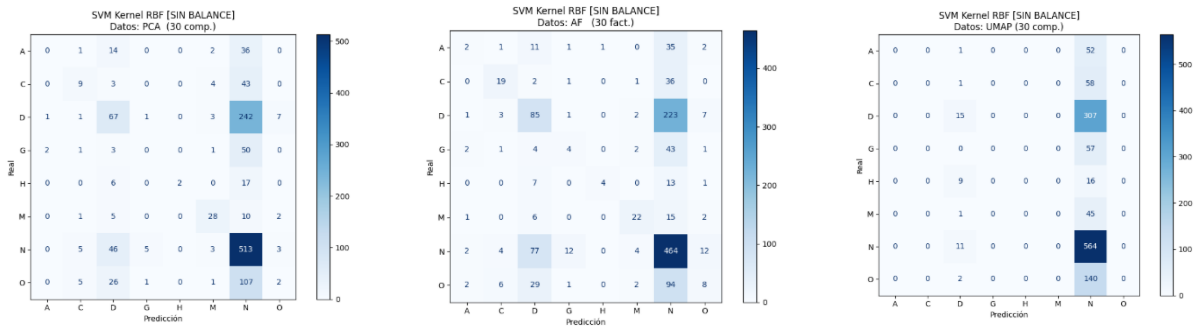
El Cuadro 2 permite comparar directamente el desempeño de las tres representaciones. PCA genera la mayor exactitud global ($0{,}4855$) y preserva información útil para Normal y Miopía patológica, ambas con F1-score de $0{,}64$. Un hallazgo particular es el de Hipertensión: precisión perfecta de $1{,}00$ pero recall de apenas $0{,}08$, lo que indica que el modelo solo se atreve a predecir esa clase cuando está muy seguro, aunque detecta muy pocos de sus casos reales. Sin embargo, Degeneración macular y Glaucoma registran F1-score de $0{,}00$, lo que sugiere que PCA no logra separar visualmente estas enfermedades de Normal en el espacio reducido.

AF presenta un perfil diferente: aunque su exactitud global es algo menor ($0{,}4754$), logra cobertura en todas las clases, incluyendo Degeneración macular y Glaucoma que con PCA quedaban sin detectar. El cambio más significativo es en Catarata, cuyo F1-score sube de $0{,}22$ a $0{,}41$, lo que indica que AF preserva mejor las características visuales que distinguen esta enfermedad. El macro avg de F1-score de $0{,}30$ frente a $0{,}24$ de PCA confirma que AF genera una representación más equitativa entre clases, aunque el desbalanceo sigue condicionando los resultados.

Con UMAP, el modelo reconoce únicamente Normal de forma útil: recall de $0{,}98$ pero con seis clases en $0{,}00$. Esto indica que la estructura no lineal de UMAP no genera

fronteras de decisión aprovechables para un clasificador de margen máximo bajo condiciones de desbalanceo, reproduciendo el mismo comportamiento observado con la Regresión Logística sobre esta representación.

Figura 12. Matrices de confusión — SVM-RBF sin balanceo.



PCA AF UMAP

Fuente: Elaboración propia .

La Figura 12 permite identificar hacia dónde se dirigen los errores en cada representación. En PCA, el destino predominante es Normal: 242/322 casos de Diabetes, 36/53 de Degeneración macular y 50/57 de Glaucoma son clasificados en esa clase, lo que sugiere que estas enfermedades comparten características visuales con los ojos sanos en el espacio reducido por PCA. En AF, la misma tendencia se mantiene pero con menor intensidad: Diabetes dirige 223 de sus casos hacia Normal, frente a los 242 de PCA, y la distribución fuera de la diagonal es más variada, con Degeneración macular y Otras anomalías registrando al menos algunos aciertos propios. En UMAP, prácticamente todas las clases son redirigidas hacia Normal: 52/53 imágenes de Degeneración macular, las 58/59 de Catarata, las 57 de Glaucoma, las 16/25 de Hipertensión, las 45/46 de Miopía patológica y 140/142 de Otras anomalías quedan sin ninguna predicción correcta.

La exactitud global de las tres representaciones, $0{,}4855$, $0{,}4754$ y $0{,}4527$, no refleja el desempeño real del modelo: se sostiene principalmente sobre los aciertos en Normal, clase que por sí sola concentra el $44{,}9\,\%$ del conjunto de prueba. El macro avg de F1-score, $0{,}24$, $0{,}30$ y $0{,}09$ respectivamente, ofrece una visión más informativa porque promedia el desempeño de forma equitativa entre todas las clases, independientemente de su tamaño. Bajo este criterio, AF es la representación más equitativa para el SVM-RBF en esta condición, pues aunque su exactitud global es menor que la de

PCA, su macro avg de $0{,}30$ refleja que el modelo logra distribuir sus predicciones de forma más amplia entre las ocho categorías diagnósticas.

5.3.5.3. Red Neuronal Convolucional (CNN). A diferencia de los modelos anteriores, la CNN incluye además curvas de pérdida y exactitud que permiten analizar el comportamiento del aprendizaje durante el entrenamiento [325]. Los resultados por clase se presentan en el Cuadro 3.

Cuadro 3. Reporte de clasificación — CNN sin balanceo

		PCA			AF			UMAP		
Clase	**Support**	**Precisión**	**Recall**	**F1**	**Precisión**	**Recall**	**F1**	**Precisión**	**Recall**	**F1**
A	53	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
C	59	0,57	0,20	0,30	0,59	0,27	0,37	0,60	0,05	0,09
D	322	0,42	0,18	0,25	0,35	0,15	0,21	0,45	0,11	0,17
G	57	1,00	0,02	0,03	0,00	0,00	0,00	0,00	0,00	0,00
H	25	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
M	46	0,85	0,48	0,61	0,64	0,54	0,59	0,50	0,02	0,04
N	575	0,48	0,91	0,63	0,48	0,90	0,63	0,46	0,96	0,62
O	142	0,50	0,01	0,01	0,00	0,00	0,00	0,00	0,00	0,00
macro avg	1279	0,48	0,22	0,23	0,26	0,23	0,22	0,25	0,14	0,12
Exactitud	–	0,4832			0,4738			0,4605		

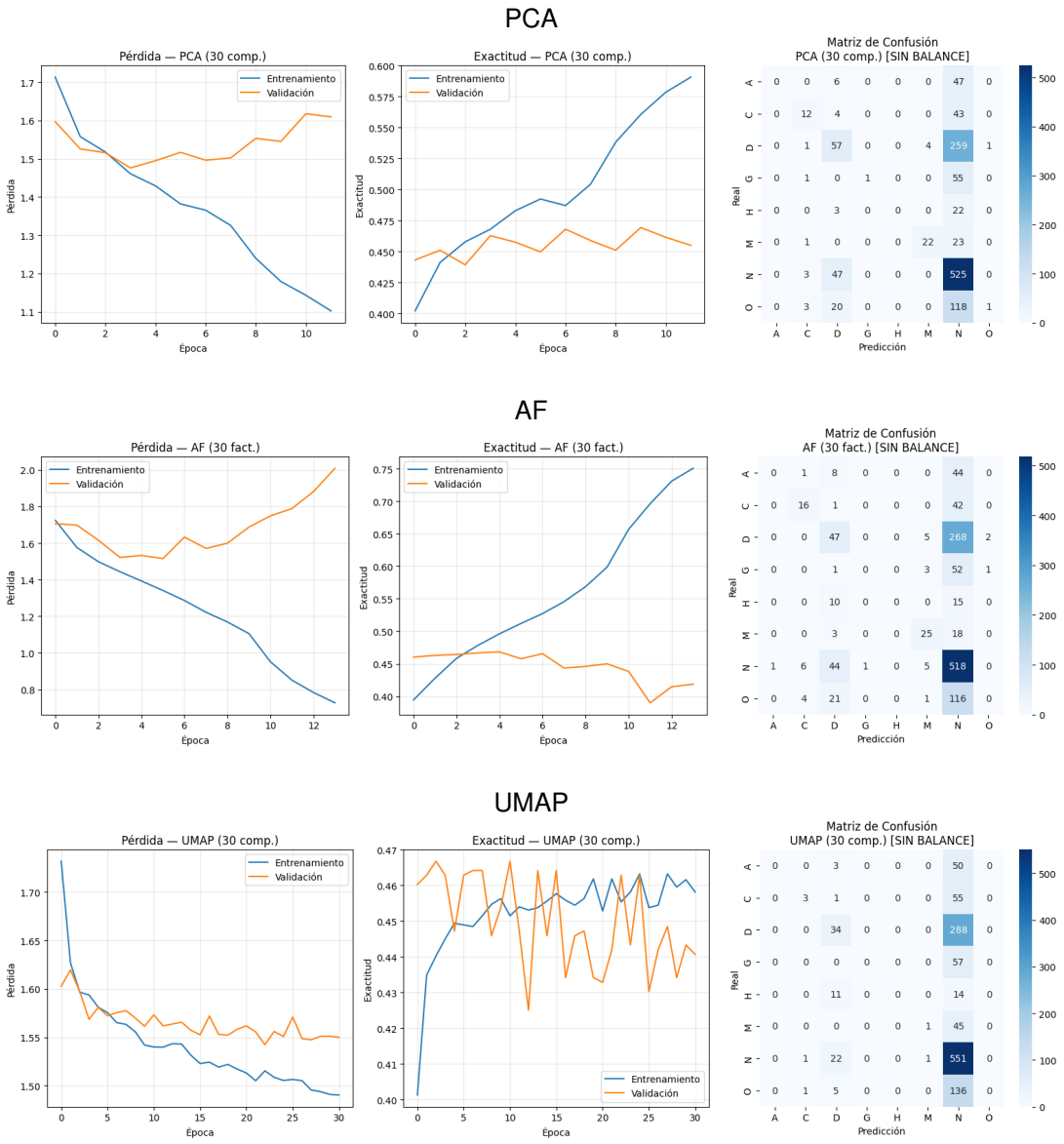
Fuente: Elaboración propia.

El Cuadro 3 revela que las tres representaciones comparten un patrón común: Normal lidera en F1-score en los tres casos con valores de $0{,}63$, $0{,}63$ y $0{,}62$, y Miopía patológica es consistentemente la clase minoritaria mejor identificada. Sin embargo, hay diferencias importantes entre representaciones. PCA genera el mejor desempeño general: Miopía patológica alcanza su mayor F1-score $(0{,}61)$ con una precisión alta de $0{,}85$, y Glaucoma obtiene una predicción correcta con precisión perfecta de $1{,}00$, aunque su recall de $0{,}02$ indica que el modelo solo se atreve a predecir esa clase cuando está muy seguro. Con AF, Glaucoma pierde esa única predicción y cae a F1-score de $0{,}00$, y Miopía patológica baja de $0{,}61$ a $0{,}59$, lo que sugiere que AF no preserva de

[325] LeCUN, Yann et al. Op. cit., 1998.

forma tan nítida las características que distinguen estas clases. Con UMAP, el deterioro es más marcado: Miopía patológica cae a $0{,}04$ a pesar de una precisión de $0{,}50$, y solo Normal y Diabetes obtienen aciertos relevantes. En las tres representaciones, Degeneración macular e Hipertensión registran F1-score de $0{,}00$, lo que indica que estas clases resultan indetectables para la CNN sin estrategia de balanceo.

Figura 13. Curvas de pérdida, exactitud y matriz de confusión — CNN sin balanceo.



Fuente: Elaboración propia.

Las Figuras en 13 permiten analizar dos aspectos simultáneamente: el comportamiento del aprendizaje a través de las curvas, y los patrones de error a través de las matrices de confusión. En cuanto al aprendizaje, PCA muestra el menor sobreajuste de las tres: la curva de validación se estabiliza alrededor de $0{,}45$ sin seguir a la de entrena-

miento, pero sin alejarse drásticamente. AF presenta una brecha mayor entre ambas curvas, con la de entrenamiento alcanzando valores cercanos a $0{,}75$ mientras la de validación decrece hacia el final, señal de sobreajuste más pronunciado. UMAP genera el aprendizaje más inestable: ambas curvas oscilan frecuentemente sin mostrar una tendencia clara, lo que sugiere que la estructura no lineal de UMAP no es compatible con la forma en que la CNN procesa vectores de baja dimensión.

En cuanto a los errores, el destino predominante en las tres representaciones es Normal. En PCA, Diabetes dirige 259/322 casos hacia esa clase y Degeneración macular 47/53. En AF, Diabetes aumenta a 268 casos hacia Normal, lo que sugiere que que AF no genera mejor separación para esta clase. En UMAP, el comportamiento es más extremo: Diabetes dirige 288/322 casos hacia Normal, Degeneración macular 50/53, Hipertensión 14/25 y Glaucoma la totalidad de sus casos.

La exactitud global de $0{,}4832$, $0{,}4738$ y $0{,}4605$ no refleja el desempeño real del modelo: se sostiene principalmente sobre los aciertos en Normal. El macro avg de F1-score, $0{,}23$, $0{,}22$ y $0{,}12$ respectivamente, muestra que PCA es la representación más equitativa para la CNN bajo condiciones de desbalanceo, aunque las diferencias entre PCA y AF son mínimas. Frente a la Regresión Logística y el SVM-RBF, la CNN no muestra mejoras significativas, lo que sugiere que su mayor complejidad arquitectónica no se traduce en ventajas reales cuando opera sobre vectores reducidos sin estrategia de balanceo.

5.3.5.4. Transformador de Visión (ViT). Como la CNN, el ViT incluye curvas de pérdida y exactitud que permiten analizar el comportamiento del aprendizaje durante el entrenamiento [326]. Los resultados por clase se presentan en el Cuadro 4.

[326] VASWANI. Op. cit.

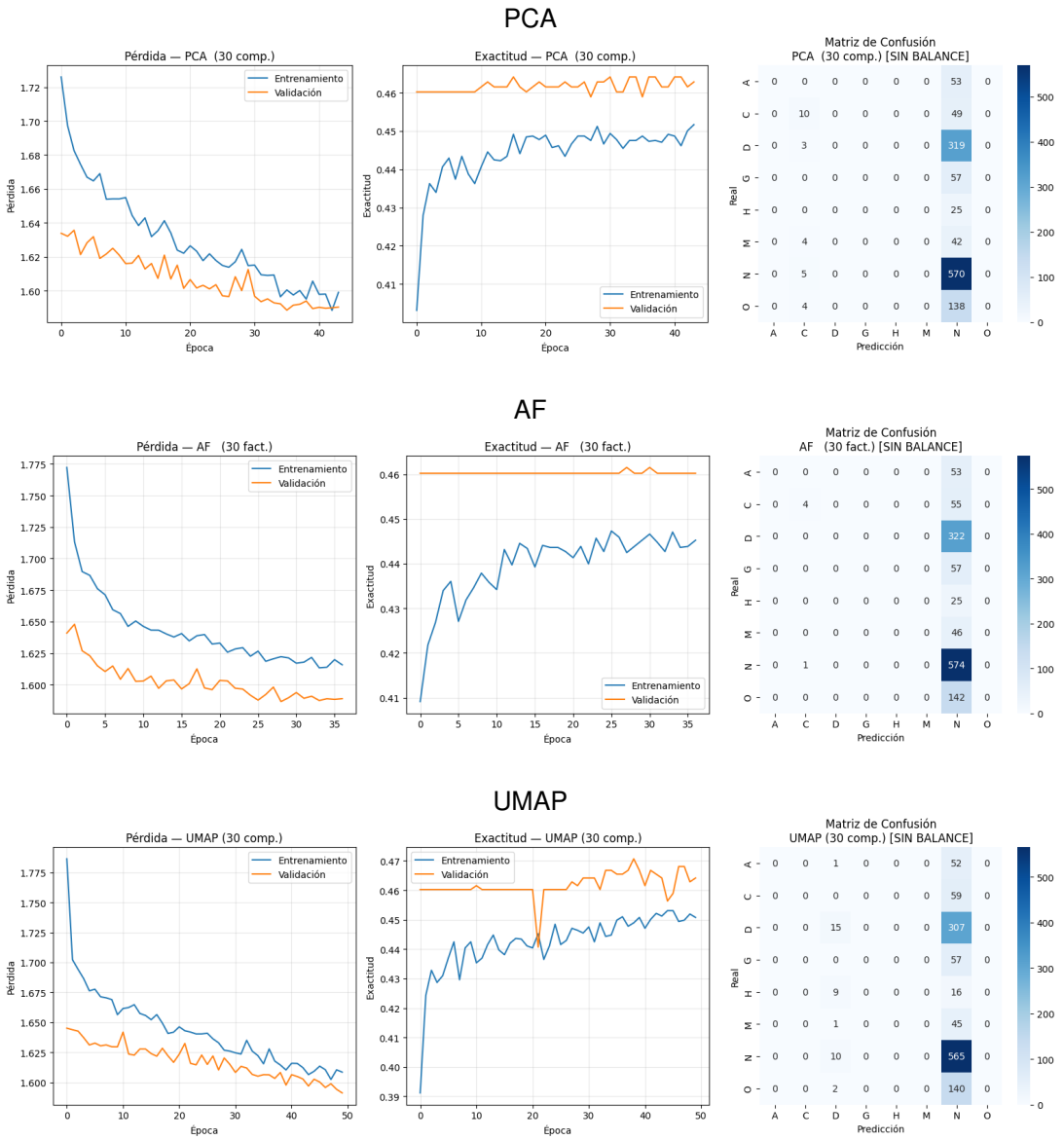
Cuadro 4. Reporte de clasificación — ViT sin balanceo

		PCA			AF			UMAP		
Clase	**Support**	**Precisión**	**Recall**	**F1**	**Precisión**	**Recall**	**F1**	**Precisión**	**Recall**	**F1**
A	53	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
C	59	0,38	0,17	0,24	0,00	0,07	0,12	0,00	0,00	0,00
D	322	0,00	0,00	0,00	0,00	0,00	0,00	0,39	0,05	0,08
G	57	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
H	25	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
M	46	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
N	575	0,45	0,99	0,62	0,45	1,00	0,62	0,46	0,98	0,62
O	142	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
macro avg	1279	0,10	0,15	0,11	0,16	0,13	0,09	0,11	0,13	0,09
Exactitud	–	0,4535			0,4519			0,4535		

Fuente: Elaboración propia.

El Cuadro 4 revela el patrón más extremo observado en toda la sección sin balanceo: el ViT concentra prácticamente todas sus predicciones en Normal en las tres representaciones, con macro avg de F1-score que no superan $0{,}11$ en ningún caso. En PCA, únicamente Catarata escapa parcialmente con un F1-score de $0{,}24$, mientras que seis clases registran $0{,}00$. Con AF, el deterioro es mayor: Catarata baja a $0{,}12$ y Diabetes, que al menos obtenía algunos aciertos con otros modelos sobre esta representación, cae también a $0{,}00$. Con UMAP, Diabetes recupera 15 aciertos con un F1-score de $0{,}08$, siendo el único cambio positivo entre las tres representaciones. A diferencia de la Regresión Logística y el SVM-RBF, que sobre PCA y AF lograban F1-scores superiores a $0{,}55$ en Miopía patológica, el ViT no supera $0{,}24$ en ninguna clase fuera de Normal, lo que sugiere que el problema no está en la representación sino en la incompatibilidad entre la arquitectura del ViT y los vectores de baja dimensión: la reducción dimensional elimina precisamente la estructura espacial que el mecanismo de atención necesita para funcionar correctamente.

Figura 14. Curvas de pérdida, exactitud y matriz de confusión — ViT sin balanceo.



Fuente: Elaboración propia .

Las Figuras en 14 muestran un rasgo distintivo respecto a la CNN: el ViT no presenta sobreajuste pronunciado en ninguna de las tres representaciones, pues las curvas de entrenamiento y validación convergen hacia valores similares alrededor de $0{,}45$ en los tres casos. Sin embargo, esta convergencia no implica buen desempeño, sino que refleja que el modelo alcanzó un límite en su capacidad de aprendizaje sobre vectores reducidos, independientemente de la representación utilizada. En cuanto a los errores, las matrices de confusión confirman el patrón observado en la tabla: en PCA, Diabetes dirige 319/322 casos hacia Normal, Degeneración macular, Glaucoma e Hipertensión la totalidad de sus casos, y Miopía patológica 42/46. En AF, la concentración es aún más marcada: Diabetes, Glaucoma, Hipertensión, Miopía patológica y

Otras anomalías dirigen la totalidad de sus casos hacia Normal. En UMAP, el patrón es prácticamente idéntico al de PCA, con Diabetes como única clase que obtiene algunos aciertos propios.

La exactitud global de $0{,}4535$, $0{,}4519$ y $0{,}4535$ no refleja el desempeño real del modelo. El macro avg de F1-score, $0{,}11$, $0{,}09$ y $0{,}09$ respectivamente, es el más bajo registrado entre todos los modelos evaluados en esta sección, esto confirma que el ViT es el modelo que peor aprovecha las representaciones reducidas bajo condiciones de desbalanceo.

5.3.6. Resultados con balance. En esta sección se presentan los resultados obtenidos al aplicar pesos de clase durante el entrenamiento sobre las mismas tres representaciones reducidas. Esta estrategia produce una caída en la exactitud global respecto a los resultados sin balanceo, consecuencia esperada dado que el modelo distribuye su capacidad de aprendizaje entre todas las clases en lugar de concentrarse en las más frecuentes. En un contexto clínico este resultado es favorable, pues identificar correctamente una enfermedad minoritaria puede ser tan importante como clasificar bien los casos más comunes [327].

5.3.6.1. Regresión Logística. La Regresión Logística, por su naturaleza lineal, permite observar con claridad cómo el balanceo redistribuye las predicciones entre clases [328]. Los resultados por clase se presentan en el Cuadro 5.

[327] JOHNSON, Justin M. Op. cit.
[328] HASTIE. Op. cit.

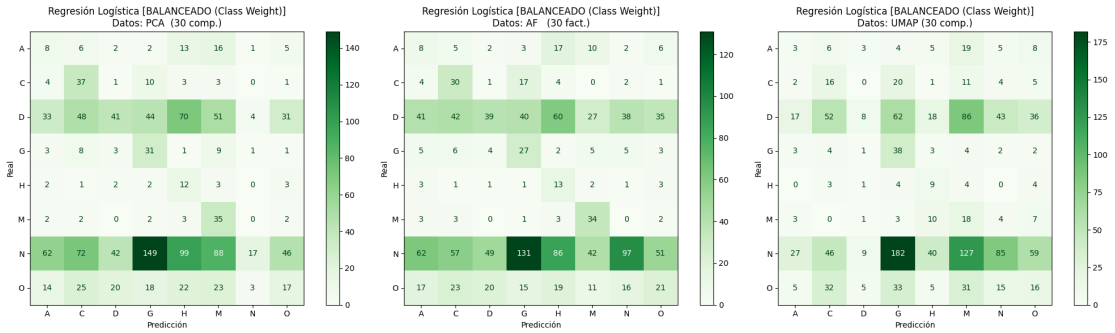
Cuadro 5. Reporte de clasificación — Regresión Logística con balanceo

		PCA			AF			UMAP		
Clase	**Support**	**Precisión**	**Recall**	**F1-score**	**Precisión**	**Recall**	**F1-score**	**Precisión**	**Recall**	**F1-score**
A	53	0,06	0,15	0,09	0,06	0,15	0,08	0,05	0,06	0,05
C	59	0,19	0,63	0,29	0,18	0,51	0,27	0,10	0,27	0,15
D	322	0,37	0,13	0,19	0,34	0,12	0,18	0,29	0,02	0,05
G	57	0,12	0,54	0,20	0,11	0,47	0,18	0,11	0,67	0,19
H	25	0,05	0,48	0,10	0,06	0,52	0,11	0,10	0,36	0,16
M	46	0,15	0,76	0,26	0,26	0,74	0,38	0,06	0,39	0,10
N	575	0,65	0,03	0,06	0,60	0,17	0,26	0,54	0,15	0,23
O	142	0,16	0,12	0,14	0,17	0,15	0,16	0,12	0,11	0,11
macro avg	1279	0,22	0,35	0,16	0,22	0,35	0,20	0,17	0,25	0,13
Exactitud	–	0,1548			0,2103			0,1509		

Fuente: Elaboración propia.

El Cuadro 5 confirma el efecto esperado del balanceo: todas las clases obtienen ahora predicciones correctas en las tres representaciones, incluyendo Hipertensión y Otras anomalías que sin balanceo registraban F1-score de $0{,}00$. Las diferencias entre representaciones son relevantes. AF genera el mejor desempeño general con una exactitud de $0{,}2103$ y un macro avg de F1-score de $0{,}20$: Miopía patológica alcanza su mejor resultado ($0{,}38$) y Normal recupera un recall de $0{,}17$, logrando el equilibrio más adecuado entre clases mayoritarias y minoritarias. PCA muestra recalls altos en varias clases minoritarias, incluyendo Miopía patológica ($0{,}76$) y Catarata ($0{,}63$), pero a un costo importante: Normal cae a un recall de apenas $0{,}03$, lo que significa que el modelo casi deja de reconocer la clase más frecuente. Con UMAP, el balanceo desplaza las predicciones principalmente hacia Glaucoma, que obtiene el recall más alto entre las tres representaciones ($0{,}67$), aunque con una precisión de apenas $0{,}11$ que indica que la mayoría de esas predicciones son incorrectas. Diabetes cae a un recall de $0{,}02$ en UMAP, lo que indica que esta representación no genera información suficiente para distinguir esa clase bajo ninguna condición.

Figura 15. Matrices de confusión — Regresión Logística con balanceo.



PCA AF UMAP

Fuente: Elaboración propia .

La Figura 15 permite visualizar cómo se redistribuyen los errores respecto a la condición sin balanceo. En PCA, Normal dispersa sus 575 casos principalmente hacia Glaucoma (149) e Hipertensión (99), con apenas 17 aciertos propios, mientras que Diabetes distribuye sus casos mayormente entre Miopía (51), Hipertensión (70) y Diabetes correcta (41). En AF, la distribución es más variada: Normal obtiene 97 aciertos propios y dispersa el resto principalmente hacia Glaucoma (131) e Hipertensión (86), y Diabetes obtiene 39 aciertos propios con sus casos repartidos principalmente entre Hipertensión (60) y Catarata (42). En UMAP, Glaucoma absorbe la mayor parte de las predicciones fuera de la diagonal: 182 casos de Normal y 62 de Diabetes son clasificados en esa categoría, lo que explica su alto recall pero baja precisión.

La exactitud global de $0{,}1548$, $0{,}2103$ y $0{,}1509$ no refleja el desempeño real del modelo bajo esta condición. El macro avg de F1-score, $0{,}16$, $0{,}20$ y $0{,}13$ respectivamente, muestra que AF es la representación más equitativa para la Regresión Logística con pesos de clase, confirmando la tendencia observada sin balanceo donde AF también lideraba en este indicador.

5.3.6.2. Máquinas de Soporte Vectorial (SVM-RBF). Los resultados del SVM-RBF con pesos de clase se presentan en el Cuadro 6.

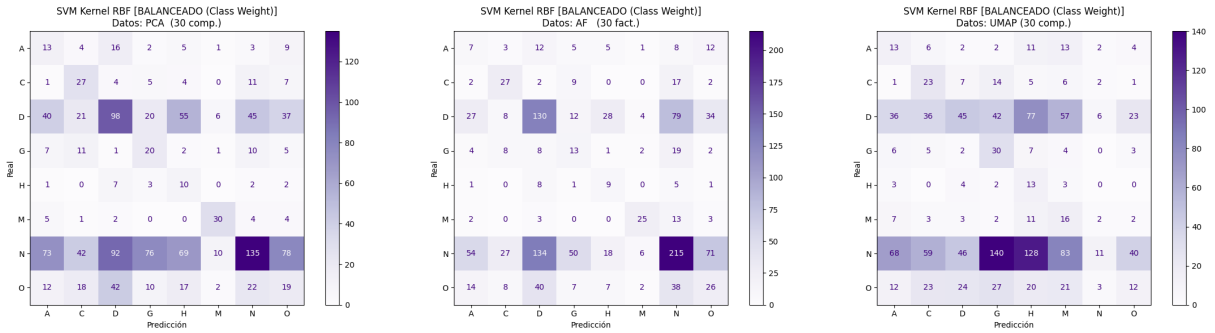
Cuadro 6. Reporte de clasificación — SVM-RBF con balanceo

		PCA			AF			UMAP		
Clase	**Support**	**Precisión**	**Recall**	**F1-score**	**Precisión**	**Recall**	**F1-score**	**Precisión**	**Recall**	**F1-score**
A	53	0,09	0,25	0,13	0,06	0,13	0,09	0,09	0,25	0,13
C	59	0,22	0,46	0,30	0,33	0,46	0,39	0,15	0,39	0,21
D	322	0,37	0,30	0,34	0,39	0,40	0,39	0,34	0,14	0,20
G	57	0,15	0,35	0,21	0,13	0,23	0,17	0,12	0,53	0,19
H	25	0,06	0,40	0,11	0,13	0,36	0,19	0,05	0,52	0,09
M	46	0,60	0,65	0,62	0,62	0,54	0,58	0,08	0,35	0,13
N	575	0,58	0,23	0,33	0,55	0,37	0,44	0,42	0,02	0,04
O	142	0,12	0,13	0,13	0,17	0,18	0,18	0,14	0,08	0,11
macro avg	1279	0,27	0,35	0,27	0,30	0,34	0,30	0,17	0,28	0,14
Exactitud	–	0,2752			0,3534			0,1274		

Fuente: Elaboración propia.

El Cuadro 6 muestra que el balanceo produce resultados muy distintos según la representación. AF genera el mejor desempeño general con una exactitud de $0{,}3534$ y un macro avg de F1-score de $0{,}30$: todas las clases obtienen predicciones correctas y el modelo logra un equilibrio notable, con Miopía patológica liderando con F1-score de $0{,}58$, Diabetes con $0{,}39$ y Normal recuperando un recall de $0{,}37$ con 215/575 posibles. PCA ocupa el segundo lugar con exactitud de $0{,}2752$ y macro avg de $0{,}27$: Miopía patológica obtiene su mejor resultado entre las tres representaciones con F1-score de $0{,}62$ y recall de $0{,}65$, y Degeneración macular y Glaucoma, que sin balanceo registraban $0{,}00$, alcanzan recalls de $0{,}25$ y $0{,}35$ respectivamente. Con UMAP, el balanceo desplaza las predicciones principalmente hacia Glaucoma e Hipertensión, con recalls de $0{,}53$ y $0{,}52$ pero precisiones muy bajas, mientras que Normal cae a un recall de apenas $0{,}02$ y Miopía patológica baja drásticamente de $0{,}62$ en PCA a $0{,}13$, esto confirma que UMAP es la representación menos adecuada para el SVM-RBF bajo condiciones de balanceo.

Figura 16. Matrices de confusión — SVM-RBF con balanceo.



PCA AF UMAP

Fuente: Elaboración propia .

La Figura 16 permite visualizar los patrones de error bajo esta condición. En PCA, la diagonal presenta valores en todas las clases, con Normal obteniendo 135/575 aciertos y Diabetes 98/322. Fuera de la diagonal, Normal dispersa sus casos principalmente entre Diabetes (92) y Glaucoma (76), mientras que Diabetes reparte los suyos entre Glaucoma (20), Hipertensión (55) y Normal (45). En AF, la distribución es aún más amplia: Normal obtiene 215 aciertos y Diabetes 130, con sus casos dispersos principalmente entre Normal (79) y Otras anomalías (34). En UMAP, Glaucoma absorbe la mayor parte de los errores fuera de la diagonal: 140 casos de Normal y 42 de Diabetes son clasificados en esa categoría, y Normal obtiene apenas 11 aciertos de sus 575 casos.

La exactitud global de $0{,}2752$, $0{,}3534$ y $0{,}1274$ no refleja el desempeño real del modelo. El macro avg de F1-score, $0{,}27$, $0{,}30$ y $0{,}14$ respectivamente, muestra que AF es la representación más equitativa para el SVM-RBF con pesos de clase, resultado consistente con lo observado sin balanceo donde AF también lideraba en este indicador. Comparado con la Regresión Logística bajo la misma condición, el SVM-RBF obtiene un mejor desempeño en las tres representaciones, con macro avg de F1-score superiores en PCA ($0{,}27$ frente a $0{,}16$), AF ($0{,}30$ frente a $0{,}20$) y UMAP ($0{,}14$ frente a $0{,}13$).

5.3.6.3. Red Neuronal Convolucional (CNN). Los resultados de la CNN con pesos de clase se presentan en el Cuadro 7.

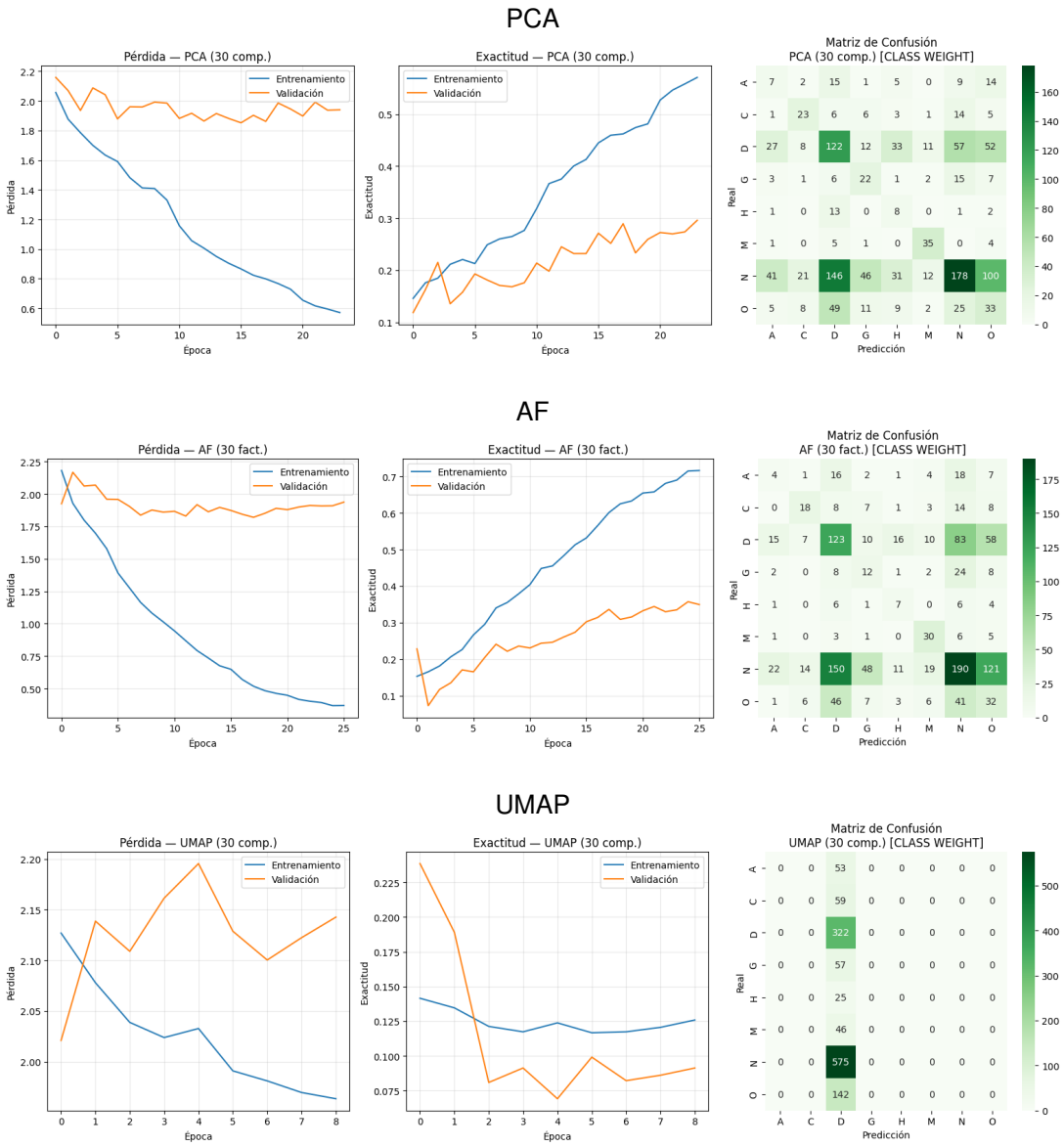
Cuadro 7. Reporte de clasificación — CNN con balanceo

		PCA			AF			UMAP		
Clase	**Support**	**Precisión**	**Recall**	**F1-score**	**Precisión**	**Recall**	**F1-score**	**Precisión**	**Recall**	**F1-score**
A	53	0,08	0,13	0,10	0,09	0,08	0,08	0,00	0,00	0,00
C	59	0,37	0,39	0,38	0,39	0,31	0,34	0,00	0,00	0,00
D	322	0,34	0,38	0,36	0,34	0,38	0,36	0,25	1,00	0,40
G	57	0,22	0,39	0,28	0,14	0,21	0,17	0,00	0,00	0,00
H	25	0,09	0,32	0,14	0,17	0,28	0,22	0,00	0,00	0,00
M	46	0,56	0,76	0,64	0,41	0,65	0,50	0,00	0,00	0,00
N	575	0,60	0,31	0,41	0,50	0,33	0,42	0,00	0,00	0,00
O	142	0,15	0,23	0,18	0,13	0,23	0,17	0,00	0,00	0,00
macro avg	1279	0,30	0,36	0,31	0,27	0,31	0,28	0,03	0,12	0,05
Exactitud	–	0,3346			0,3253			0,2518		

Fuente: Elaboración propia.

El Cuadro 7 revela diferencias importantes entre las tres representaciones bajo esta condición. PCA genera el mejor desempeño general con exactitud de $0{,}3346$ y macro avg de F1-score de $0{,}31$: todas las clases obtienen predicciones correctas, incluyendo Degeneración macular e Hipertensión que sin balanceo registraban $0{,}00$, y Miopía patológica lidera con F1-score de $0{,}64$ y recall de $0{,}76$. AF muestra un perfil similar al de PCA pero con desempeño algo inferior en la mayoría de las clases: Miopía patológica baja de $0{,}64$ a $0{,}50$ en F1-score y Degeneración macular de $0{,}10$ a $0{,}08$, aunque Hipertensión mejora ligeramente de $0{,}14$ a $0{,}22$. UMAP presenta el caso más extremo de todo el análisis: el modelo clasifica la totalidad de las imágenes como Diabetes, obteniendo un recall de $1{,}00$ para esa clase pero F1-score de $0{,}00$ en las siete categorías restantes. Este comportamiento es el inverso al observado sin balanceo, donde el modelo colapsaba hacia Normal, lo que indica que la representación UMAP no genera suficiente información discriminativa para la CNN bajo ninguna de las dos condiciones.

Figura 17. Curvas de pérdida, exactitud y matriz de confusión — CNN con balanceo.



Fuente: Elaboración propia .

Las Figuras en 17 permiten analizar el aprendizaje y los patrones de error simultáneamente. En PCA, las curvas muestran sobreajuste moderado: la exactitud de entrenamiento alcanza $0{,}50$ mientras la de validación se estabiliza alrededor de $0{,}30$, y la matriz de confusión muestra una diagonal activa con Normal obteniendo 178/575, Diabetes 122/322 y Miopía patológica 35/46. Fuera de la diagonal, Normal dispersa sus casos principalmente hacia Diabetes (146) y Otras anomalías (100). En AF, el sobreajuste es más pronunciado: la exactitud de entrenamiento sube hasta $0{,}70$ mientras la de validación se estabiliza alrededor de $0{,}33$, y la distribución de errores es similar a PCA aunque con Normal obteniendo 190 aciertos y Diabetes 123. En UMAP, ambas curvas oscilan frecuentemente sin mostrar tendencia clara, estabilizándose alrededor

de $0{,}12$, y la matriz de confusión confirma el colapso hacia Diabetes: todas las imágenes de las siete clases restantes son clasificadas en esa categoría.

La exactitud global de $0{,}3346$, $0{,}3253$ y $0{,}2518$ no refleja el desempeño real del modelo. El macro avg de F1-score, $0{,}31$, $0{,}28$ y $0{,}05$ respectivamente, muestra que PCA es la representación más equitativa para la CNN con pesos de clase. Comparada con la Regresión Logística y el SVM-RBF bajo la misma condición, la CNN obtiene el mejor macro avg en PCA ($0{,}31$ frente a $0{,}16$ y $0{,}27$) y en AF ($0{,}28$ frente a $0{,}20$ y $0{,}30$), aunque en UMAP su sesgo hacia Diabetes la convierte en la peor opción entre los tres modelos.

5.3.6.4. Transformador de Visión (ViT). Los resultados del ViT con pesos de clase se presentan en el Cuadro 8.

Cuadro 8. Reporte de clasificación — ViT con balanceo

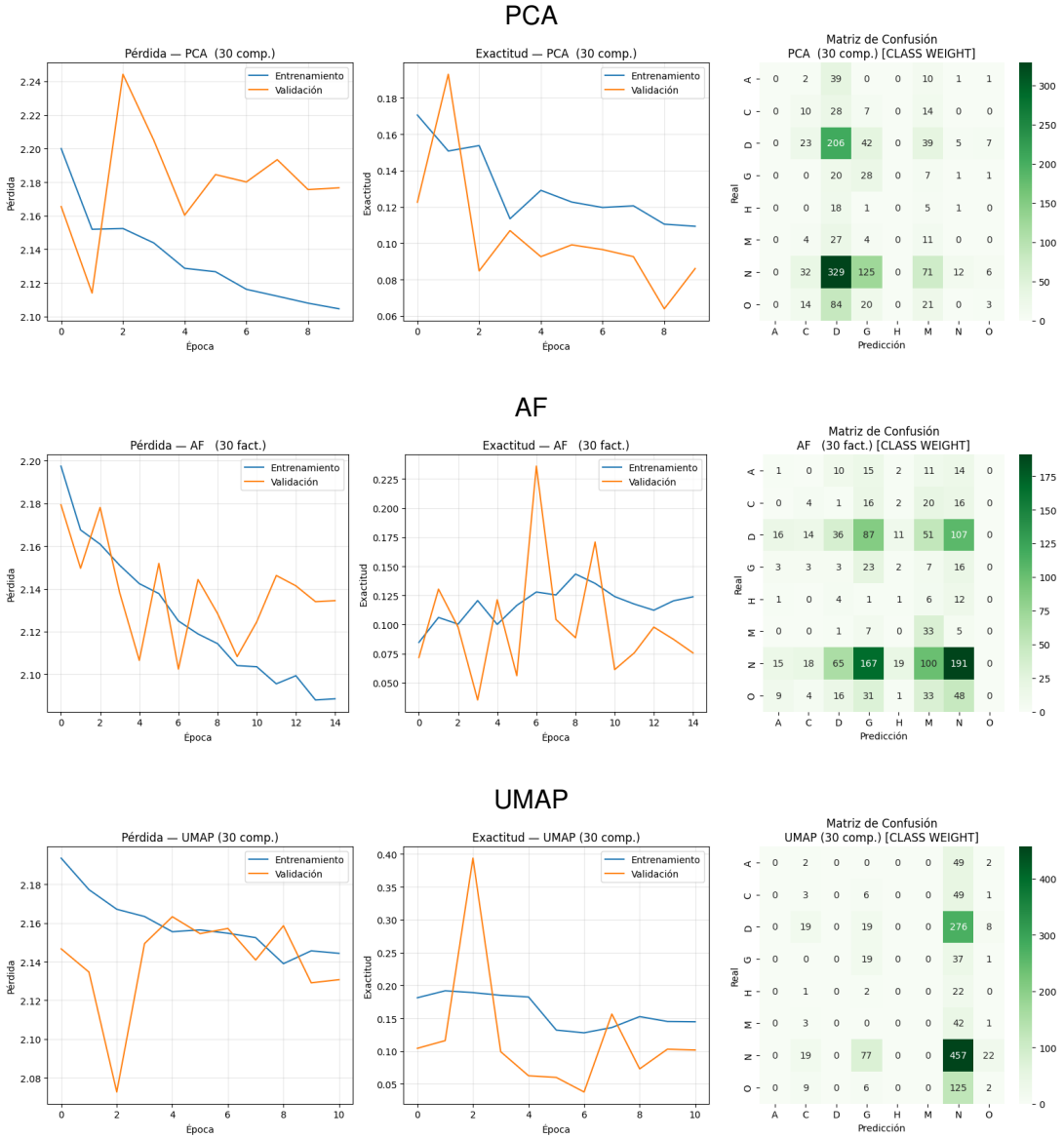
		PCA			AF			UMAP		
Clase	**Support**	**Precisión**	**Recall**	**F1**	**Precisión**	**Recall**	**F1**	**Precisión**	**Recall**	**F1**
A	53	0,00	0,00	0,00	0,02	0,02	0,02	0,00	0,00	0,00
C	59	0,12	0,17	0,14	0,09	0,07	0,08	0,05	0,05	0,05
D	322	0,27	0,64	0,38	0,26	0,11	0,16	0,00	0,00	0,00
G	57	0,12	0,49	0,20	0,07	0,40	0,11	0,15	0,33	0,20
H	25	0,00	0,00	0,00	0,03	0,04	0,03	0,00	0,00	0,00
M	46	0,06	0,24	0,10	0,13	0,72	0,21	0,00	0,00	0,00
N	575	0,60	0,02	0,04	0,47	0,33	0,39	0,43	0,79	0,56
O	142	0,17	0,02	0,04	0,00	0,00	0,00	0,05	0,01	0,02
macro avg	1279	0,17	0,20	0,11	0,13	0,21	0,13	0,09	0,15	0,10
Exactitud	–	0,2111			0,2260			0,3761		

Fuente: Elaboración propia.

El Cuadro 8 revela el comportamiento más inconsistente observado entre todos los modelos evaluados: cada representación concentra sus predicciones en una clase distinta. Con PCA, el balanceo desplaza las predicciones hacia Diabetes, que obtiene un recall de $0{,}64$ con 206/322 aciertos, mientras que Normal cae a un recall de

apenas $0{,}02$ y Degeneración macular e Hipertensión registran F1-score de $0{,}00$. Con AF, Miopía patológica obtiene el recall más alto con $0{,}72$, identificando 33/46 casos, y Normal recupera un recall de $0{,}33$ con 191 aciertos, aunque Diabetes cae a apenas $0{,}11$ y Otras anomalías registra F1-score de $0{,}00$. Con UMAP, el modelo vuelve a concentrarse en Normal con recall de $0{,}79$ y 457 aciertos de 575, mientras que Diabetes, Degeneración macular, Hipertensión y Miopía patológica registran F1-score de $0{,}00$. En ninguna de las tres representaciones el balanceo logra distribuir el aprendizaje de forma equitativa entre todas las clases.

Figura 18. Curvas de pérdida, exactitud y matriz de confusión — ViT con balanceo.



Fuente: Elaboración propia .

Las Figuras en 18 muestran un rasgo común en las tres representaciones: las cur-

vas de validación presentan oscilaciones frecuentes sin una tendencia clara, lo que indica que el ViT no logra estabilizar su aprendizaje bajo condiciones de balanceo independientemente de la representación utilizada. En PCA, la curva de exactitud de validación oscila sin superar $0{,}20$, y la matriz de confusión muestra que Normal dirige 329/575 casos hacia Diabetes y apenas conserva 12 aciertos propios. En AF, el sobreajuste es más pronunciado con la curva de entrenamiento estabilizándose alrededor de $0{,}13$ mientras la validación oscila entre $0{,}05$ y $0{,}22$, y la matriz muestra que Diabetes dirige 87 de sus casos hacia Glaucoma y 107 hacia Normal. En UMAP, las oscilaciones de validación son las más amplias de las tres, con picos que alcanzan $0{,}40$, y la matriz confirma que Normal absorbe 276 casos de Diabetes, 49 de Degeneración macular y 42 de Miopía patológica.

La exactitud global de $0{,}2111$, $0{,}2260$ y $0{,}3761$ no refleja el desempeño real del modelo. El macro avg de F1-score, $0{,}11$, $0{,}13$ y $0{,}10$ respectivamente, es el más bajo registrado entre todos los modelos evaluados bajo la condición de pesos de clase, confirmando que el ViT es el modelo que peor responde a la estrategia de balanceo sobre vectores reducidos. La exactitud más alta corresponde a UMAP, pero se explica principalmente por los 457 aciertos en Normal, lo que indica que el balanceo no cumple su objetivo en ninguna de las tres representaciones para este modelo.

5.3.7. Entropía global de las predicciones. La entropía global de las predicciones mide qué tan distribuidas quedaron las predicciones de cada modelo entre las ocho categorías diagnósticas en el conjunto de prueba. Una entropía alta indica una distribución más equilibrada entre clases, mientras que una entropía baja refleja concentración en pocas categorías. La entropía máxima posible es $H_{max} = \log(8) = 2{,}0794$ nats, valor que representa la distribución más uniforme posible y sirve como referencia comparativa [329]. Los resultados bajo ambas condiciones se presentan en las Tablas 17 y 18.

[329] SHANNON. Op. cit.

Tabla 17. Entropía global de las predicciones — sin balanceo

Modelo	**PCA**	**%** H_{max}	**AF**	**%** H_{max}	**UMAP**	**%** H_{max}
Regresión Logística	1,0642	51,18 %	0,6270	30,15 %	0,1112	5,35 %
SVM-RBF	0,7366	35,42 %	0,9486	45,62 %	0,1391	6,69 %
CNN	0,5366	25,80 %	0,5979	28,75 %	0,2623	12,61 %
ViT	0,0993	4,78 %	0,0256	1,23 %	0,1089	5,24 %

Fuente: Elaboración propia.

Tabla 18. Entropía global de las predicciones — con balanceo

Modelo	**PCA**	**%** H_{max}	**AF**	**%** H_{max}	**UMAP**	**%** H_{max}
Regresión Logística	1,9524	93,89 %	2,0510	98,63 %	1,8658	89,73 %
SVM-RBF	2,0014	96,25 %	1,8133	87,20 %	1,9430	93,44 %
CNN	1,8609	89,49 %	1,7299	83,19 %	0,0000	0,00 %
ViT	1,1991	57,67 %	1,6174	77,78 %	0,6284	30,22 %

Fuente: Elaboración propia.

Sin balanceo, todos los modelos presentan entropías bajas, resultado que coincide con lo observado en los reportes de clasificación: los modelos tienden a concentrar sus predicciones en Normal, la clase más frecuente. El ViT registra la menor entropía, alcanzando apenas el $1{,}23\,\%$ de H_{max} sobre AF, mientras que la Regresión Logística sobre PCA obtiene la mayor con $51{,}18\,\%$, distribuyendo sus predicciones de forma más variada que el resto de los modelos. Al aplicar pesos de clase, la entropía aumenta considerablemente en todos los modelos, lo que indica que el balanceo redistribuye las predicciones de forma más uniforme entre clases. La Regresión Logística con AF alcanza la entropía más alta ($2{,}051$ nats, $98{,}63\,\%$ de H_{max}), siendo la representación con predicciones más equilibradas de todas las evaluadas. El caso más extremo en dirección opuesta es la CNN sobre UMAP con pesos de clase, cuya entropía de $0{,}00\,\%$ confirma el colapso total hacia Diabetes observado en la sección anterior. Una entropía alta no implica necesariamente un buen desempeño: un modelo puede distribuir sus predicciones entre muchas clases de forma incorrecta. La entropía global debe interpretarse siempre en conjunto con las métricas de clasificación por clase [330].

[330] Ibid.

5.4. RESULTADOS DE MODELOS SIN REDUCCIÓN DIMENSIONAL

En esta sección se presentan los resultados obtenidos al aplicar los modelos de clasificación directamente sobre las imágenes completas, sin aplicar ninguna técnica de reducción dimensional. Se evalúan los mismos cuatro modelos de la sección anterior, incorporando adicionalmente EfficientNetB3, una red convolucional preentrenada sobre millones de imágenes, para explorar si el conocimiento previo de la red permite obtener mejores resultados [331].

5.4.1. Resultados sin balanceo.

5.4.1.1. Regresión Logística. La Regresión Logística fue evaluada sobre vectores aplanados de $128 \times 128 \times 3 = 49.152$ características por imagen. Los resultados por clase se recogen en el Cuadro 9.

Cuadro 9. Reporte de clasificación — Regresión Logística sin balanceo, imágenes completas

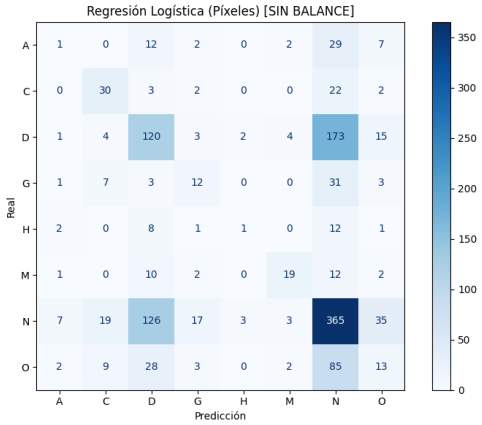
Clase	Precisión	Recall	F1-score	Support
A	0,07	0,02	0,03	53
C	0,43	0,51	0,47	59
D	0,39	0,37	0,38	322
G	0,29	0,21	0,24	57
H	0,17	0,04	0,06	25
M	0,63	0,41	0,50	46
N	0,50	0,63	0,56	575
O	0,17	0,09	0,12	142
macro avg	0,33	0,29	0,30	1279
Exactitud	0,4386			

Fuente: Elaboración propia.

[331] TAN. Op. cit.

El Cuadro 9 revela un desempeño más equilibrado que el observado con reducción dimensional: todas las clases obtienen predicciones correctas, incluyendo Degeneración macular e Hipertensión que anteriormente registraban F1-score de $0{,}00$. Normal lidera con F1-score de $0{,}56$ y un recall de $0{,}63$, identificando 362/575 casos. Miopía patológica obtiene un F1-score de $0{,}50$ con precisión de $0{,}63$, destacando como la clase minoritaria mejor identificada. Catarata alcanza $0{,}47$ con 30/59 aciertos posibles. Diabetes obtiene un F1-score de $0{,}38$ identificando 119/322 casos, mientras que Glaucoma alcanza $0{,}24$ con 12/57 casos. Degeneración macular e Hipertensión siguen siendo las clases más difíciles, con apenas 1 acierto cada una y F1-scores de $0{,}03$ y $0{,}06$ respectivamente.

Figura 19. Matriz de confusión Regresión Logística sin balanceo.



Fuente: Elaboración propia .

La Figura 19 muestra un patrón ya conocido: prácticamente todas las categorías presentan confusión hacia Normal y Diabetes. Degeneración macular dirige 29/53 casos hacia Normal y 12 hacia Diabetes, logrando apenas 1 acierto propio. Glaucoma identifica 12/57 casos correctamente pero dirige 31 hacia Normal. Hipertensión obtiene solo 1 acierto de 25 posibles, dispersando sus errores principalmente hacia Normal (12) y Diabetes (8). Miopía patológica logra 19 aciertos de 46 pero confunde 12 casos con Normal y 10 con Diabetes. Otras anomalías obtiene 13 aciertos propios dirigiendo 85 casos hacia Normal y 28 hacia Diabetes. Este patrón confirma que el modelo sigue siendo atraído por las clases dominantes incluso operando sobre imágenes completas.

La exactitud global de $0{,}4386$ es ligeramente inferior al mejor resultado con reducción dimensional ($0{,}4558$ con AF), aunque el macro avg de F1-score de $0{,}30$ es superior al

obtenido con PCA ($0{,}25$) y UMAP ($0{,}08$) sin balanceo, lo que indica que trabajar con imágenes completas mejora la distribución del desempeño entre clases aunque no la exactitud global.

5.4.1.2. Máquinas de Soporte Vectorial (SVM-RBF). Como en la sección de reducción dimensional, se utilizó el kernel RBF para mantener consistencia metodológica y permitir una comparación directa entre ambas secciones se usará el mismo kernel [332]. Los resultados por clase se recogen en el Cuadro 10.

Cuadro 10. Reporte de clasificación — SVM-RBF sin balanceo, imágenes completas

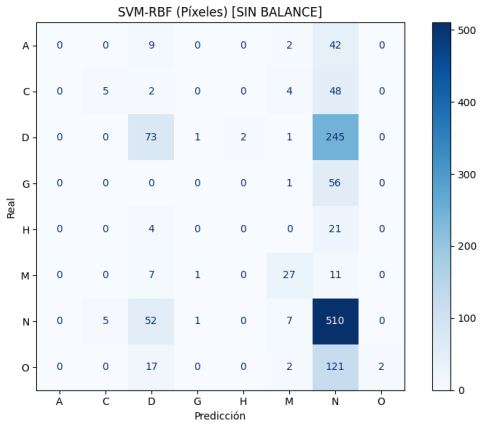
Clase	Precisión	Recall	F1-score	Support
A	0,00	0,00	0,00	53
C	0,50	0,08	0,14	59
D	0,45	0,23	0,30	322
G	0,00	0,00	0,00	57
H	0,00	0,00	0,00	25
M	0,61	0,59	0,60	46
N	0,48	0,89	0,63	575
O	1,00	0,01	0,03	142
macro avg	0,38	0,22	0,21	1279
Exactitud	0,4824			

Fuente: Elaboración propia.

El cuadro 10 muestra un patrón similar al observado con reducción dimensional: Normal y Miopía patológica son las clases mejor identificadas, con F1-scores de $0{,}63$ y $0{,}60$ respectivamente, mientras que Degeneración macular, Glaucoma e Hipertensión registran $0{,}00$. Un hallazgo particular es el de Otras anomalías: precisión perfecta de $1{,}00$ pero recall de apenas $0{,}01$, lo que indica que el modelo solo predice esa clase cuando está muy seguro, detectando apenas 1/142 casos. Diabetes obtiene un F1-score de $0{,}30$ identificando 74/322 casos, y Catarata apenas $0{,}14$ con 5/59 aciertos.

[332] HASTIE. Op. cit.

Figura 20. Matriz de confusión SVM-RBF sin balanceo, imágenes completas.



Fuente: Elaboración propia .

La Figura 20 muestra que el modelo prácticamente colapsa hacia Normal: la gran mayoría de los casos de Degeneración macular, Glaucoma, Hipertensión, Diabetes y Otras anomalías son clasificados en esa categoría, con apenas algunos aciertos aislados fuera de ella. Este comportamiento indica que sin estrategia de balanceo, el SVM-RBF no logra aprender suficientes patrones para distinguir las enfermedades minoritarias de la clase dominante.

La exactitud global de $0{,}4824$ supera a la Regresión Logística $(0{,}4386)$ y es comparable al mejor resultado con reducción dimensional ($0{,}4855$ con PCA). Sin embargo, el macro avg de F1-score de $0{,}21$ es inferior al de la Regresión Logística $(0{,}30)$, lo que indica que aunque el SVM-RBF obtiene mayor exactitud global, la Regresión Logística logra una distribución más equilibrada entre clases sobre las imágenes completas.

5.4.1.3. Red Neuronal Convolucional (CNN). La CNN fue evaluada sobre imágenes completas de 224×224 píxeles, manteniendo su estructura tridimensional para aprovechar la información espacial [333]. Los resultados por clase se recogen en el Cuadro 11.

[333] LeCUN, Yann et al. Op. cit., 1998.

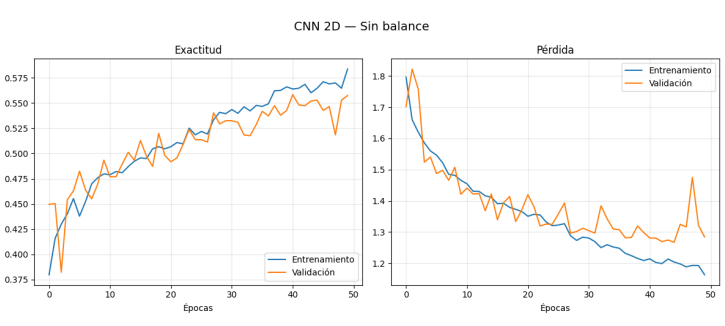
Cuadro 11. Reporte de clasificación — CNN sin balanceo, imágenes completas

Clase	Precisión	Recall	F1-score	Support
A	0,67	0,08	0,14	53
C	0,62	0,75	0,68	59
D	0,54	0,25	0,34	322
G	0,43	0,11	0,17	57
H	0,00	0,00	0,00	25
M	0,87	0,87	0,87	46
N	0,54	0,92	0,68	575
O	0,50	0,01	0,03	142
macro avg	0,52	0,37	0,36	1279
Exactitud	0,5528			

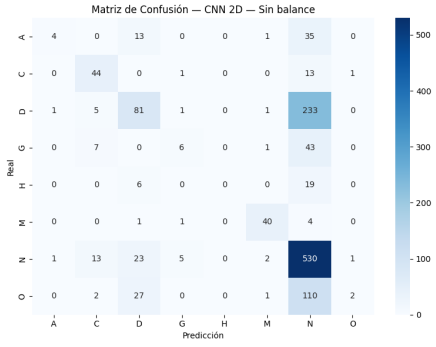
Fuente: Elaboración propia.

El Cuadro 11 muestra el mayor desempeño en exactitud global obtenido hasta ahora en esta sección, aunque con una distribución entre clases aún desigual. Miopía patológica lidera con un F1-score de $0{,}87$, el más alto registrado para esta clase en todo el análisis, identificando 40/46 casos con precisión y recall equilibrados. Normal y Catarata comparten F1-score de $0{,}68$, con Normal recuperando 529/575 casos y Catarata 44/59. Diabetes obtiene $0{,}34$ identificando 81/322 casos, mientras que Glaucoma alcanza apenas $0{,}17$ con 6/57 aciertos. Hipertensión registra $0{,}00$ sin ninguna predicción correcta, y Otras anomalías apenas $0{,}03$ con 1/142 acierto.

Figura 21. CNN sin balanceo, imágenes completas.



Curvas de pérdida y exactitud



Matriz de confusión

Fuente: Elaboración propia .

La Figura 21 muestra un aprendizaje progresivo sin sobreajuste marcado: tanto la exactitud de entrenamiento como la de validación crecen de forma conjunta desde $0{,}40$ hasta aproximadamente $0{,}57$ a lo largo de las 50 épocas. La curva de pérdida desciende de forma consistente en ambos conjuntos, aunque la validación presenta un pico notable alrededor de la época 45, lo que sugiere cierta sensibilidad al desbalanceo del conjunto de datos. En la matriz de confusión, el sesgo hacia Normal persiste: 233/322 casos de Diabetes, 43/57 de Glaucoma, 19/25 de Hipertensión y 110/142 de Otras anomalías son clasificados en esa categoría. Sin embargo, la diagonal es más activa que en los modelos anteriores, con Catarata obteniendo 44/59 aciertos y Miopía patológica 40/46.

La exactitud global de $0{,}5528$ es la primera vez en el análisis que un modelo supera el umbral de $0{,}50$, lo que indica que las imágenes completas proporcionan información más rica para la CNN que los vectores reducidos. El macro avg de F1-score de $0{,}36$ es también el más alto registrado hasta ahora en esta sección, aunque el desbalanceo sigue condicionando el desempeño en Hipertensión y Otras anomalías.

5.4.1.4. Transformador de Visión (ViT). El ViT fue evaluado sobre imágenes completas de 224×224 píxeles. Los resultados por clase se recogen en el Cuadro 12.

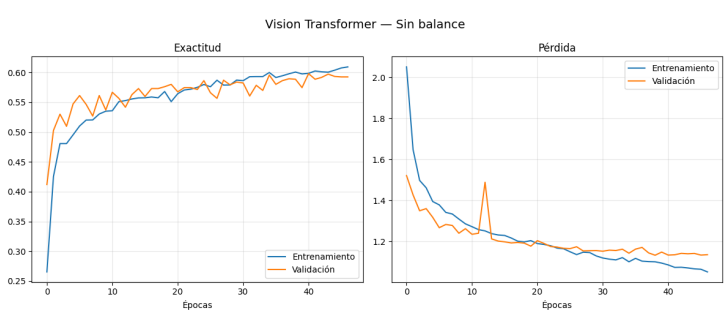
Cuadro 12. Reporte de clasificación — ViT sin balanceo, imágenes completas

Clase	Precisión	Recall	F1-score	Support
A	0,50	0,30	0,38	53
C	0,73	0,81	0,77	59
D	0,54	0,45	0,49	322
G	0,48	0,21	0,29	57
H	0,00	0,00	0,00	25
M	0,89	0,85	0,87	46
N	0,59	0,85	0,69	575
O	0,55	0,04	0,08	142
macro avg	0,53	0,44	0,45	1279
Exactitud	0,5887			

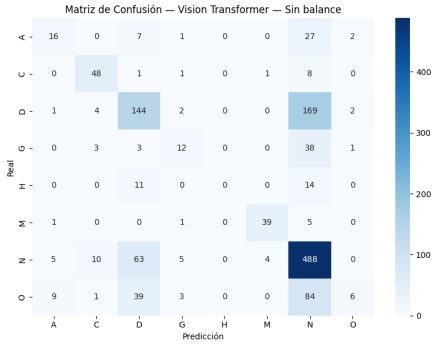
Fuente: Elaboración propia.

El cuadro 12 muestra el mejor desempeño en exactitud global de esta sección, aunque con distribución desigual entre clases. Miopía patológica mantiene un F1-score de $0{,}87$ identificando 39/46 casos, y Catarata mejora respecto a la CNN alcanzando $0{,}77$ con 48/59 aciertos. Diabetes mejora notablemente respecto a los modelos anteriores con un F1-score de $0{,}49$ con 145/322 aciertos. Degeneración macular obtiene $0{,}38$ con 16/53, siendo el primer modelo en esta sección que logra una detección notable para esta clase. Glaucoma alcanza $0{,}29$ con 12/57 aciertos, mientras que Hipertensión registra $0{,}00$ sin ninguna predicción correcta y Otras anomalías apenas $0{,}08$ con 6/142 aciertos.

Figura 22. ViT sin balanceo, imágenes completas.



Curvas de pérdida y exactitud



Matriz de confusión

Fuente: Elaboración propia .

La Figura 22 muestra un aprendizaje estable: ambas curvas parten desde $0{,}30$ y se estabilizan alrededor de $0{,}55$-$0{,}60$ a partir de la época 20, sin separación significativa entre entrenamiento y validación. La curva de pérdida presenta un pico notable en la validación alrededor de la época 12, tras el cual retoma una tendencia descendente estabilizándose cerca de $1{,}2$. En la matriz de confusión, la atracción hacia Normal persiste: 169/322 casos de Diabetes, 38/57 de Glaucoma y 84/142 de Otras anomalías son clasificados en esa categoría. Sin embargo, la diagonal es más activa que en los modelos anteriores: Diabetes obtiene 144/322 aciertos, Catarata 48/59 y Degeneración macular 16/53.

La exactitud global de $0{,}5887$ y el macro avg de F1-score de $0{,}45$ son los mejores resultados hasta ahora. A diferencia de lo observado con reducción dimensional, donde el ViT mostraba el peor desempeño, aquí se posiciona como un modelo competitivo, lo que sugiere que su arquitectura requiere la estructura espacial completa de las imágenes para funcionar correctamente.

5.4.1.5. EfficientNetB3. EfficientNetB3 fue evaluado sobre imágenes de 300×300 píxeles, aprovechando el conocimiento adquirido durante su entrenamiento previo sobre millones de imágenes [334]. Los resultados por clase se presentan en el Cuadro 13.

[334] TAN. Op. cit.

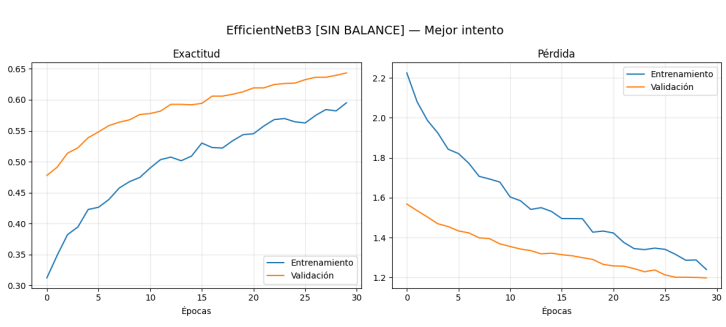
Cuadro 13. Reporte de clasificación — EfficientNetB3 sin balanceo, imágenes completas

Clase	Precisión	Recall	F1-score	Support
A	0,62	0,34	0,44	53
C	0,82	0,76	0,79	59
D	0,71	0,45	0,55	322
G	0,54	0,37	0,44	57
H	0,00	0,00	0,00	25
M	0,86	0,83	0,84	46
N	0,61	0,93	0,74	575
O	0,62	0,15	0,24	142
macro avg	0,60	0,48	0,50	1279
Exactitud	0,6435			

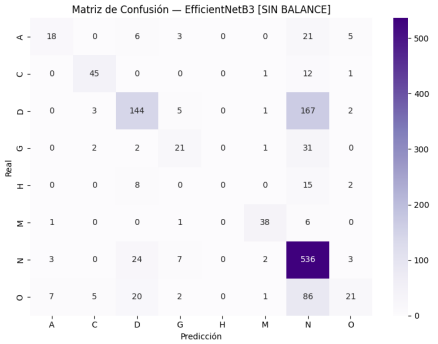
Fuente: Elaboración propia.

El Cuadro 13 muestra el mejor desempeño en exactitud global de toda la sección, aunque Hipertensión sigue registrando $0{,}00$ sin ninguna predicción correcta. Catarata alcanza un F1-score de $0{,}79$ con 45/59 aciertos, y Miopía patológica $0{,}84$ con 38/46. Normal obtiene $0{,}74$ identificando 535/575 casos. Diabetes logra su mejor resultado en esta sección con un F1-score de $0{,}55$ y 145/322 aciertos. Degeneración macular y Glaucoma obtienen F1-scores de $0{,}44$ con 18 y 21 aciertos respectivamente, mientras que Otras anomalías alcanza $0{,}24$ con 21/142 aciertos.

Figura 23. EfficientNetB3 sin balanceo, imágenes completas.



Curvas de pérdida y exactitud



Matriz de confusión

Fuente: Elaboración propia .

La Figura 23 muestra el efecto del preentrenamiento en las curvas: la validación parte desde $0{,}48$ y se mantiene por encima del entrenamiento durante la mayor parte del proceso, lo que indica que el modelo generaliza bien desde las primeras épocas. Ambas curvas convergen hacia $0{,}60$-$0{,}63$ sin sobreajuste importante, y la pérdida desciende de forma estable en ambos conjuntos hasta aproximadamente $1{,}2$. En la matriz de confusión, el patrón de atracción hacia Normal persiste: 167/322 casos de Diabetes, 31/57 de Glaucoma, 21/53 de Degeneración macular y 86/142 de Otras anomalías son clasificados en esa categoría. Sin embargo, la diagonal presenta los valores más altos observados hasta ahora, con Diabetes obteniendo 144/322 aciertos y Degeneración macular 18/53.

La exactitud global de $0{,}6435$ y el macro avg de F1-score de $0{,}50$ son los mejores resultados de toda la sección sin reducción dimensional, lo que confirma que el conocimiento previo de EfficientNetB3 proporciona una ventaja significativa sobre los modelos entrenados desde cero [335].

5.4.2. Resultados con balanceo. En esta sección se presentan los resultados obtenidos al aplicar pesos de clase. Esta estrategia produce una caída en la exactitud global, consecuencia esperada dado que el modelo distribuye su capacidad de aprendizaje entre todas las clases.

5.4.2.1. Regresión Logística. Con pesos de clase, la Regresión Logística alcanzó una exactitud de $0{,}3847$, inferior a la obtenida sin balanceo ($0{,}4386$). Los resultados por clase se recogen en el Cuadro 14.

[335] Ibid.

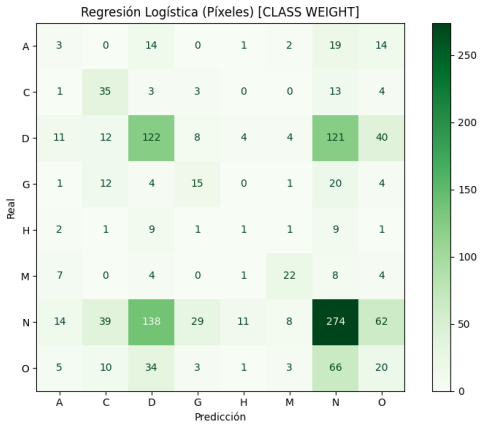
Cuadro 14. Reporte de clasificación — Regresión Logística con pesos de clase, imágenes completas

Clase	Precisión	Recall	F1-score	Support
A	0,07	0,06	0,06	53
C	0,32	0,59	0,42	59
D	0,37	0,38	0,38	322
G	0,25	0,26	0,26	57
H	0,05	0,04	0,05	25
M	0,54	0,48	0,51	46
N	0,52	0,48	0,50	575
O	0,13	0,14	0,14	142
macro avg	0,28	0,30	0,29	1279
Exactitud	0,3847			

Fuente: Elaboración propia.

El Cuadro 14 muestra que el balanceo produce una distribución más equilibrada entre clases. Todas las clases obtienen predicciones correctas, incluyendo Hipertensión que ahora alcanza un F1-score de $0{,}05$ con 1/25 aciertos. Catarata mejora su recall de $0{,}51$ a $0{,}59$ identificando 35/59 casos, y Glaucoma sube de $0{,}24$ a $0{,}26$ con 15/57 aciertos. Normal cae de un recall de $0{,}63$ a $0{,}48$, identificando 276/575 casos, mientras que Miopía patológica se mantiene estable con un F1-score de $0{,}51$.

Figura 24. Matriz de confusión Regresión Logística con pesos de clase, imágenes completas.



Fuente: Elaboración propia .

La Figura 24 muestra una distribución más dispersa que en la condición sin balanceo. Normal reduce sus aciertos a 274/575 y Diabetes obtiene 122, mientras que Degeneración macular dirige 19/53 casos hacia Normal y 14 hacia Otras anomalías, logrando apenas 3 aciertos propios. Hipertensión registra solo 1/25 aciertos, dispersando sus errores entre Diabetes y Normal. Otras anomalías obtiene 20/142 aciertos, aunque 66 de sus casos son clasificados como Normal y 34 como Diabetes.

La exactitud global de $0{,}3847$ y el macro avg de F1-score de $0{,}29$ son ligeramente inferiores a los obtenidos sin balanceo ($0{,}4386$ y $0{,}30$), lo que indica que en este modelo la estrategia de pesos de clase no mejora el desempeño global sobre imágenes completas.

5.4.2.2. Máquinas de Soporte Vectorial(SVM-RBF). Con pesos de clase, el SVM-RBF alcanzó una exactitud de $0{,}3268$, inferior a la obtenida sin balanceo ($0{,}4824$). Los resultados por clase se recogen en el Cuadro 15.

Cuadro 15. Reporte de clasificación — SVM-RBF con balanceo, imágenes completas

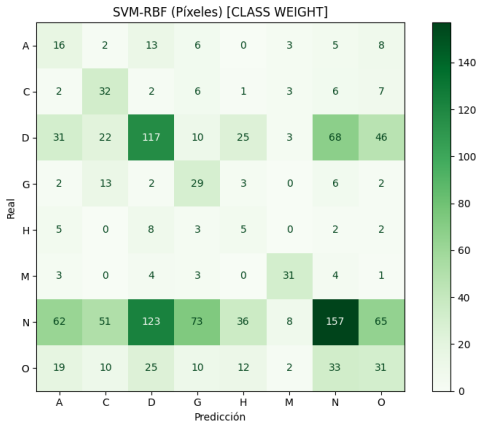
Clase	Precisión	Recall	F1-score	Support
A	0,11	0,30	0,17	53
C	0,25	0,54	0,34	59
D	0,40	0,36	0,38	322
G	0,21	0,51	0,29	57
H	0,06	0,20	0,09	25
M	0,62	0,67	0,65	46
N	0,56	0,27	0,37	575
O	0,19	0,22	0,20	142
macro avg	0,30	0,39	0,31	1279
Exactitud	0,3268			

Fuente: Elaboración propia.

El Cuadro 15 muestra que el balanceo produce el cambio más notable observado

hasta ahora en esta sección: todas las clases obtienen predicciones correctas, incluyendo Degeneración macular, Glaucoma e Hipertensión que sin balanceo registraban F1-score de $0{,}00$. Miopía patológica lidera con F1-score de $0{,}65$ identificando 31/46 casos. Glaucoma mejora de $0{,}00$ a $0{,}29$ con un recall de $0{,}51$ y 29/57 aciertos, e Hipertensión pasa de $0{,}00$ a $0{,}09$ con 5/25 aciertos. Degeneración macular alcanza $0{,}17$ con 16/53 aciertos. Normal, en cambio, cae de un recall de $0{,}89$ a $0{,}27$, identificando apenas 155/575 casos.

Figura 25. Matriz de confusión SVM-RBF con balanceo, imágenes completas.



Fuente: Elaboración propia .

La Figura 25 muestra una distribución más dispersa que en la condición sin balanceo. Normal reduce sus aciertos a 157/575, mientras que Diabetes obtiene 117 y Miopía patológica 31. Los errores se reparten entre múltiples categorías: Diabetes dirige 68 de sus casos hacia Normal y 46 hacia Otras anomalías, e Hipertensión dispersa sus errores principalmente entre Diabetes (8 casos) y Degeneración macular (5 casos). Otras anomalías obtiene 31 aciertos de 142, con 33 casos dirigidos hacia Normal y 25 hacia Diabetes.

La exactitud global de $0{,}3268$ es inferior a la obtenida sin balanceo, aunque el macro avg de F1-score de $0{,}31$ supera al de $0{,}21$ sin balanceo, lo que indica que la estrategia de pesos de clase mejora la distribución del desempeño entre clases aunque reduce la exactitud global.

5.4.2.3. Red Neuronal Convolucional (CNN). Con pesos de clase, la CNN alcanzó una exactitud de $0{,}2846$, inferior a la obtenida sin balanceo ($0{,}5528$). Los resultados por clase se recogen en el Cuadro 16.

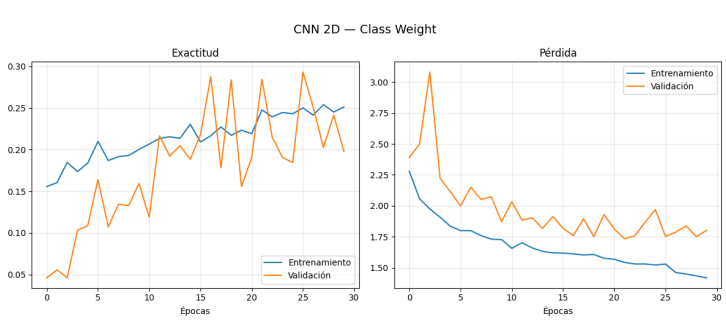
Cuadro 16. Reporte de clasificación — CNN con balanceo, imágenes completas

Clase	Precisión	Recall	F1-score	Support
A	0,11	0,43	0,17	53
C	0,47	0,78	0,59	59
D	0,35	0,34	0,34	322
G	0,16	0,26	0,20	57
H	0,04	0,44	0,07	25
M	0,77	0,87	0,82	46
N	0,60	0,20	0,30	575
O	0,19	0,06	0,09	142
macro avg	0,34	0,42	0,32	1279
Exactitud	0,2846			

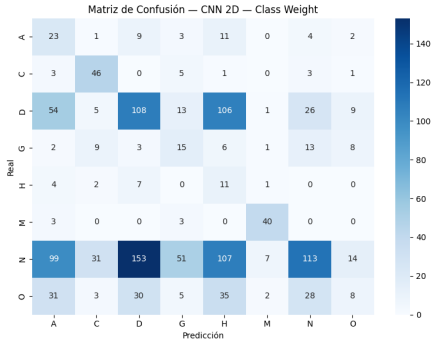
Fuente: Elaboración propia.

El Cuadro 16 muestra que el balanceo redistribuye las predicciones de forma notable. Miopía patológica mantiene un F1-score de $0{,}82$ identificando 40/46 casos. Hipertensión obtiene por primera vez predicciones correctas en la CNN, con un recall de $0{,}44$ y 11/25 aciertos, aunque con una precisión muy baja de $0{,}04$. Degeneración macular mejora de un recall de $0{,}08$ a $0{,}43$ con 23/53 aciertos. Catarata sube su recall de $0{,}75$ a $0{,}78$ con 46/59 aciertos, aunque su F1-score baja de $0{,}68$ a $0{,}59$ por la caída en precisión. Normal sufre la caída más pronunciada: de un recall de $0{,}92$ a apenas $0{,}20$, identificando solo 115/575 casos.

Figura 26. CNN con balanceo, imágenes completas.



Curvas de pérdida y exactitud



Matriz de confusión

Fuente: Elaboración propia .

La Figura 26 muestra un aprendizaje inestable: la curva de validación presenta oscilaciones muy pronunciadas sin seguir la tendencia de entrenamiento, y la pérdida de validación parte desde $3{,}0$ sin converger hacia la de entrenamiento, lo que indica que los pesos de clase generan dificultades de generalización en la CNN. En la matriz de confusión, la redistribución es notable pero problemática: 99/575 casos de Normal son clasificados como Degeneración macular, 153 como Diabetes y 107 como Hipertensión, lo que explica la caída drástica en el recall de esta clase. De igual forma, 54/322 casos de Diabetes son clasificados como Degeneración macular y 106 como Hipertensión, lo que indica que el modelo desarrolla una tendencia excesiva hacia las clases minoritarias.

La exactitud global de $0{,}2846$ y el macro avg de F1-score de $0{,}32$ son inferiores a los obtenidos sin balanceo ($0{,}5528$ y $0{,}36$), lo que indica que en la CNN sobre imágenes completas la estrategia de pesos de clase no mejora el desempeño global, aunque sí permite que Hipertensión y Degeneración macular obtengan predicciones correctas por primera vez.

5.4.2.4. Transformador de Visión (ViT). Con pesos de clase, el ViT alcanzó una exactitud de $0{,}4605$, inferior a la obtenida sin balanceo ($0{,}5887$). Los resultados por clase se recogen en el Cuadro 17.

Cuadro 17. Reporte de clasificación — ViT con balanceo, imágenes completas

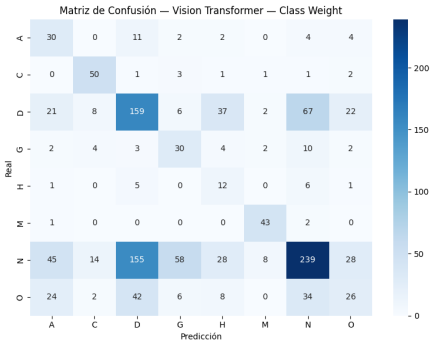
Clase	Precisión	Recall	F1-score	Support
A	0,24	0,57	0,34	53
C	0,64	0,85	0,73	59
D	0,42	0,49	0,46	322
G	0,29	0,53	0,37	57
H	0,13	0,48	0,21	25
M	0,77	0,93	0,84	46
N	0,66	0,42	0,51	575
O	0,31	0,18	0,23	142
macro avg	0,43	0,56	0,46	1279
Exactitud	0,4605			

Fuente: Elaboración propia.

El Cuadro 17 muestra el desempeño más equilibrado entre clases de todos los modelos evaluados bajo esta condición: ninguna clase registra F1-score de $0{,}00$. Miopía patológica lidera con $0{,}84$ identificando 43/46 casos, y Catarata alcanza $0{,}73$ con 50/59 aciertos. Hipertensión mejora de $0{,}00$ a $0{,}21$ con 12/25 aciertos, y Degeneración macular sube de un recall de $0{,}30$ a $0{,}57$ con 30/53 aciertos. Glaucoma pasa de $0{,}29$ a $0{,}37$ con 30/57 aciertos. Normal cae de un recall de $0{,}85$ a $0{,}42$, identificando 241/575 casos.

Figura 27. ViT con balanceo, imágenes completas.

Curvas de pérdida y exactitud



Matriz de confusión

Fuente: Elaboración propia .

La Figura 27 muestra un aprendizaje gradual: ambas curvas parten desde $0{,}20$ y se estabilizan alrededor de $0{,}40$-$0{,}45$, manteniéndose relativamente cercanas aunque con oscilaciones en la validación. La curva de pérdida desciende de forma estable en entrenamiento hasta cerca de 1, mientras que la validación se estabiliza alrededor de $1{,}4$-$1{,}5$, lo que sugiere sobreajuste moderado en las épocas finales. En la matriz de confusión, la distribución de aciertos es la más variada observada bajo la condición de balanceo: Miopía patológica obtiene 43/46 aciertos, Catarata 50/59, Degeneración macular 30/53 y Glaucoma 30/57. Los errores más relevantes se concentran en Diabetes, con 67 casos dirigidos hacia Normal y 37 hacia Hipertensión, y en Normal, con 155 casos clasificados como Diabetes.

La exactitud global de $0{,}4605$ cae menos que en los demás modelos respecto a la condición sin balanceo, y el macro avg de F1-score de $0{,}46$ supera al de $0{,}45$ sin balanceo, lo que confirma que el ViT es el modelo que mejor responde a la estrategia de pesos de clase.

5.4.2.5. EfficientNetB3. Con pesos de clase, EfficientNetB3 alcanzó una exactitud de $0{,}3909$, inferior a la obtenida sin balanceo ($0{,}6435$). Los resultados por clase se recogen en el Cuadro 18.

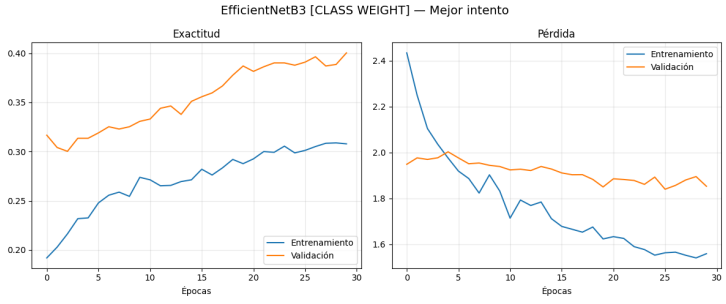
Cuadro 18. Reporte de clasificación — EfficientNetB3 con pesos de clase, imágenes completas

Clase	Precisión	Recall	F1-score	Support
A	0,34	0,66	0,45	53
C	0,44	0,88	0,59	59
D	0,49	0,16	0,24	322
G	0,21	0,60	0,31	57
H	0,05	0,52	0,09	25
M	0,51	0,96	0,67	46
N	0,64	0,45	0,53	575
O	0,26	0,08	0,13	142
macro avg	0,37	0,54	0,38	1279
Exactitud	0,3909			

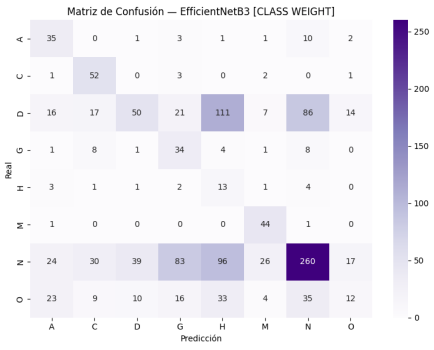
Fuente: Elaboración propia.

El Cuadro 18 muestra el efecto del balanceo sobre la distribución de predicciones. Miopía patológica lidera con F1-score de $0{,}67$ y un recall de $0{,}96$ identificando 44 de sus 46 casos. Catarata alcanza $0{,}59$ con 52/59 aciertos, y Degeneración macular mejora de un recall de $0{,}34$ a $0{,}66$ con 35/53 aciertos. Glaucoma sube de $0{,}37$ a $0{,}60$ en recall con 34/57 aciertos, e Hipertensión obtiene por primera vez un recall de $0{,}52$ con 13/25 aciertos, aunque con precisión muy baja de $0{,}05$. Sin embargo, Diabetes sufre la caída más pronunciada: de un recall de $0{,}45$ a apenas $0{,}16$, identificando solo 52/322 casos, Normal también disminuye su recall de $0{,}93$ a $0{,}45$, identificando 259/575 casos.

Figura 28. EfficientNetB3 con balanceo, imágenes completas.



Curvas de pérdida y exactitud



Matriz de confusión

Fuente: Elaboración propia .

La Figura 28 muestra el efecto del preentrenamiento en las curvas: la validación parte desde $0{,}30$ y se mantiene por encima del entrenamiento durante todo el proceso, alcanzando $0{,}38$-$0{,}40$ al final. La pérdida de entrenamiento desciende de forma estable hasta $1{,}6$, mientras que la de validación se mantiene con oscilaciones moderadas alrededor de $1{,}9$, lo que indica que los pesos de clase dificultan la generalización del modelo sobre datos nuevos. En la matriz de confusión, el patrón más llamativo es el de Diabetes: 111/322 casos son clasificados como Hipertensión y 86 como Normal, lo que explica su caída pronunciada en recall. Normal dispersa 83 de sus casos hacia Glaucoma y 96 hacia Hipertensión, evidenciando que el modelo desarrolla una tendencia excesiva hacia Hipertensión que afecta la detección de Diabetes y Normal.

La exactitud global de $0{,}3909$ y el macro avg de F1-score de $0{,}38$ son inferiores a los obtenidos sin balanceo ($0{,}6435$ y $0{,}50$), lo que indica que la estrategia de pesos de clase afecta el desempeño global de EfficientNetB3, aunque permite que Hipertensión y Degeneración macular obtengan predicciones correctas que sin balanceo eran muy limitadas.

5.4.3. Entropía global de las predicciones. Como en la sección de reducción dimensional, se calculó la entropía global de las predicciones para evaluar qué tan repartidas quedaron las predicciones de cada modelo entre las ocho categorías diagnósticas. Los resultados se presentan en las Tablas 19 y 20.

Tabla 19. Entropía global de las predicciones — sin balanceo, imágenes completas

Modelo	**Entropía**	**% de H_{max}**	**Clase dominante**
Regresión Logística	1,2695	61,05 %	Normal
SVM-RBF	0,6111	29,39 %	Normal
CNN	0,8278	39,81 %	Normal
ViT	1,0857	52,21 %	Normal
EfficientNetB3	1,0929	52,56 %	Normal

Fuente: Elaboración propia.

Tabla 20. Entropía global de las predicciones — con balanceo, imágenes completas

Modelo	**Entropía**	**% de H_{max}**	**Clase dominante**
Regresión Logística	1,6050	77,18 %	Normal
SVM-RBF	1,9522	93,88 %	Diabetes
CNN	1,8882	90,80 %	Diabetes
ViT	1,8259	87,81 %	Diabetes
EfficientNetB3	1,8743	90,13 %	Normal

Fuente: Elaboración propia.

Sin balanceo, todos los modelos presentan Normal como clase dominante. La Regresión Logística obtiene la mayor entropía ($61{,}05\,\%$) y el SVM-RBF la menor ($29{,}39\,\%$), confirmando su mayor concentración hacia Normal. En comparación con la sección de reducción dimensional, las entropías sin balanceo son más altas en general, lo que indica que las imágenes completas permiten una distribución más variada de las predicciones.

Al aplicar pesos de clase, la entropía aumenta en todos los modelos y SVM-RBF, CNN y ViT cambian su clase dominante de Normal a Diabetes, mientras que Regresión Logística y EfficientNetB3 mantienen Normal. Como en la sección anterior, una entropía alta no implica necesariamente un buen desempeño y debe interpretarse junto con las métricas por clase[336].

5.4.4. Modelos de exploración adicionales. Durante el desarrollo del trabajo se evaluaron tres arquitecturas preentrenadas adicionales, ResNet50, DenseNet121 e InceptionV3, así como el MLP de TensorFlow. La Tabla 21 presenta la comparativa de exactitud entre todos los modelos evaluados en esta sección. Los reportes de clasificación completos y matrices de confusión están disponibles en el repositorio del proyecto[337].

[336] SHANNON. Op. cit.

[337] `https://bit.ly/notebook-ODIR5K`

Tabla 21. Comparativa global de exactitud Modelos sin reducción dimensional

Modelo	**Sin balanceo**	**Con pesos de clase**
Regresión Logística	0,4386	0,3847
SVM-RBF	0,4824	0,3268
MLP (TensorFlow)	0,4808	0,1970
CNN 2D	0,5528	0,2846
ViT	0,5887	0,4605
ResNet50	0,6153	0,4668
DenseNet121	0,6255	0,4707
InceptionV3	0,6380	0,5395
EfficientNetB3	0,6435	0,3909

Fuente: Elaboración propia.

ResNet50 y **DenseNet121** muestran un comportamiento similar: Miopía patológica y Catarata son las clases mejor identificadas sin balanceo, con F1-scores de $0{,}82$ y $0{,}77$ para ResNet50, y $0{,}86$ y $0{,}77$ para DenseNet121. Con pesos de clase ambos mejoran en clases minoritarias como Glaucoma e Hipertensión manteniendo una exactitud global razonable.

InceptionV3 es el modelo con mayor estabilidad entre ambas condiciones, alcanzando un macro avg de F1-score de $0{,}51$ con pesos de clase. **MLP (TensorFlow)**, en cambio, obtiene el resultado más bajo de la sección con pesos de clase ($0{,}1970$), con tendencia a concentrar predicciones en Normal.

El patrón observado en los modelos principales se mantiene: las arquitecturas preentrenadas obtienen mejores resultados sin balanceo, mientras que el balanceo mejora el desempeño en clases minoritarias a costa de una reducción en la exactitud global.

5.5. COMPARATIVA DE DESEMPEÑO ENTRE ESCENARIOS

La Tabla 22 y la Tabla 24 comparan el macro avg de F1-score de cada modelo en ambos escenarios bajo las condiciones sin balanceo y con balanceo respectivamente. Se utiliza esta métrica por ser la más informativa en contextos de desbalanceo, ya que promedia el desempeño de forma equitativa entre las ocho categorías diagnósticas [338]. Las Tablas 23 y 25 muestran el porcentaje de variación respecto al escenario sin reducción dimensional, donde valores negativos indican caída en el desempeño y valores positivos indican mejora.

Tabla 22. Macro avg F1-score — Sin balanceo.

Modelo	**PCA**	**AF**	**UMAP**	**Sin RD**
Regresión Logística	0,25	0,20	0,08	0,30
SVM-RBF	0,24	0,30	0,09	0,21
CNN	0,23	0,22	0,12	0,36
ViT	0,11	0,09	0,09	0,45
EfficientNetB3	—	—	—	0,50

Fuente: Elaboración propia.

Tabla 23. Porcentaje de variación en macro avg F1-score respecto al escenario sin reducción dimensional — Sin balanceo.

Modelo	**Sin RD**	**$\triangle$ PCA**	**$\triangle$ AF**	**$\triangle$ UMAP**
Regresión Logística	0,30	$-16{,}7\,\%$	$-33{,}3\,\%$	$-73{,}3\,\%$
SVM-RBF	0,21	$+14{,}3\,\%$	$+42{,}9\,\%$	$-57{,}1\,\%$
CNN	0,36	$-36{,}1\,\%$	$-38{,}9\,\%$	$-66{,}7\,\%$
ViT	0,45	$-75{,}6\,\%$	$-80{,}0\,\%$	$-80{,}0\,\%$

Fuente: Elaboración propia.

[338] HASTIE. Op. cit.

Tabla 24. Macro avg F1-score — Con balanceo.

Modelo	**PCA**	**AF**	**UMAP**	**Sin RD**
Regresión Logística	0,16	0,20	0,13	0,29
SVM-RBF	0,27	0,30	0,14	0,31
CNN	0,31	0,28	0,05	0,32
ViT	0,11	0,13	0,10	0,46
EfficientNetB3	—	—	—	0,38

Fuente: Elaboración propia.

Tabla 25. Porcentaje de variación en macro avg F1-score respecto al escenario sin reducción dimensional — Con balanceo.

Modelo	**Sin RD**	**$\triangle$ PCA**	**$\triangle$ AF**	**$\triangle$ UMAP**
Regresión Logística	0,29	$-44{,}8\,\%$	$-31{,}0\,\%$	$-55{,}2\,\%$
SVM-RBF	0,31	$-12{,}9\,\%$	$-3{,}2\,\%$	$-54{,}8\,\%$
CNN	0,32	$-3{,}1\,\%$	$-12{,}5\,\%$	$-84{,}4\,\%$
ViT	0,46	$-76{,}1\,\%$	$-71{,}7\,\%$	$-78{,}3\,\%$

Fuente: Elaboración propia.

La diferencia más notable se observa en el ViT: al aplicar reducción dimensional, su macro avg de F1-score cae entre un $75{,}6\,\%$ y un $80{,}0\,\%$ sin balanceo, y entre un $71{,}7\,\%$ y un $78{,}3\,\%$ con balanceo, evidenciando que esta arquitectura pierde casi toda su capacidad discriminativa al operar sobre vectores comprimidos. En contraste, el SVM-RBF muestra una estabilidad destacable: con AF bajo balanceo su caída es de apenas $3{,}2\,\%$, siendo el modelo que mejor resiste la compresión dimensional. Respecto a los métodos de reducción, UMAP produce las caídas más pronunciadas en todos los modelos: la CNN llega a perder un $84{,}4\,\%$ de su desempeño con balanceo, y ningún modelo mejora respecto al escenario sin reducción bajo esta representación. PCA y AF muestran caídas más moderadas, y el SVM-RBF incluso mejora con AF sin balanceo en un $42{,}9\,\%$, confirmando que para los modelos clásicos la reducción dimensional puede ser una alternativa viable.

6. CONCLUSIONES

Los tres métodos de reducción evaluados lograron comprimir el espacio original de 49.152 características a 30 componentes, conservando más del 93 % de la diversidad informativa medida mediante entropía de Shannon. La selección de 30 componentes para PCA y AF se determinó mediante el Análisis Paralelo de Horn y el criterio de Yeomans-Golder, lo que garantizó que la reducción preservara más del $94\,\%$ de la varianza original. Para UMAP, la calidad de la reducción se verificó mediante métricas de confiabilidad y continuidad superiores a $0{,}96$, y para AF el índice KMO de $0{,}9796$ confirmó que los datos presentaban la estructura de correlación necesaria para aplicar el método. Aunque UMAP retuvo el mayor porcentaje de diversidad informativa ($96{,}24\,\%$), seguido de PCA ($94{,}57\,\%$) y AF ($93{,}55\,\%$), estas diferencias no generaron ventajas claras al clasificar las ocho categorías diagnósticas, lo que indica que los tres métodos preservan información suficiente para la tarea y ninguno demostró ser superior a los demás de forma clara.

Con reducción dimensional, los modelos clásicos como la Regresión Logística y el SVM-RBF mostraron resultados competitivos, siendo el SVM-RBF sobre PCA el que alcanzó la mayor exactitud global ($0{,}4855$). Sin reducción, las arquitecturas de aprendizaje profundo obtuvieron mejores resultados en exactitud global, con EfficientNetB3 liderando con $0{,}6435$. Sin embargo, esta ventaja de las redes profundas no aplica a todos los escenarios: la Regresión Logística obtuvo mejores resultados sobre representaciones reducidas que sobre imágenes completas, lo que sugiere que para los modelos clásicos la reducción dimensional funciona como preprocesamiento efectivo: elimina ruido y facilita el aprendizaje. Las redes neuronales, en cambio, necesitan la imagen completa para aprovechar su capacidad de detectar patrones espaciales complejos.

La aplicación de pesos de clase mejoró la detección de enfermedades minoritarias en todos los modelos, permitiendo que patologías como Hipertensión, Glaucoma y Degeneración macular obtuvieran predicciones correctas donde antes fallaban. Sin

embargo, esta mejora tuvo un costo directo sobre la exactitud global, lo que evidencia que mejorar la detección de enfermedades poco frecuentes y mantener una alta exactitud general son objetivos difíciles de alcanzar al mismo tiempo. Este resultado demuestra que en problemas médicos con clases desbalanceadas, la exactitud global es insuficiente como métrica única y debe complementarse con otras métricas que evalúen el desempeño por clase.

Los resultados evidencian una diferencia entre modelos clásicos y arquitecturas profundas. Las redes neuronales como la CNN y el ViT están diseñadas para detectar patrones visuales en imágenes completas, por lo que pierden su ventaja cuando operan sobre vectores comprimidos. En ese escenario, el SVM-RBF y la Regresión Logística igualaron o superaron a las redes neuronales en varios casos. Por el contrario, sobre imágenes completas las redes recuperan su ventaja gracias a su capacidad de identificar patrones visuales en la imagen original, y EfficientNetB3 se destaca como la más efectiva gracias al conocimiento previo adquirido durante su preentrenamiento sobre millones de imágenes.

La utilidad de la reducción dimensional depende del contexto en que se aplique. Cuando los recursos computacionales son limitados, la reducción dimensional es una alternativa válida, pues permite entrenar modelos clásicos con resultados competitivos usando un número muy reducido de características, reduciendo considerablemente el tiempo de entrenamiento y el uso de memoria. Sin embargo, cuando el objetivo es maximizar el desempeño diagnóstico, especialmente en enfermedades minoritarias, trabajar con las imágenes completas y arquitecturas preentrenadas produce resultados superiores. La reducción dimensional es una herramienta útil y eficiente, pero no reemplaza la información contenida en la imagen original cuando se dispone de los recursos para procesarla.

Al analizar el desempeño por categoría diagnóstica, EfficientNetB3 con pesos de clase sobre imágenes completas fue el modelo más efectivo para Miopía patológica, Catarata, Degeneración macular e Hipertensión, destacando por su capacidad de identificar correctamente casos de enfermedades poco frecuentes. Para Diabetes, el ViT con pesos de clase sobre imágenes completas logró el mejor equilibrio entre precisión y

recall. En Glaucoma, la Regresión Logística con pesos de clase sobre UMAP fue el modelo más efectivo dentro del escenario con reducción dimensional. Otras anomalías resultó ser la categoría más difícil en todos los experimentos, lo que se explica porque agrupa enfermedades visualmente muy diferentes entre sí, dificultando que cualquier modelo aprenda un patrón común.

EfficientNetB3 con pesos de clase es el modelo más recomendable cuando se busca el mayor desempeño diagnóstico, ya que combina el conocimiento visual previo del preentrenamiento con la sensibilidad hacia clases minoritarias que aporta el balanceo. La Regresión Logística con pesos de clase sobre representaciones reducidas es la alternativa más práctica cuando los recursos son limitados: es rápida de entrenar, fácil de interpretar para un profesional médico y produce resultados competitivos en múltiples patologías. El SVM-RBF, por su parte, mostró estabilidad entre ambos escenarios, funcionando de manera similar con o sin reducción, lo que lo convierte en una opción confiable cuando se busca un modelo robusto sin importar el tipo de representación.

7. RECOMENDACIONES

A partir de los resultados obtenidos en este trabajo, se proponen las siguientes recomendaciones para mejorar la clasificación de enfermedades oculares en investigaciones futuras.

Explorar otras formas de manejar el desbalanceo de clases. En este trabajo se usaron pesos de clase como estrategia para que los modelos prestaran más atención a las enfermedades menos frecuentes. Aunque esto mejoró la detección de algunas categorías minoritarias, también redujo el desempeño general en la mayoría de los modelos. Una alternativa es generar imágenes artificiales de las clases con menos datos mediante técnicas de aumentación avanzada para equilibrar el conjunto de datos. Sin embargo, en el caso de imágenes médicas esto debe hacerse con mucho cuidado para no generar imágenes que no correspondan a condiciones reales, lo que requiere conocimiento clínico especializado que excedía el alcance de este trabajo. Esta alternativa requiere el acompañamiento de profesionales de la salud visual para garantizar la validez clínica de las imágenes generadas [339].

Mejorar el preprocesamiento de las imágenes. En este trabajo se aplicaron transformaciones básicas a las imágenes como rotaciones, cambios de brillo y zoom para aumentar la variedad de los datos de entrenamiento. Sin embargo, las imágenes de fondo de ojo tienen características particulares relacionadas con la iluminación y el contraste que podrían aprovecharse mejor con técnicas más específicas para este tipo de imágenes. Por ejemplo, mejorar el contraste de las imágenes o resaltar estructuras como los vasos sanguíneos podría ayudar a los modelos a identificar mejor las diferencias entre enfermedades. Estas técnicas no se implementaron por requerir conocimiento clínico que garantice que los cambios aplicados no afecten la información diagnóstica [340].

Considerar trabajar con menos categorías diagnósticas. El conjunto de datos uti-

[339] SHORTEN. Op. cit.
[340] GONZALEZ. Op. cit.

lizado incluye ocho categorías, algunas de las cuales corresponden a enfermedades del cuerpo que tienen manifestaciones en los ojos, como la Diabetes y la Hipertensión, mientras que otras son enfermedades propias del ojo como la Catarata, el Glaucoma o la Miopía patológica. Las primeras tienden a ser más difíciles de identificar visualmente porque sus señales en las imágenes no son tan claras ni específicas. Reducir el número de categorías a las enfermedades estrictamente oculares podría simplificar el problema y mejorar los resultados. En este trabajo se mantuvo la estructura original del conjunto de datos para respetar su definición y poder comparar los resultados con otros estudios que lo utilizan como referencia [341].

Probar con un mayor número de componentes en la reducción dimensional. En este trabajo se redujeron las imágenes a 30 componentes para las tres técnicas evaluadas. Una prueba exploratoria con 50 y 100 componentes no mostró mejoras significativas, pero evaluar configuraciones mayores, como 100 o 200 componentes, podría permitir conservar más información relevante para distinguir enfermedades poco frecuentes, aunque a un costo computacional considerablemente mayor.

Analizar con más detalle las arquitecturas preentrenadas. Entre los modelos evaluados como exploración adicional, InceptionV3 mostró el mejor comportamiento tanto sin balanceo como con pesos de clase, lo que sugiere que podría ser un buen punto de partida para futuras investigaciones. Se recomienda dedicarles un análisis más detallado, con distintas configuraciones y estrategias para imágenes de fondo de ojo[342].

Usar un conjunto de datos más grande. Una de las principales limitaciones de este trabajo fue el tamaño del conjunto de datos, que contaba con poco más de 6.000 imágenes distribuidas entre ocho categorías. Este número de imágenes es relativamente pequeño para entrenar modelos de clasificación con tantas categorías, especialmente cuando algunas de ellas tienen muy pocos casos. Contar con un mayor número de imágenes, o combinar varios conjuntos de datos, permitiría a los modelos aprender patrones más claros y mejorar su capacidad para identificar correctamente todas las enfermedades, incluyendo las más difíciles de detectar[343].

[341] SHANGGONG MEDICAL TECHNOLOGY CO., LTD. Op. cit.
[342] TAN. Op. cit.
[343] LITJENS. Op. cit.

Incorporar herramientas que expliquen las decisiones del modelo. En el contexto médico no basta con que un modelo acierte en su clasificación, sino que también es importante poder entender por qué tomó esa decisión. Saber en qué parte de la imagen se basó el modelo para identificar una enfermedad permite verificar si está aprendiendo los patrones correctos y genera mayor confianza en su uso clínico. Incorporar esta capacidad es especialmente relevante para cualquier trabajo futuro que busque aplicar estos modelos en entornos reales de diagnóstico[344].

Estas recomendaciones parten de las limitaciones identificadas en este trabajo y apuntan hacia líneas de investigación concretas que podrían mejorar la precisión del diagnóstico automático de enfermedades oculares.

[344] Ibid.

BIBLIOGRAFÍA

ANDERSON, T. W. *An Introduction to Multivariate Statistical Analysis*. Wiley, 2003.

BISHOP, Christopher M. *Pattern Recognition and Machine Learning*. Springer, 2006.

BOYD, Stephen y VANDENBERGHE, Lieven. *Convex Optimization*. Cambridge University Press, 2004.

BROWNE, Michael W. Asymptotically Distribution-Free Methods for the Analysis of Covariance Structures. En: *British Journal of Mathematical and Statistical Psychology*. 1984, vol. 37, nro. 1, pp. 62-83.

CASELLA, George y BERGER, Roger L. *Statistical Inference*. 2 ed. Pacific Grove, CA: Duxbury Press, 2002. ISBN 978-0534243128.

CHAWLA, N. V.; BOWYER, K. W.; HALL, L. O. y KEGELMEYER, W. P. SMOTE: Synthetic Minority Over-sampling Technique. En: *Journal of Artificial Intelligence Research*. 2002, vol. 16, pp. 321-357.

CHOLLET, François et al. Keras [en línea]. 2015. Disponible en: https://keras.io

CORTES, Corinna y VAPNIK, Vladimir. Support-vector networks. En: *Machine Learning*. 1995, vol. 20, nro. 3, pp. 273-297.

COVER, T. y THOMAS, J. *Elements of Information Theory*. Wiley, 2006.

DEMPSTER, A. P.; LAIRD, N. M. y RUBIN, D. B. Maximum Likelihood from Incomplete Data via the EM Algorithm. En: *Journal of the Royal Statistical Society. Series B (Methodological)*. 1977, vol. 39, nro. 1, pp. 1-38.

DENG, Jia et al. ImageNet: A large-scale hierarchical image database. En: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009, pp. 248-255.

DOSOVITSKIY, Alexey et al. An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. En: *International Conference on Learning Representations (ICLR)*. 2021.

GONZALEZ, Rafael C. y WOODS, Richard E. *Digital Image Processing*. 4 ed. Pearson, 2018.

GOODFELLOW, Ian; BENGIO, Yoshua y COURVILLE, Aaron. *Deep Learning*. Cambridge, MA: MIT Press, 2016. Disponible en: https://www.deeplearningbook.org

HAIR, J. et al. *Multivariate Data Analysis*. Cengage, 2019.

HARMAN, Harry H. *Modern Factor Analysis*. 3 ed. Chicago: University of Chicago Press, 1976.

HASTIE, Trevor; TIBSHIRANI, Robert y FRIEDMAN, Jerome. *The Elements of Statistical Learning*. Springer, 2009.

HAYKIN, Simon. *Neural Networks and Learning Machines*. 3 ed. Upper Saddle River, NJ: Prentice Hall, 2009.

HE, Haibo y GARCIA, Edwardo A. Learning from Imbalanced Data. En: *IEEE Transactions on Knowledge and Data Engineering*. 2009, vol. 21, nro. 9, pp. 1263-1284.

HE, Kaiming; ZHANG, Xiangyu; REN, Shaoqing y SUN, Jian. Deep Residual Learning for Image Recognition. En: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770-778.

HERNÁNDEZ LALINDE, J. D. et al. Sobre el uso adecuado del coeficiente de corre-

lación de Pearson. En: *Sociedad Venezolana de Farmacología Clínica y Terapéutica*. 2018.

HORN, John L. A Rationale and Test for the Number of Factors in Factor Analysis. En: *Psychometrika*. 1965, vol. 30, nro. 2, pp. 179-185.

HORN, Roger A. y JOHNSON, Charles R. *Matrix Analysis*. 2 ed. Cambridge University Press, 2013.

HOSMER, David W.; LEMESHOW, Stanley y STURDIVANT, Rodney X. *Applied Logistic Regression*. Wiley, 2013.

HOWARD, Andrew G. et al. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. En: *arXiv preprint arXiv:1704.04861*. 2017.

HSU, Chih-Wei y LIN, Chih-Jen. A comparison of methods for multiclass support vector machines. En: *IEEE Transactions on Neural Networks*. 2002, vol. 13, nro. 2, pp. 415-425.

JAMES, Gareth; WITTEN, Daniela; HASTIE, Trevor y TIBSHIRANI, Robert. *An Introduction to Statistical Learning: with Applications in R*. New York: Springer, 2013. ISBN 978-1-4614-7137-0.

JOHNSON, Justin M. y KHOSHGOFTAAR, Taghi M. Survey on deep learning with class imbalance. En: *Journal of Big Data*. 2019, vol. 6, nro. 1, pp. 1-54.

JOHNSON, Richard A. y WICHERN, Dean W. *Applied Multivariate Statistical Analysis*. 6 ed. Pearson, 2007.

JOLLIFFE, Ian T. *Principal Component Analysis*. 2 ed. Springer, 2002.

JÖRESKOG, Karl G. Some Contributions to Maximum Likelihood Factor Analysis. En: *Psychometrika*. 1967, vol. 32, nro. 4, pp. 443-482.

KAISER, Henry F. The varimax criterion for analytic rotation in factor analysis. En: *Psychometrika*. 1958, vol. 23, pp. 187-200.

KAISER, Henry F. A second generation little jiffy. En: *Psychometrika*. 1970, vol. 35, nro. 4, pp. 401-415.

KAISER, Henry F. An index of factorial simplicity. En: *Psychometrika*. 1974, vol. 39, nro. 1, pp. 31-36.

KOTSIANTIS, S. B.; ZAHARAKIS, I. y PINTELAS, P. Supervised machine learning: A review of classification techniques. En: *Informatica*. 2007, vol. 31, pp. 249-268.

LARXEL. Ocular Disease Recognition ODIR5K [en línea]. Kaggle, 2020. [Consultado: 2024].Disponible en: https://www.kaggle.com/datasets/andrewmvd/ocular-disease-recognition-odir5k

LAWLEY, D. N. y MAXWELL, A. E. *Factor Analysis as a Statistical Method*. 2 ed. London: Butterworths, 1971.

LeCUN, Yann; BOTTOU, Léon; BENGIO, Yoshua y HAFFNER, Patrick. Gradient-based learning applied to document recognition. En: *Proceedings of the IEEE*. 1998, vol. 86, nro. 11, pp. 2278-2324.

LeCUN, Yann; BENGIO, Yoshua y HINTON, Geoffrey. Deep Learning. En: *Nature*. 2015, vol. 521, pp. 436-444.

LIN, Min; CHEN, Qiang y YAN, Shuicheng. Network In Network. En: *International Conference on Learning Representations (ICLR)*. 2014.

LITJENS, Geert et al. A survey on deep learning in medical image analysis. En: *Medical Image Analysis*. 2017, vol. 42, pp. 60-88.

LÓPEZ, Juan Carlos. *Redes neuronales artificiales: fundamentos, modelos y aplica-*

ciones. Madrid: Editorial Alfaomega, 2008.

McCULLAGH, P. y NELDER, J. *Generalized Linear Models*. Chapman and Hall, 1989.

McINNES, Leland; HEALY, John y MELVILLE, James. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction [en línea]. 2020. Disponible en: https://arxiv.org/abs/1802.03426

McINNES, Leland; HEALY, John y MELVILLE, James. How UMAP Works [en línea]. 2024. Disponible en: https://umap-learn.readthedocs.io/en/latest/how$_u map_w orks.html$

MUNKRES, James R. *Topología*. 2 ed. Prentice Hall, 2002.

MURPHY, Kevin P. *Machine Learning: A Probabilistic Perspective*. MIT Press, 2012.

Ocular Disease Recognition (ODIR-5K) [Anónimo]. Kaggle Dataset. Disponible en: https://www.kaggle.com/datasets/andrewmvd/ocular-disease-recognition-odir5k

PEDREGOSA, Fabian et al. Scikit-learn: Machine Learning in Python. En: *Journal of Machine Learning Research*. 2011, vol. 12, pp. 2825-2830.

PINILLA, J. O. y RICO, A. F. ¿Pearson y Spearman, coeficientes intercambiables? En: *Comunicaciones en Estadística*. 2021.

RUDIN, Walter. *Principios de Análisis Matemático*. 3 ed. McGraw-Hill, 1976.

RUKUNDO, Olivier. Effects of Image Size on Deep Learning. En: *arXiv preprint arXiv:2101.11508*. 2021.

SABOTTKE, Carl F. y SPIELER, Bradley M. The Effect of Image Resolution on Deep Learning in Radiography. En: *Radiology: Artificial Intelligence*. 2020, vol. 2, nro. 1, p. e190015.

SANDLER, Mark et al. MobileNetV2: Inverted Residuals and Linear Bottlenecks. En: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 4510-4520.

SHANNON, Claude E. A Mathematical Theory of Communication. En: *Bell System Technical Journal*. 1948, vol. 27, nro. 3, pp. 379-423.

SHANGGONG MEDICAL TECHNOLOGY CO., LTD. ODIR-2019: Ocular Disease Intelligent Recognition Dataset [en línea]. Grand Challenge, 2019. [Consultado: 2024]. Disponible en: https://odir2019.grand-challenge.org/dataset/

SHEN, Dinggang; WU, Guorong y SUK, Heung-Il. Deep Learning in Medical Image Analysis. En: *Annual Review of Biomedical Engineering*. 2017, vol. 19, pp. 221-248.

SHORTEN, Connor y KHOSHGOFTAAR, Taghi M. A survey on Image Data Augmentation for Deep Learning. En: *Journal of Big Data*. 2019, vol. 6, nro. 1, pp. 1-48.

STASIS, S.; STABLES, R. y HOCKMAN, J. Semantically Controlled Adaptive Equalisation in Reduced Dimensionality Parameter Space. En: *Applied Sciences*. 2016, vol. 6, nro. 4, p. 116.

STRANG, Gilbert. *Introduction to Linear Algebra*. 5 ed. Wellesley, MA: Wellesley-Cambridge Press, 2016.

TAN, Mingxing y LE, Quoc V. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. En: *Proceedings of the 36th International Conference on Machine Learning (ICML)*. 2019, pp. 6105-6114.

TING, Daniel S.W. et al. Development and validation of a deep learning system for diabetic retinopathy and related eye diseases. En: *JAMA*. 2017, vol. 318, nro. 22, pp. 2211-2223.

TUSELL, Fernando. *Análisis multivariante*. F. Tusell, 1999.

VAN DER MAATEN, L. J. P.; POSTMA, E. O. y VAN DEN HERIK, H. J. Dimensionality Reduction: A Comparative Review. En: *Journal of Machine Learning Research*. 2009, vol. 10, pp. 1-41.

VAPNIK, Vladimir N. *Statistical Learning Theory*. New York: Wiley, 1998.

VASWANI, Ashish et al. Attention Is All You Need. En: *Advances in Neural Information Processing Systems*. 2017, vol. 30.

WASSERMAN, Larry. *All of Statistics: A Concise Course in Statistical Inference*. New York: Springer, 2004.

WORLD HEALTH ORGANIZATION. *World Report on Vision* [en línea]. Geneva: WHO, 2019. Disponible en: https://www.who.int/publications-detail-redirect/9789241516570

YEOMANS, K. A. y GOLDER, P. A. The Guttman-Kaiser criterion as a predictor of the number of common factors. En: *Journal of the Royal Statistical Society. Series D (The Statistician)*. 1982, vol. 31, nro. 3, pp. 221-229.

ANEXOS

ANEXO A. CÓDIGO FUENTE

El código fuente de esta investigación está disponible en el siguiente repositorio de GitHub: `https://bit.ly/notebook-ODIR5K`

El conjunto de datos ODIR-5K utilizado está disponible en: `https://www.kaggle.com/datasets/andrewmvd/ocular-disease-recognition-odir5k`