

**IMPLEMENTACIÓN DE UN DETECTOR DE ROSTROS EN POSICIÓN  
FRONTAL MEDIANTE UN ALGORITMO DE BOOSTING**

HUGO ARMANDO GUALDRÓN COLMENARES

ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA  
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS  
UNIVERSIDAD INDUSTRIAL DE SANTANDER  
BUCARAMANGA

2011

**IMPLEMENTACIÓN DE UN DETECTOR DE ROSTROS EN POSICIÓN  
FRONTAL MEDIANTE UN ALGORITMO DE BOOSTING**

**HUGO ARMANDO GUALDRÓN COLMENARES**

Trabajo de grado para optar al título de  
Ingeniería de Sistemas

Director:

PhD(c) Lola Xiomara Bautista Rozo

Codirectores:

MSc. Darío José Delgado Quintero

Ing. Lina María Garcés Rodríguez

ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA  
FACULTAD DE INGENIERÍAS FISICOMECÁNICAS  
UNIVERSIDAD INDUSTRIAL DE SANTANDER  
BUCARAMANGA

2011

A Dios por ser la luz que alumbra mi sendero,

A toda mi familia, a mi hijo Santiago, a mi esposa Diana, mi padre Hugo, mi madre Juanita, mi hermano Cristian y mi hermana Zareth, por su compañía y apoyo en cada uno de mis logros y equivocaciones; por siempre combatir a mi lado las adversidades de la vida.

A la profesora Lola Xiomara Bautista por la confianza depositada y por compartir sus conocimientos.

Al Ing. Darío Delgado y a la Ing. Lina María Garces por su apoyo infinito que permitió la realización exitosa de este proyecto.

A mis compañeros del Grupo de Investigación en Ingeniería Biomédica GIIB, Andres, Dario, Lina, Cindy, Angel, Lizeth, Luis, Manuel, Wilfredo, Betty, por creer que la investigación es la única forma de construir un mejor país para todos.

## RESUMEN

**TÍTULO:**

IMPLEMENTACIÓN DE UN DETECTOR DE ROSTROS EN POSICIÓN FRONTAL MEDIANTE UN ALGORITMO DE BOOSTING \*

**AUTOR:**

HUGO ARMANDO GUALDRON COLMENARES \*\*

**PALABRAS CLAVE:**

Detección de Rostros, Boosting, Imagen Integral, Características Tipo Haar

**DESCRIPCIÓN:**

En los últimos años se han desarrollado diversos algoritmos que permiten la detección de rostros, los cuales contribuyen al diseño de sistemas de reconocimiento e identificación de personas. Para el desarrollo de estos algoritmos se ha recurrido a diferentes técnicas de inteligencia artificial que permiten la construcción de modelos computacionales basados en Máquinas de aprendizaje[6][7][2],[10] y Boosting[17][3].

En este trabajo se realizó la implementación de un detector de rostros mediante la utilización de un algoritmo de Boosting [4] y un clasificador multinivel. Este detector se ajusta a una arquitectura que a través de la imagen integral y los características de Haar, realiza operaciones rápidas por bloques más no por píxeles, disminuyendo el tiempo de respuesta en la clasificación.

El detector rechaza eficazmente las imágenes que no son rostros con una especificidad del 0.9998 y un valor predictivo negativo de 0.9998. Adicionalmente tiene una capacidad considerable para detectar rostros en ambientes heterógeneos con una sensibilidad de  $0.7076 \pm 0.2803$  y un valor predictivo positivo de  $0.6767 \pm 0.2618$ , y un tiempo de respuesta por imagen analizada de  $0.1456 \pm 0.0419$  milisegundos, lo cual indica que el detector aproximadamente verifica 6868.13 imágenes por segundo, donde una imagen es un ejemplo a verificar por el clasificador que puede variar en tamaño ya que el clasificador tiene la capacidad de analizar muestras a diferentes resoluciones indicando si son o no un rostro.

---

\*TRABAJO DE GRADO

\*\*FACULTAD DE INGENIERIAS FISICOMECAICAS. ESCUELA DE INGENIERIA DE SISTEMAS E INFORMATICA. DIRECTOR: PhD(c) LOLA XIOMARA BAUTISTA ROZO. CODIRECTORES: MSc. DARIO JOSE DELGADO QUINTERO, Ing. LINA MARIA GARCES RODRIGUEZ

## SUMMARY

**TITLE:**

FACE DETECTION IMPLEMENTATION IN FRONT POSITION USING BOOSTING ALGORITHM\*\*\*

**AUTHORS:**

HUGO ARMANDO GUALDRON COLMENARES \*\*\*\*

**KEYWORDS:**

Face detection, Boosting, Integral Image, Haar-Like features

**DESCRIPTION:**

In recent years several algorithms have been developed, these are able to make face detection, which contribute to the system design of recognition and persons' identification. For the development of these algorithms have been used several techniques of artificial intelligence that allow the construction of computational models based on machine learning [6][7][2],[10] and Boosting[17][3].

This work was carried out to implement a face detection using a Boosting algorithm [4] and multilevel classifier. This detector adjusts to an architecture that across the integral image and Haar's characteristics, makes rapid operations for blocks more not for pixels, reducing the response time in the classification.

The detector rejects effectively the images that are not faces with a specificity of 0.9998 and a negative predictive value of 0.9998. In addition, it has a considerable aptitude to detect faces in heterogeneous environments with a sensibility of 0.7076 and a positive predictive value of  $0.6767 \pm 0.2618$ , and a response time of  $0.1456 \pm 0.0419$  milliseconds per image analyzed, indicating that the detector checks 6868.13 images per second, where an image is an example to verify the classifier which can vary in size, since the classifier is capable of analyzing samples at different resolutions indicating whether or not a face.

---

\*\*\* WORK DEGREE

\*\*\*\* FACULTY OF ENGINEERINGS PHYSICOMECHANICAL. SCHOOL OF SYSTEMS ENGINEERING. DIRECTOR: PhD(c) LOLA XIOMARA BAUTISTA ROZO. CODIRECTORS: MSc. DARIO JOSE DELGADO QUINTERO, Ing. LINA MARIA GARCES RODRIGUEZ

# Índice general

<b>1. Introducción</b>	<b>12</b>
<b>2. Marco teórico y estado el arte</b>	<b>14</b>
2.1. Detección de Rostros . . . . .	14
2.2. Detección Rápida de objetos mediante una cascada de clasificadores simples. . . . .	15
2.2.1. La Imagen Integral . . . . .	16
2.2.2 Características de Haar . . . . .	16
2.2.3. Relación entre Características de Haar y la Imagen Integral . . . .	18
2.2.4. Clasificador entrenado por Boosting . . . . .	23
2.2.5. Clasificador en Cascada . . . . .	26
<b>3. Planteamiento del problema</b>	<b>30</b>
<b>4. Justificación</b>	<b>32</b>
<b>5. Planteamiento de la solución</b>	<b>33</b>
5.1. Base de Datos . . . . .	33
5.2. Características de Haar . . . . .	34
5.3. Normalización de los datos . . . . .	35
5.4. Entrenamiento del Detector de Rostros . . . . .	40

<b>6. Resultados</b>	<b>45</b>
<b>7. Discusión de los Resultados</b>	<b>58</b>
<b>8. Conclusiones y Recomendaciones</b>	<b>60</b>
8.1. Conclusiones . . . . .	60
8.2. Recomendaciones . . . . .	61
<b>Bibliografía</b>	<b>63</b>

# Índice de figuras

1.	Detección de Rostros Humanos Fuente: <a href="http://www.sony.com/">http://www.sony.com/</a> . . . .	15
2.	Valor de imagen integral en un punto $(x_k, y_m)$ . Fuente: <a href="http://research.microsoft.com/.../violajones_IJCV.pdf">http:// research.microsoft.com/.../violajones_IJCV.pdf</a> . . . . .	16
3.	Modelos de referencia empleados para la generación de las características de Haar. Fuente: Elaboración propia . . . . .	17
4.	Variante del modelo tipo 1. Fuente: Elaboración propia . . . . .	17
5.	Zona de una característica, donde $P_1, P_2, P_3, P_4$ son los puntos significativos de la region $z_j$ Fuente: Elaboración propia . . . . .	18
6.	Característica de Haar basada en el modelo de referencia (3), (Ver Figura 3), donde los puntos $P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9$ son los vértices de las regiones . . . . .	20
7.	Clasificador en cascada donde $S_1, S_2, \dots, S_n$ son clasificadores fuertes y $W_1, W_2, \dots, W_m$ son los clasificadores débiles. Fuente: Elaboración propia . . . . .	27
8.	Ejemplos de rostros de la base de datos CBCL FACE DETECTION #1 .	34
9.	Ejemplos de no rostros de la base de datos CBCL FACE DETECTION #1	34
10.	Imagen de referencia para normalización de un punto $P_{x,y}$ de una subimagen . . . . .	38
11.	Ejemplo de normalización de una subimagen . . . . .	39

12.	Análisis de Sensibilidad por capa del Detector en cascada de rostros con la base de datos CBCL FACE DETECTION #1 . . . . .	47
13.	Análisis de Especificidad por capa del Detector en cascada de rostros con la base de datos CBCL FACE DETECTION #1 . . . . .	48
14.	Análisis del Valor Predictivo Positivo VPP por capa del Detector en cascada de rostros con la base de datos CBCL FACE DETECTION #1 . . . . .	49
15.	Análisis del Valor Predictivo Negativo VPN por capa del Detector en cascada de rostros con la base de datos CBCL FACE DETECTION #1 . . . . .	49
16.	Prueba #1 . . . . .	51
17.	Prueba #2 . . . . .	52
18.	Prueba #3 . . . . .	54
19.	Prueba #4 . . . . .	55
20.	Prueba #5 . . . . .	56

# 1. Introducción

La detección de rostros es uno de los pilares en la solución de problemas de identificación de personas en sistemas de seguridad biométricos, en el reconocimiento de expresiones faciales y en la interacción hombre-máquina. La detección de rostros es una derivación de la detección de objetos relacionada a la visión por computador, que permite la localización de rostros humanos mediante la caracterización de zonas relevantes como ojos, nariz y boca. [15]

Aunque la detección de rostros es muy conocida debido a su gran uso en las cámaras fotográficas para la captura óptima de rostros con mejoras en los niveles de exposición y enfoque, el conocimiento de los algoritmos y la matemática de soporte es crucial debido a que la detección es normalmente una etapa que presenta mayor complejidad en sistemas de reconocimiento facial.

En este documento se explican los conceptos teóricos y lineamientos a seguir para implementar un detector de rostros. El documento se distribuye de la siguiente forma: en la primera parte se enuncia formalmente el problema a resolver, en donde se muestra el marco teórico y el estado del arte de la problemática que se abordará en este trabajo; luego se presenta la justificación del trabajo en el sentido de la pertinencia que tiene este tema de investigación y la validez del mismo para el desarrollo de un trabajo de pregrado en Ingeniería de Sistemas.

Se enuncia el planteamiento del problema y a continuación la solución propuesta. Por último se muestran los resultados obtenidos de este trabajo, un discusión acerca de los resultados y las conclusiones y recomendaciones originadas a partir de este trabajo de investigación.

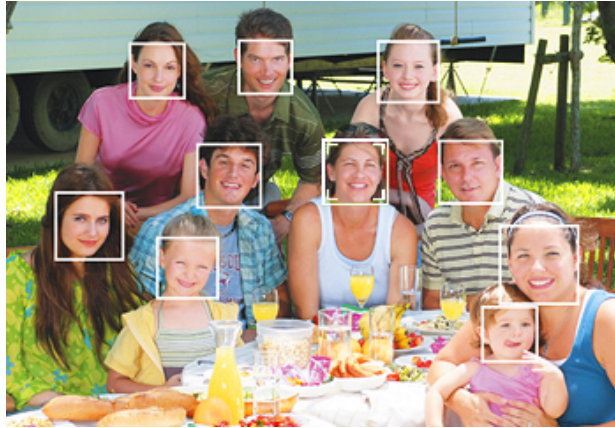
## **2. Marco teórico y estado el arte**

Las técnicas de visión artificial están presentes en diversas de aplicaciones, entre las que se encuentran el control de tráfico vehicular, detección de enfermedades, reconocimiento de individuos mediante huellas dactilares y la detección de rostros, donde se enmarca el proyecto que se presenta a continuación.

### **2.1. Detección de Rostros**

La detección de rostros ha sido un problema de interés que ha cuestionado a los investigadores y que a lo largo del tiempo los ha inspirado en la construcción de modelos computacionales que han evolucionado hasta formar sistemas de alta precisión y velocidad.

El objetivo principal de la detección de rostros es la identificación de rostros humanos sin importar la posición, tamaño, orientación y condiciones de iluminación presentes en una imagen, como se ilustra en la figura 1[16].



**Figura 1:** Detección de Rostros Humanos

Fuente: <http://www.sony.com/>

Recientemente el interés de facultar a los sistemas artificiales de la capacidad de identificación de objetos, ha consolidado frameworks como el propuesto por Papa-georgiou, Oren y Poggio que mediante máquinas de soporte vectorial realizan detección de vehiculos y personas [8], y el propuesto por Viola y Jones que emerge como mejora y se enfoca en la detección rápida de objetos mediante una cascada de clasificadores simples [13], que se explica a continuación.

## **2.2. Detección Rápida de objetos mediante una cascada de clasificadores simples.**

Este framework propone un modelo robusto que mediante una búsqueda exhaustiva de patrones dentro de un conjunto de ejemplos de entrenamiento, extrae las características que con menor error separan el espacio a clasificar, las organiza en una arquitectura en cascada que identifica y rechaza rápidamente los elementos no deseados, reevaluando con mayor exigencia los elementos que posiblemente son los objetos a identificar. Para realizar estas operaciones en el menor tiempo posible se

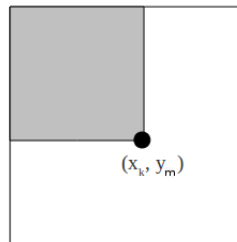
soporta en la técnica de la imagen integral, que en conjunto con las características de Haar aceleran el proceso de evaluación de los elementos de estudio [11].

### 2.2.1 La Imagen Integral

La imagen integral  $ii$  se define como la sumatoria de los pixeles inmediatamente anteriores a un punto  $(x_k, y_m)$  inclusive, lo cual se expresa de la siguiente manera

$$ii(x_k, y_m) = \sum_{x=1}^k \sum_{y=1}^m i(x, y) \quad (1)$$

donde  $i(\bullet, \bullet)$  es la imagen original y  $(x_k, y_m)$  es un punto cualquiera dentro de la imagen, como se aprecia en la Figura 2.

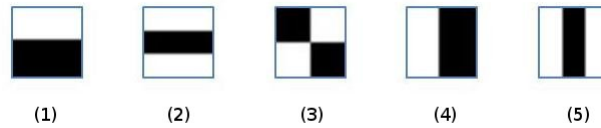


**Figura 2:** Valor de imagen integral en un punto  $(x_k, y_m)$ .

Fuente: [http:// research.microsoft.com/... /violajones\\_IJCV.pdf](http://research.microsoft.com/.../violajones_IJCV.pdf)

### 2.2.2 Características de Haar

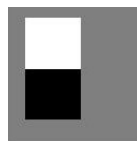
Las características de Haar son mediciones que se realizan a las imágenes a través de las cuales se extrae información relevante, que permite caracterizar conjuntos de datos. Conceptualmente se aprecian como regiones rectangulares adyacentes en una localización específica dentro de un ventana de detección. Fueron desarrolladas por Viola y Jones a partir de la idea de las características Wavelet de Papageorgiou[8].



**Figura 3:** Modelos de referencia empleados para la generación de las características de Haar.  
Fuente: Elaboración propia

Las características se originan a partir de cinco modelos de referencia (Figura 3). El valor de las características de dos regiones (Figura 3, subfiguras 1 y 4)) en una imagen se obtiene de la diferencia entre las dos regiones rectangulares. En el caso de las características formadas por tres regiones rectangulares (Figura 3, subfiguras 2 y 5)) se resta la región del centro de las externas.

En general, una característica es una matriz formada por dos tipos de regiones (blancas, negras) y un fondo (gris), donde la sumatoria del valor de las regiones negras se substrahe de la sumatoria de las blancas, y el fondo gris de la característica contribuye únicamente en la variación de la posición de la zona útil, que presenta una resolución mínima de 8 píxeles (Figura 4).

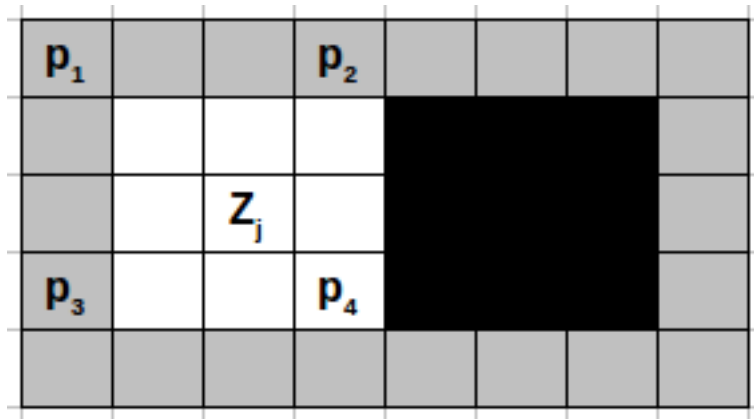


**Figura 4:** Variante del modelo tipo 1. Fuente: Elaboración propia

Las características de Haar optimizan los procesos de clasificación debido a que calcular una característica mediante la utilización de la imagen integral reduce el número de operaciones necesarias para el cálculo de la medición de la característica,

pues su operación por bloques, mas no por pixeles, es eficiente.

### 2.2.3. Relación entre Características de Haar y la Imagen Integral





**Figura 5:** Zona de una característica, donde  $P_1, P_2, P_3, P_4$  son los puntos significativos de la region  $z_j$  Fuente: Elaboración propia

La aceleración del cálculo de las características sobre la imagen integral  $ii$  impacta en el tiempo de operación del detector. Analizando en detalle la configuración de una característica (Figura 5), se observan los vértices de las regiones, donde una región  $z_j$  es un recuadro (negro o blanco) de la característica formado por vértices  $P_1, P_2, P_3, P_4$ , que evaluada sobre la  $ii$  de una imagen  $i$  equivale a:

$$g(ii, z_j) = ii(P_1) + ii(P_4) - (ii(P_2) + ii(P_3)) \quad (2)$$

donde  $g(ii, z_j)$  es la función que evalúa una región  $z_j$  de una característica sobre una imagen integral. Mediante las propiedades de la  $ii$  [11], se reduce el cálculo de una característica sobre la imagen integral a:

$$f(ii) = \sum_{j=1}^z a * g(ii, z_j) \quad (3)$$

Característica	x	y	*
	8	3	1
	13	3	-2
	16	3	1
	8	6	-2
	13	6	4
	16	6	-2
	8	11	1
	13	11	-2
	16	11	1
	9	0	1
	12	0	-2
	15	0	2
	18	0	-1
	9	18	-1
	12	18	2
	15	18	-2
	18	18	1

**Cuadro 1:** Representación de las características

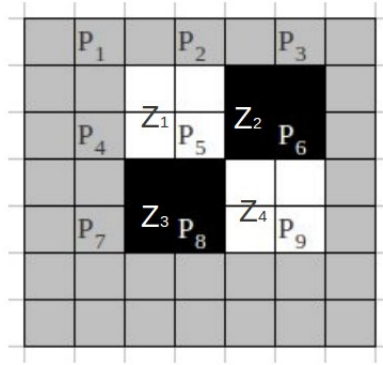
Fuente: Elaboración propia

donde  $a$  es el tipo de zona; si  $a = 1$  la zona es blanca y si  $a = -1$  es negra. [13]

Una característica de tamaño  $n \times n$  pixeles puede representarse mediante una matriz de tamaño  $P \times 3$ , donde  $P$  es el total de puntos que forman las regiones negras y blancas. El Cuadro 1 muestra la representación de una característica, en la que las

dos primeras columnas son las coordenadas  $(x, y)$  de los vértices de las zonas de la característica y la última columna el coeficiente asociado a este, que se obtiene de la simplificación del modelo de referencia, y se explica a continuación.

La construcción de la matriz que representa la característica de Haar se hace a partir de un modelo de referencia (Figura 6).



**Figura 6:** Característica de Haar basada en el modelo de referencia (3), (Ver Figura 3), donde los puntos  $P_1, P_2, P_3, P_4, P_5, P_6, P_7, P_8, P_9$  son los vértices de las regiones

Se aplica la ecuación 3 donde

$$f(ii) = g(ii, z_1) - g(ii, z_3) - (g(ii, z_2) + g(ii, z_4)) \quad (4)$$

$$g(ii, z_1) = ii(P_5) + ii(P_1) - (ii(P_2) + ii(P_4)) \quad (5)$$

$$g(ii, z_2) = ii(P_6) + ii(P_2) - (ii(P_5) + ii(P_3)) \quad (6)$$

$$g(ii, z_3) = ii(P_8) + ii(P_4) - (ii(P_5) + ii(P_7)) \quad (7)$$

$$g(ii, z_4) = ii(P_9) + ii(P_5) - (ii(P_6) + ii(P_8)) \quad (8)$$

$$\begin{aligned}
f(ii) = & \quad ii(P_1) \\
& -2 * ii(P_2) \\
& \quad +ii(P_3) \\
& -2 * ii(P_4) \\
& +4 * ii(P_5) \\
& -2 * ii(P_6) \\
& \quad +ii(P_7) \\
& -2 * ii(P_8) \\
& \quad +ii(P_9)
\end{aligned} \tag{9}$$

Por tanto la matriz que representa esta característica de Haar es

x	y	*
$P_{1x}$	$P_{1y}$	+1
$P_{2x}$	$P_{2y}$	- 2
$P_{3x}$	$P_{3y}$	+1
$P_{4x}$	$P_{4y}$	- 2
$P_{5x}$	$P_{5y}$	+4
$P_{6x}$	$P_{6y}$	- 2
$P_{7x}$	$P_{7y}$	+1
$P_{8x}$	$P_{8y}$	- 2
$P_{9x}$	$P_{9y}$	+1

**Cuadro 2:** Representación de una característica de Haar que presenta el modelo de referencia  
(3). Fuente: Elaboración propia

La matrices de los cinco modelos de referencia aplicando el mismo proceso de simplificación anterior son

Modelo de Referencia	x	y	*
	$P_{1x}$	$P_{1y}$	+1
	$P_{2x}$	$P_{2y}$	-1
	$P_{3x}$	$P_{3y}$	-2
	$P_{4x}$	$P_{4y}$	+2
	$P_{5x}$	$P_{5y}$	+1
	$P_{6x}$	$P_{6y}$	-1
	$P_{1x}$	$P_{1y}$	+1
	$P_{2x}$	$P_{2y}$	-1
	$P_{3x}$	$P_{3y}$	-2
	$P_{4x}$	$P_{4y}$	+2
	$P_{5x}$	$P_{5y}$	+2
	$P_{6x}$	$P_{6y}$	-2
	$P_{7x}$	$P_{7y}$	-1
	$P_{8x}$	$P_{8y}$	+1
	$P_{1x}$	$P_{1y}$	+1
	$P_{2x}$	$P_{2y}$	-2
	$P_{3x}$	$P_{3y}$	+1
	$P_{4x}$	$P_{4y}$	-2
	$P_{5x}$	$P_{5y}$	+4
	$P_{6x}$	$P_{6y}$	-2
	$P_{7x}$	$P_{7y}$	+1
	$P_{8x}$	$P_{8y}$	-2
	$P_{9x}$	$P_{9y}$	+1

Modelo de Referencia	x	y	*
	$P_{1x}$	$P_{1y}$	+1
	$P_{2x}$	$P_{2y}$	-2
	$P_{3x}$	$P_{3y}$	+1
	$P_{4x}$	$P_{4y}$	-1
	$P_{5x}$	$P_{5y}$	+2
	$P_{6x}$	$P_{6y}$	-1
		$P_{1x}$	$P_{1y}$
$P_{2x}$		$P_{2y}$	-2
$P_{3x}$		$P_{3y}$	+2
$P_{4x}$		$P_{4y}$	-1
$P_{5x}$		$P_{5y}$	-1
$P_{6x}$		$P_{6y}$	+2
$P_{7x}$		$P_{7y}$	-2
$P_{8x}$		$P_{8y}$	+1

**Cuadro 3:** Representación Matricial de los Modelos de Referencia de las características de Haar. Fuente: Elaboración propia

## 2.2.4. Clasificador entrenado por Boosting

Un clasificador entrenado por Boosting se basa en la posibilidad de crear un clasificador fuerte formado de un conjunto de clasificadores débiles, donde un clasificador fuerte es un clasificador con la capacidad de separar conjuntos de datos con el mínimo error, mientras que un clasificador débil es aquel que ligeramente separa los conjuntos de datos.

Existen diferentes algoritmos de entrenamiento que se basan en el concepto de boosting tales como Adaboost [4], LPBoost [9] [14], BrownBoost [5], entre otros. El Adaboost es el más popular de todos debido a que fue el primer algoritmo que construyó un clasificador aplicando el concepto de boosting.

El entrenamiento de este clasificador se efectúa mediante un proceso exhaustivo de selección, que consiste en buscar el mejor clasificador débil por iteración en función del mínimo error obtenido, a través de la evaluación sobre un conjunto de entrenamiento.

Un clasificador débil es una función  $h(i, f, p, \Theta)$  definida así:

$$h(i, f, p, \Theta) = \begin{cases} 1 & \text{si } pf(i) < p\Theta \\ 0 & \text{en otro caso} \end{cases} \quad (10)$$

donde se evalúa una imagen  $i$  sobre una característica de Haar  $f$ , y si el valor retornado aplicando la polaridad  $p$  es mayor al umbral ( $\Theta$ ) obtenido del entrenamiento, la imagen  $i$  es un rostro. El umbral ( $\Theta$ ) es un valor constante que se obtiene del entrenamiento y es quien explícitamente separa los rostros de las demás imágenes, que dependiendo de la polaridad, estarán por encima o por debajo, donde  $p \in \{-1, 1\}$ .

El entrenamiento del clasificador usando boosting se realiza mediante el Algoritmo (1) planteado por Viola y Jones; donde se parte de un conjunto de  $n$  imágenes  $x_1, x_2, \dots, x_n$  (rostros y no rostros) etiquetadas con valores  $y \in \{1, 0\}$ , para rostros y no rostros respectivamente.

Luego a partir de esto, se define un valor representativo (peso) para cada imagen que determina el nivel de importancia de cada muestra en el entrenamiento, donde cada clase (rostros y no rostros) inicia con un valor específico de acuerdo al número

de muestras por clase.

A partir de esto se realiza un proceso iterativo donde se escoge el mejor clasificador débil por iteración en función del peso actual de las muestras de entrenamiento, que aumenta para las muestras mal clasificadas y disminuye para las muestras clasificadas correctamente; ocasionando que el clasificador de la iteración posterior se enfoque en las muestras que los clasificadores débiles de iteraciones previas no logran clasificar adecuadamente, fortaleciendo la estructura final del clasificador fuerte entrenado por *Boosting*.

---

**Algoritmo 1** Entrenamiento por *boosting* [13]

---

**Entrada:** Imágenes de Entrenamiento  $x_1, \dots, x_n$  etiquetados con  $y_1, \dots, y_n$ , donde  $y_j = 0, 1$  para rostros y no rostros respectivamente.

- Inicialización de pesos  $w_{1,j} = \frac{1}{2m}, \frac{1}{2l}$ , para  $j = 1, \dots, (m + l)$ ; donde  $m$  y  $l$  son el número de rostros y no rostros de entrada, respectivamente.

**para**  $t = 1, \dots, T$  **hacer**

- Normalización de pesos

$$w_{t,j} = \frac{w_{t,j}}{\sum_{k=1}^n w_{t,k}}$$

donde  $w_{t,j}$  es peso del clasificador débil  $j$  en la iteración  $t$

- Selección del clasificador débil con el mínimo error  $h_t$ , donde el error por clasificador débil  $e_k$  equivale a

$$e_k = \sum_{j=1}^n w_{t,j} |h(i, f, p, \Theta) - y_j|$$

- Actualización de pesos

$$w_{t+1} = w_t * \beta_t \text{ donde } \beta_t = \frac{e_k}{1-e_k}$$

**fin para**

- El clasificador final es

$$C(x) = \begin{cases} 1 & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0 & \text{en otro caso} \end{cases}$$

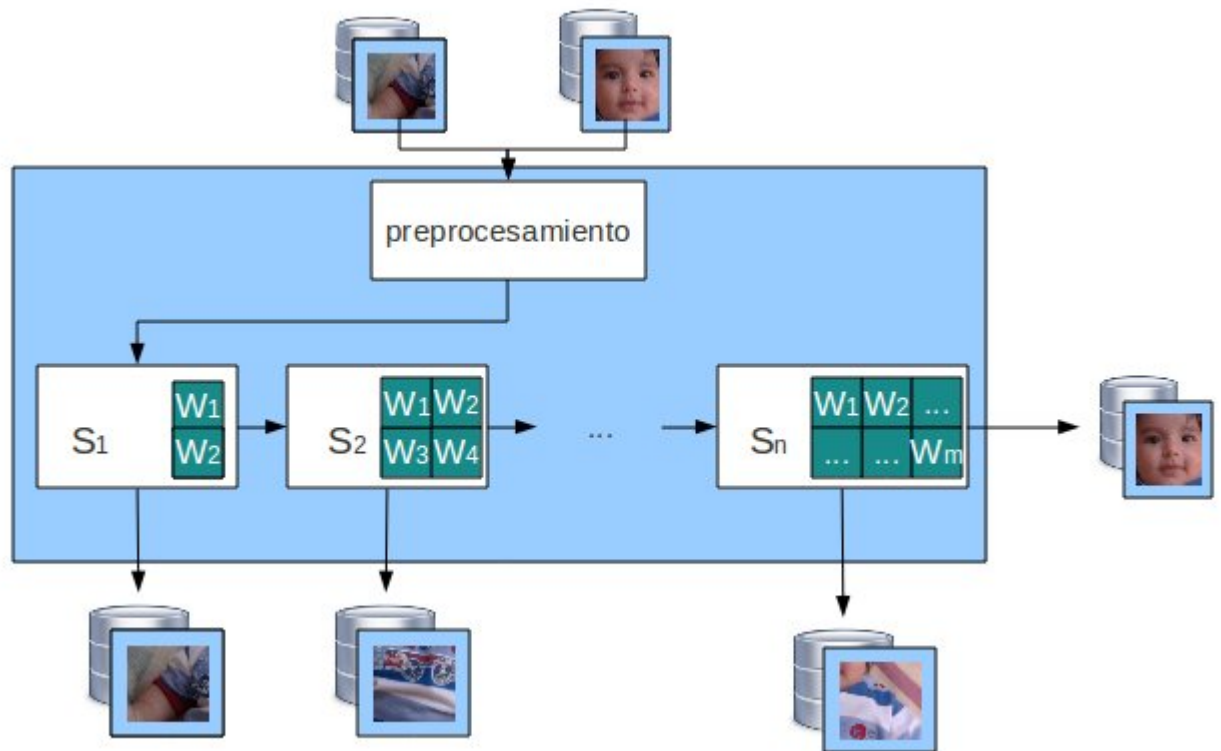
donde  $\alpha_t = \log \frac{1}{\beta_t}$

---

### 2.2.5. Clasificador en Cascada

Un clasificador en Cascada es una combinación de clasificadores fuertes, donde cada clasificador fuerte es entrenado mediante un algoritmo de boosting. El clasificador en cascada es la estructura final del detector de rostros por lo cuál estos dos términos a lo largo del documento se tratarán de manera indiferente. Los clasifica-

dores fuertes se organizan de menor a mayor complejidad, debido a que el tiempo de respuesta crece en función del número de clasificadores débiles que presente definidos. Cada clasificador fuerte tiene la capacidad de rechazar las imágenes que considera no rostros mientras que las posibles detecciones son reevaluadas en un clasificador más riguroso.



**Figura 7:** Clasificador en cascada donde  $S_1, S_2, \dots, S_n$  son clasificadores fuertes y  $W_1, W_2, \dots, W_m$  son los clasificadores débiles. Fuente: Elaboración propia

De acuerdo a la Figura 7, se evidencia que previo a la clasificación, una imagen cualquiera requiere de un preprocesamiento, que consiste en una normalización de la varianza de la imagen, con el fin de disminuir el efecto ocasionado por diferentes condiciones de iluminación.

El entrenamiento del clasificador en cascada se realiza mediante el Algoritmo ( 2), donde a partir de dos conjuntos de datos, rostros  $P$  y no rostros  $N$  y unos parámetros que definen los criterios de convergencia del algoritmo (máximo aceptable de falsos positivos por capa  $f$ , mínimo aceptable de detección por capa  $d$  y falsos positivos sobre todas las capas  $F_{target}$ , que se encuentran en un intervalo  $[0, 1]$ ), se crea un clasificador en cascada.

El algoritmo en su proceso iterativo va aumentando el número de clasificadores débiles por clasificador fuerte hasta que la estructura del detector que se está construyendo cumple con un mínimo de falsos positivos en cada capa. El detector en cascada de rostros termina de entrenarse cuando los falsos positivos del detector son inferiores al máximo aceptable de falsos positivos sobre todas las capas  $F_{target}$ .

---

**Algoritmo 2** Entrenamiento Clasificador en Cascada[13]

---

**Entrada:** Los valores para  $f$ , falsos positivos aceptables por capa y  $d$ , mínimo aceptable de detección por capa.

**Entrada:** Falsos positivos sobre todas las capas  $F_{target}$

**Entrada:**  $P$  = conjunto de rostros

**Entrada:**  $N$  = conjunto de no rostros

-  $F_0 = 1$ ;  $D_0$ , Inicialización del rango de falsos positivos y el rango de detección de la primera capa, respectivamente

-  $i = 0$

**mientras**  $F_i > F_{target}$  **hacer**

+  $i = i + 1$

+  $n_i = 0$ ;  $F_i = F_{i-1}$

**mientras**  $F_i > f * F_{i-1}$  **hacer**

\*  $n_i = n_i + 1$

\* Usar  $P$  y  $N$  para entrenar un clasificador con  $n_i$  características usando boosting

\* Evaluar el clasificador en cascada actual sobre el conjunto de validación para determinar  $F_i$  y  $D_i$

\* Disminuir el umbral para el clasificador  $i$  hasta que el actual clasificador en cascada tenga un rango de detección de al menos  $d$  x  $D_{i-1}$

**fin mientras**

+  $N = \emptyset$

**si**  $F_i > F_{target}$  **entonces**

Evaluar el actual clasificador en cascada sobre el conjunto de las no caras y colocar en  $N$  todas las falsas detecciones.

**fin si**

**fin mientras**

---

### 3. Planteamiento del problema

Detectar rostros implica la construcción de un modelo computacional fundamentado en conocimientos sobre tratamiento digital de imágenes e inteligencia artificial, que busque dentro de un espacio de características información relevante que permita separar eficazmente los rostros de cualquier otra imagen.

Encontrar las características adecuadas implica realizar un análisis estructural dentro de un conjunto de  $k$  ejemplares, que se obtienen a partir de la variación de cinco modelos y se denominan características de Haar. El número de características de Haar depende del tamaño de las imágenes de la base de datos de entrenamiento; si las imágenes de entrenamiento son de  $19 \times 19$  píxeles se contaría con un  $k \approx 45000$  características de Haar; en cambio si es de  $24 \times 24$  píxeles,  $k \approx 160000$  características de Haar. El valor de la constante  $k$  se calcula estimando el número de posibilidades en las que se puede organizar los cinco modelos de referencia (Figura 3) dentro de una región de tamaño equivalente a las imágenes de entrenamiento, cumpliendo la restricción de un tamaño mínimo para la zona de útil de 8 píxeles. A partir de cada característica se pueden generar  $n$  clasificadores débiles, donde  $n$  equivale al número de muestras de entrenamiento, por tanto, el número de clasificadores débiles sobre los que se buscará una mejor correspondencia con el caso de estudio crece en un factor de  $O(k * n)$ . Se requiere organización adecuada de los datos pues el volumen de información crece significativamente y la solución del problema podría presentar

limitaciones de hardware en el entrenamiento.

A partir de las condiciones planteadas, el desarrollo del trabajo pretende consolidar una arquitectura formada por clasificadores débiles que en conjunto forman una estructura multinivel, donde cada nivel presenta diferente complejidad.

## 4. Justificación

Resolver el problema de detección requiere de la realización de un entrenamiento supervisado de un detector en cascada, lo cual presenta un costo computacional alto que depende de la cantidad de muestras de entrenamiento y las características utilizadas.

Entrenar el detector de rostros mediante el framework de detección de objetos puede tardar varias semanas debido al gran número de operaciones que hay que realizar para determinar el orden y peso de los clasificadores débiles en la arquitectura [13]. Es por tal razón que una implementación minuciosa de estos algoritmos es un problema de ingeniería ya que la revisión y análisis de la convergencia del modelo es fundamental.

## 5. Planteamiento de la solución

La implementación del detector de rostros en posición frontal mediante un algoritmo de boosting se realizó en el lenguaje de programación JAVA mediante la utilización del Entorno de Desarrollo (IDE) Eclipse que facilita la depuración y ejecución de aplicaciones en este lenguaje de programación.

La programación se realizó sobre la plataforma JAVA Standard Edition 6 y se utilizó adicionalmente la librería ImageJ que facilita el acceso de forma transparente a los niveles de intensidad de los pixeles en imágenes de diversos formatos.

A partir de estas herramientas se realizó la implementación de los algoritmos necesarios para el entrenamiento como para la evaluación del detector de rostros, que no presentan dependencia de técnicas de aprendizaje supervisado como redes neuronales, máquinas de soporte vectorial o alguna otra técnica adicional.

### 5.1. Base de Datos

La creación de un detector de rostros implica la búsqueda o construcción de una base de datos, sobre la que se deben aplicar las características de Haar necesarias para el procesamiento de los datos, que posteriormente se utilizan en la construcción del modelo matemático mediante la aplicación de los algoritmos de entrenamiento.

Para el entrenamiento y prueba de la estructura del detector de rostros se requiere de una base de datos de imágenes con rostros que en lo posible se encuentren correctamente recortados o al menos centrados reduciendo de esta manera errores ocasionados por información no relevante para la solución del problema.

La base de datos seleccionada es la CBCL FACE DETECTION #1 [1] que dispone para el entrenamiento de 2429 rostros y 4548 imágenes variadas; y para prueba 472 rostros y 23573 imágenes (Figuras 8 y 9).



**Figura 8:** Ejemplos de rostros de la base de datos CBCL FACE DETECTION #1



**Figura 9:** Ejemplos de no rostros de la base de datos CBCL FACE DETECTION #1

## 5.2. Características de Haar

Las características de Haar para la construcción del modelo computacional deben estar dotadas de la capacidad de escalado en espacio, debido a que la búsqueda de rostros se realiza sobre imágenes que superan el tamaño de referencia de las características (24 píxeles o 19 píxeles dependiendo de la base de datos utilizada para el entrenamiento del detector). Entre las alternativas disponibles que se presentan

están el escalado de la característica o el escalado de la imagen a procesar.

El escalado de la imagen a procesar requiere de aplicación de técnicas de interpolación de píxeles que permitan una reducción apropiada de los datos, lo cual implica pérdida de información. Para evitar las implicaciones derivadas de la reducción de la imagen se procede a la amplificación de las características que computacionalmente no implica esfuerzo adicional.

Para la amplificación de las características se requiere de la generalización de las características en función del tamaño, que se puede obtener de la división de las coordenadas  $x$  e  $y$  en el tamaño de referencia utilizado para la implementación (Ver Cuadro 4). Este proceso brinda una localización porcentual que fácilmente permite la operación de la característica sobre la imagen a diferentes resoluciones.

### 5.3. Normalización de los datos



Las imágenes que se emplean para el entrenamiento como para las pruebas del detector de rostros requieren de una normalización de su varianza, con el fin de disminuir el efecto ocasionado por diferentes condiciones de iluminación. Para esto se debe aplicar por cada píxel de la imagen la función  $f(x)$

$$f(x) = \frac{x - \bar{x}}{\sigma} \quad (11)$$

donde  $x$  es el valor de intensidad del píxel,  $\bar{x}$  es la media de las intensidades de los píxeles de la imagen y  $\sigma$  su desviación estándar, definida como:

$$\sigma = \sqrt{\left[ \frac{n}{n-1} \right] \left[ \frac{\sum x^2}{n} - \bar{x}^2 \right]} \quad (12)$$

La realización de estas operaciones para una imagen no implica gran costo computacional. A pesar de esto se debe tener en cuenta que en un ambiente real se re-

Característica	x	y	*	x'	y'
	8	3	1	0.4211	0.1579
	13	3	-2	0.6842	0.1579
	16	3	1	0.8421	0.1579
	8	6	-2	0.4211	0.3158
	13	6	4	0.6842	0.3158
	16	6	-2	0.8421	0.3158
	8	11	1	0.4211	0.5789
	13	11	-2	0.6842	0.5789
	16	11	1	0.8421	0.5789
	9	0	1	0.4737	0
	12	0	-2	0.6316	0
	15	0	2	0.7895	0
	18	0	-1	0.9474	0
	9	18	-1	0.4737	0.9474
	12	18	2	0.6316	0.9474
	15	18	-2	0.7895	0.9474
	18	18	1	0.9474	0.9474

**Cuadro 4:** Generalización de las características a partir de un tamaño de referencia de 19 píxeles, donde  $x'$  y  $y'$  son la generalización de  $x$  e  $y$  respectivamente

quiere recorrer una imagen de resolución considerable obteniendo gran cantidad de subimágenes a diferentes resoluciones y localizaciones, que deben ser analizadas por el detector de rostros. El procesamiento de estas muestras implica su normalización, que puede aumentar significativamente el tiempo de respuesta del detector ya que el número de muestras crece en función de la resolución.

Aprovechando las propiedades de la imagen integral y el uso de las características de Haar que requieren del acceso a ciertos puntos de la imagen integral, y teniendo en cuenta que estos puntos deben haber pasado por un preproceso, es decir, su correspondiente normalización y posteriormente su sumarización, se puede acceder eficientemente a esta información de la siguiente manera:

Dada una imagen  $i$ , se obtiene su imagen integral  $ii$  mediante la ecuación [1] y su imagen integral para los píxeles al cuadrado  $ii2$ , donde

$$ii2(x_k, y_m) = \sum_{x=1}^k \sum_{y=1}^m i(x, y)^2 \quad (13)$$

A partir de esto y utilizando el mismo principio de las regiones de la característica, la evaluación de un punto  $P_{x,y}$  (Ver figura 10) relativo a la subimagen en una matriz sumarizada  $M$ , (donde  $M$  puede ser obtenida con  $ii$  o  $ii2$ ) equivale a

$$g(M, P_{x,y}) = M(P_{0,0}) + M(P_{x,y}) - (M(P_{0,y}) + M(P_{x,0})) \quad (14)$$

y la normalización de la matriz sumarizada  $gn$  a

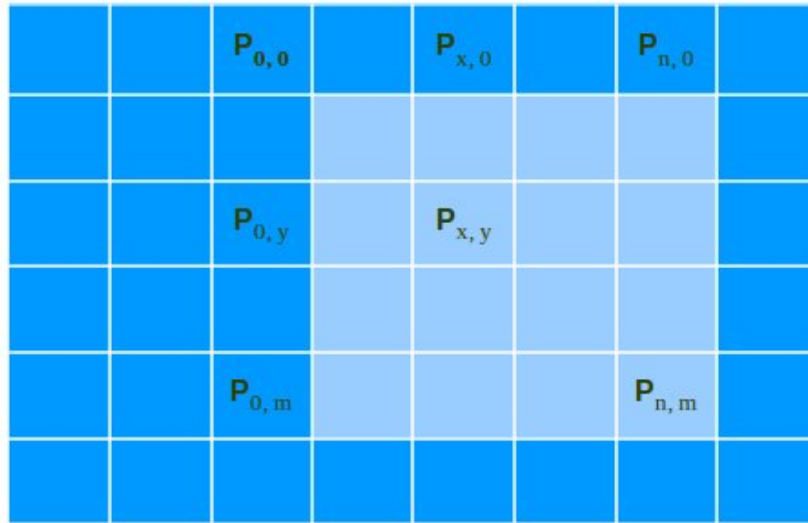
$$gn(P_{x,y}) = \frac{g(ii, P_{x,y}) - k * media}{\sigma} \quad (15)$$

donde  $k = x * y$ , el número de píxeles desde el extremo superior izquierdo hasta  $P_{x,y}$

$$media = \frac{g(ii, P_{n,m})}{N} \quad (16)$$

donde  $N = n * m$ , el número de pixeles que abarca la subimagen o muestra

$$\sigma = \sqrt{\left(\frac{N}{N-1}\right) \left(\frac{[g(ii2, P_{n,m})]^2}{N} - media^2\right)} \quad (17)$$



**Figura 10:** Imagen de referencia para normalización de un punto  $P_{x,y}$  de una subimagen

Para normalizar un punto  $P_{x,y}$  se requiere extraer  $k$  veces la media debido a que esta normalización se realiza sobre una imagen sumariada, donde para un punto con coordenadas  $(x, y)$  se han sumado  $k - 1$  puntos previamente.

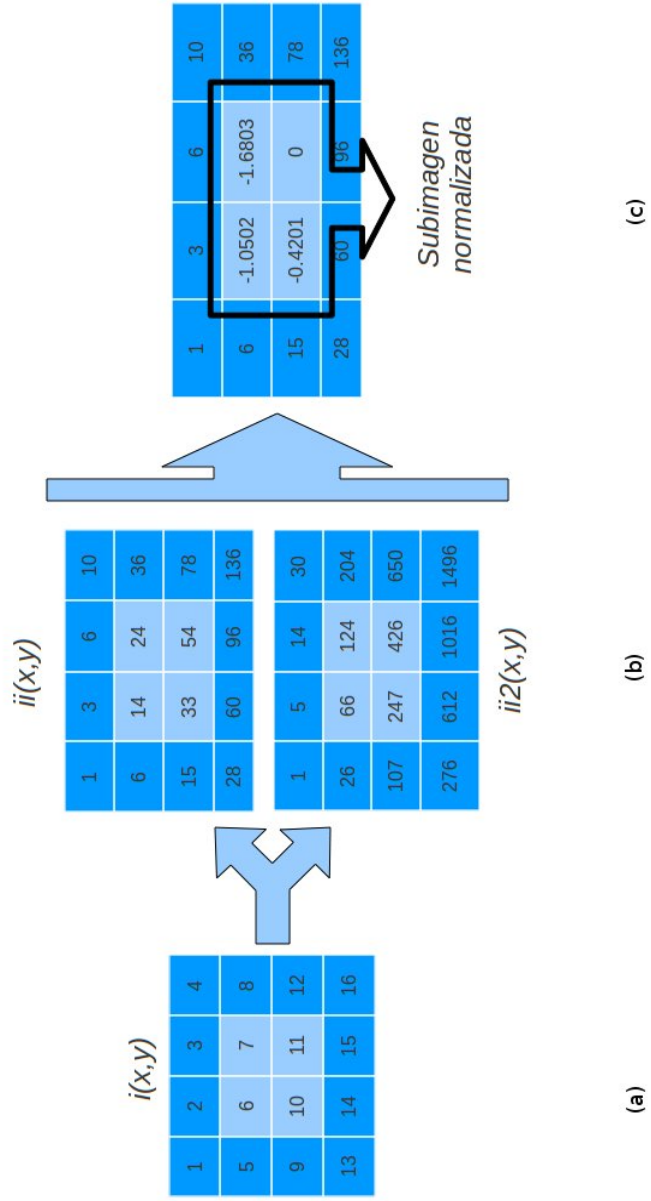


Figura 11: Ejemplo de normalización de una subimagen

La figura 11 muestra el proceso de normalización de una subimagen o muestra dentro de una imagen, en la que dada una imagen  $i(x, y)$  (figura 11 (a)), se obtiene su imagen integral  $ii(x, y)$  y su imagen integral para los pixeles de la imagen al cuadrado  $ii2(x, y)$  (figura 11 (b)). A partir de estas dos matrices y mediante la ecuación 16, se normalizan los pixeles de la subimagen o muestra (figura 11 (c)).

## 5.4. Entrenamiento del Detector de Rostros

Para entrenar el detector de rostros se debe disponer de una base de datos debidamente etiquetada, donde los rostros presenten una marca específica, que para el caso de los rostros fue el valor 1 y para los no rostros el valor de 0; adicional a esto se debe disponer de dos conjuntos de datos, uno para prueba y uno para entrenamiento.

El entrenamiento del detector de rostros requiere de la aplicación de los conceptos mencionados previamente en la sección 3.2, que permitan la creación de la estructura del detector de rostros. El entrenamiento comienza por la ejecución del Algoritmo de Entrenamiento del clasificador en cascada o detector en cascada (Algoritmo 2), que requiere de un porcentaje máximo de falsos positivos  $f$  y un porcentaje mínimo aceptable de detección por capa  $d$ , así como un máximo aceptable de falsos positivos tolerable por el detector final  $f_{target}$ .

Estos parámetros libres se establecieron a partir del análisis de convergencia del algoritmo y a partir de los requerimientos de precisión que se requerían para el detector de rostros. Estos parámetros se establecieron de la siguiente manera

Parámetro	# Valor	Rango
$f$	0.55	(0 - 1)
$d$	0.95	(0 - 1)
$f_{target}$	0.01	(0 - 1)

**Cuadro 5:** Parámetros utilizados para el entrenamiento del detector de rostros

Para el caso de los falsos positivos aceptables por capa  $f$ , aunque se considere un valor aparentemente alto del 55 %, este disminuye significativamente según lo propuesto en Viola & Jones [12], donde el rango de falsos positivos de la capa  $j$  equivale a, ecuacion 18.

$$f_j = \prod_{i=1}^j f_i \quad (18)$$

donde  $f_j$  es el rango de falsos positivos de la capa  $j$ , que se obtiene de la productoria de los  $f_i$  de las capas anteriores.

Si se revisa el Algoritmo de Entrenamiento del clasificador en cascada (Algoritmo 2), en cada iteración se construye un clasificador por Boosting de diferente complejidad mediante el Algoritmo de entrenamiento por boosting (Algoritmo 1), donde la complejidad aumenta en función del número de clasificadores débiles que lo conformen.

La selección de los clasificadores débiles es una tarea exhaustiva debido a que se debe escoger por iteración el mejor clasificador débil dentro de un conjunto de  $k * N$  elementos, donde  $k$  es el número de características de Haar y  $N$  el número de ejemplos de entrenamiento.

Analizando el algoritmo de entrenamiento por *boosting* se concluye que por cada iteración se requiere procesar el mismo volúmen de información, es decir, evaluar todos los clasificadores sobre el conjunto de ejemplos de entrenamiento; y entre capas del clasificador en cascada un subconjunto de la capa anterior debido a que a medida que se crean las capas, el conjunto de los no rostros disminuye ya que la capacidad de identificación del detector va mejorando. Partiendo de los supuestos mencionados se procede a crear una base de datos en un archivo secuencial que contiene toda la información referente a las características por imagen de entrenamiento, de fácil acceso y que reduce el tiempo de entrenamiento significativamente de 15 minutos por iteración a 3 minutos aproximadamente. Para  $k * n = 314055000$  clasificadores débiles, se reduce el tiempo de entrenamiento 5 veces aproximadamente por iteración.

La selección del mejor clasificador débil mediante la minimización del error requiere de las siguientes consideraciones: El proceso de evaluación del mejor clasificador débil se realiza mediante un único acceso al conjunto de datos, este acceso se hace por característica obteniendo todos los valores asociados a ésta sobre todo el conjunto de imágenes de entrenamiento; se ordenan de manera ascendente y se comienzan a recorrer llevando el registro de la suma de todos los pesos de las imágenes positivas  $T^+$ , la suma de todos los pesos de las imágenes negativas  $T^-$ , la suma de los pesos de las imágenes positivas sin incluir el actual  $S^+$  y la suma de los pesos de las imágenes negativas sin incluir el actual  $S^-$ . Con esto se obtiene el menor error  $e$  por característica con la ecuación 19

$$e = \min(S^+ + (T^- - S^-), S^- + (T^+ - S^+)) \quad (19)$$

Luego de iterar sobre todas las características, se selecciona la que presente el mínimo error  $e$ , y los parámetros del clasificador débil (ecuación [10]) quedan defi-

nidos en este proceso iterativo tal como muestra el algoritmo de búsqueda del mejor clasificador débil (Algoritmo 3).

---

**Algoritmo 3** Algoritmo de búsqueda del mejor clasificador débil

---

**Entrada:** Conjunto de datos con la evaluación de todos los ejemplos de entrenamiento sobre las características,  $R$  de tamaño  $k * n$

**Entrada:** Pesos  $W$  con un valor asociado a cada ejemplo que establece el grado de importancia del ejemplo de acuerdo a su comportamiento en el proceso de entrenamiento del clasificador por *boosting* (Ver Algoritmo 1).

**Entrada:** Etiquetas de los ejemplos de entrenamiento  $Y$

- Hallamos  $T^+$  y  $T^-$ , la suma de todos los ejemplos positivos y negativos respectivamente
- $S^+ = 0, S^- = 0$
- $e_f = \infty$

**para**  $i = 1, \dots, k$  **hacer**

- Obtenemos  $V$  equivalente a todos valores de los ejemplos de entrenamiento para la característica  $i$
- Ordenamos de menor a mayor  $V$

**para**  $j = 1, \dots, n$  **hacer**

$e = \min(S^+ + (T^- - S^-), S^- + (T^+ - S^+))$

**si**  $Y_j = 1$  **entonces**

$S^+ = S^+ + V_j$

**si no**

$S^- = S^- + V_j$

**fin si**

**si**  $e < e_f$  **entonces**

$e_f = e$

$\theta = V_j$

$f = j$

**si**  $e = S^+ + (T^- - S^-)$  **entonces**

$p = -1$

**si no**

$p = 1$

**fin si**

**fin si**

**fin para**

**devolver**  $p$ , la polaridad del clasificador débil

**devolver**  $\theta$ , el umbral del clasificador débil

**devolver**  $f$ , índice de la característica asociada al clasificador débil

---

Con el entrenamiento se construye un clasificador de diferentes niveles de complejidad y aunque se aplica la metodología del Algoritmo 2, el número de clasifica-

dores en los primeros niveles del clasificador se define de forma manual por recomendación de Viola y Jones [13], puesto que el entrenamiento presenta un tiempo de cómputo excesivo, equivalente a varias semanas.

El número de clasificadores en los primeros niveles del detector de rostros quedaron definidos según la tabla 6.

Nivel	# Características
1	2
2	10
3	25
4	25
5	50
6	50
7	50
8	75
9	75
10	75

**Cuadro 6:** Número de clasificadores definidos manualmente

Para los niveles siguientes partiendo de aclaraciones de implementaciones anteriores [13], se agregan por iteración 25 clasificadores débiles acelerando de esta manera el entrenamiento del detector final que presenta una arquitectura de 38 niveles y que en un Intel Quad Core 2.33 GHz tarda aproximadamente 12 días.

## 6. Resultados

A partir del entrenamiento del detector de rostros y mediante la utilización de la base de datos CBCL FACE DETECTION #1, que dispone de un conjunto de entrenamiento de 2429 rostros y 4548 imágenes variadas, el detector de rostros genera los resultados del entrenamiento expuestos en las gráficas 12, 13 , 14 y 15 sobre el conjunto de pruebas de la base de datos formado por 472 rostros y 23573 imágenes.

Las pruebas aplicadas a los resultados generados por el detector de rostros fueron la sensibilidad, la especificidad, el valor predictivo positivo VPP y el valor predictivo negativo VPN, que miden la capacidad de clasificación del modelo, donde las mediciones se realizan en las capas del clasificador durante el entrenamiento, mostrando la evolución del clasificador.

La Sensibilidad es la probabilidad de clasificar correctamente los rostros, es decir, la probabilidad de que para un rostro, se obtenga en la prueba un resultado etiquetado como rostro. La sensibilidad es, por lo tanto, la capacidad del clasificador para detectar los rostros.

$$sensibilidad = \frac{VP}{VP + FN} \quad (20)$$

donde  $VP$  son los verdaderos positivos y  $FN$  son los falsos negativos

La especificidad es la probabilidad de clasificar correctamente un no rostro, es decir,

la probabilidad de que para un no rostro se obtenga un resultado etiquetado como no rostro. En otras palabras, se puede definir la especificidad como la capacidad para detectar a los no rostros.

$$\text{especificidad} = \frac{VN}{VN + FP} \quad (21)$$

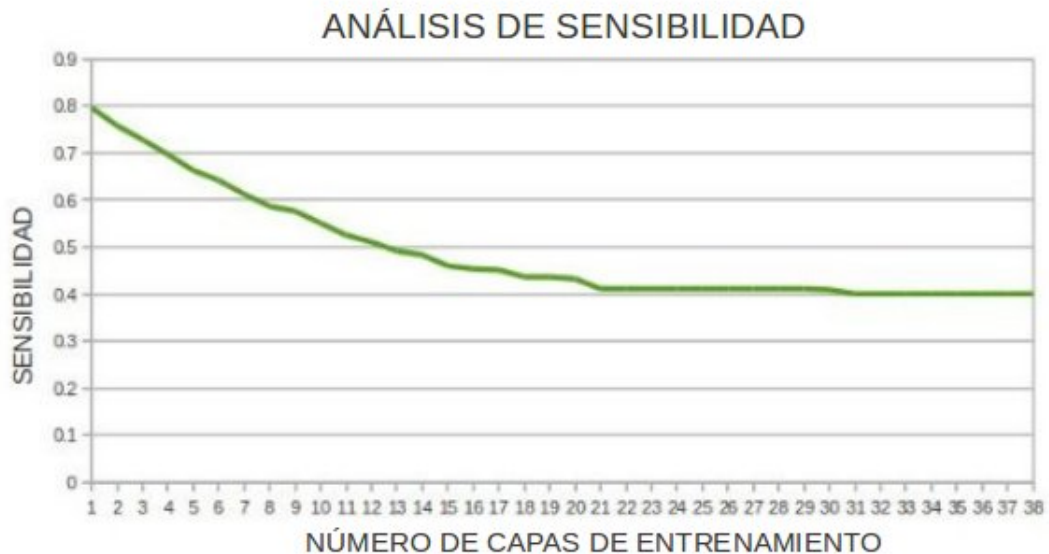
donde  $VN$  son los verdaderos negativos y  $FP$  son los falsos positivos

El Valor Predictivo Positivo  $VPP$  es la probabilidad de que una imagen sea rostro si se obtiene un resultado etiquetado como rostro en la prueba. El valor predictivo positivo puede estimarse, por tanto, a partir de la proporción de rostros con un resultado positivo en la prueba que finalmente resultaron siendo rostros.

$$VPP = \frac{VP}{VP + FP} \quad (22)$$

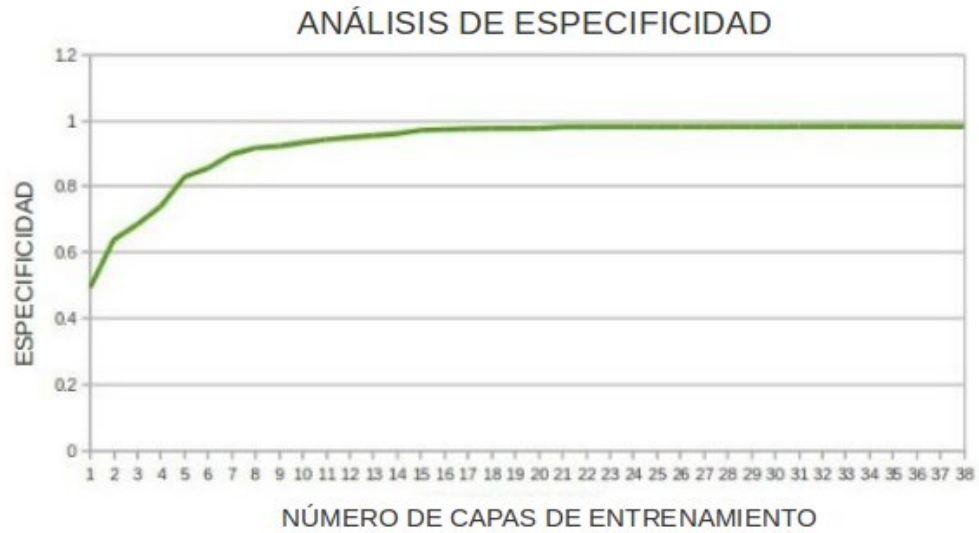
El Valor Predictivo Negativo  $VPN$  es la probabilidad de que una imagen no sea rostro si se obtiene un resultado etiquetado como no rostro en la prueba.

$$VPN = \frac{VN}{FN + VN} \quad (23)$$



**Figura 12:** Análisis de Sensibilidad por capa del Detector en cascada de rostros con la base de datos CBCL FACE DETECTION #1

En la figura 12 se aprecia que a medida que avanza el entrenamiento del clasificador, la sensibilidad del detector va disminuyendo, y aunque esto no es conveniente puesto que el objetivo del clasificador es la búsqueda de los rostros, este comportamiento se justifica con la filosofía de entrenamiento del modelo utilizado, ya que el objetivo final del clasificador es identificar explícitamente lo que no es rostro, quedando como resultado de este rechazo, las detecciones de los rostros.



**Figura 13:** Análisis de Especificidad por capa del Detector en cascada de rostros con la base de datos CBCL FACE DETECTION #1

En la figura 13 se aprecia que a medida que avanza el entrenamiento del clasificador, la especificidad aumenta hasta 99%, que muestra que el clasificador tiene la capacidad de rechazar en su mayoría lo que no se considera rostro.



**Figura 14:** Análisis del Valor Predictivo Positivo VPP por capa del Detector en cascada de rostros con la base de datos CBCL FACE DETECTION #1



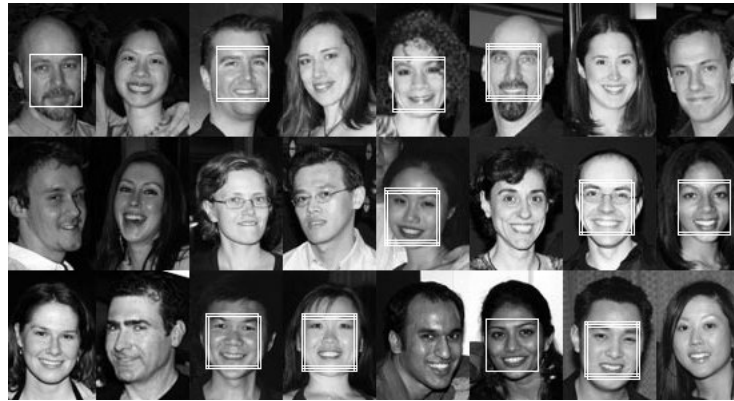
**Figura 15:** Análisis del Valor Predictivo Negativo VPN por capa del Detector en cascada de rostros con la base de datos CBCL FACE DETECTION #1

Los resultados del clasificador se muestran por capa con el propósito de entender el papel que cumple la estructura multinivel que conforma el detector de rostros, donde los valores obtenidos en la última capa (capa 38) equivalen a los resultados finales de evaluación del clasificador utilizando toda su estructura.

Adicionalmente se realizaron pruebas con imágenes que contienen rostros que presentan diferentes condiciones de iluminación, posición y tamaño, para lo cual se hizo un muestreo por imagen que permitiera la búsqueda de los rostros. Este muestreo consistió en tomar subimágenes de diversos tamaños de la imagen original, denominadas muestras con el propósito de verificar si representaban o no un rostro.

Este muestreo se debe a que el detector se encarga únicamente de verificar si una imagen dada es o no un rostro, independientemente del tamaño de la imagen, es decir, el detector no se encarga de realizar una búsqueda de rostros dada una imagen formada por rostros y no rostros.

A partir de estas consideraciones se realizaron diversas pruebas para medir la capacidad de detección del clasificador de rostros, las cuales arrojaron los resultados expuestos en los cuadros 7, 8, 9, 10 y 11.

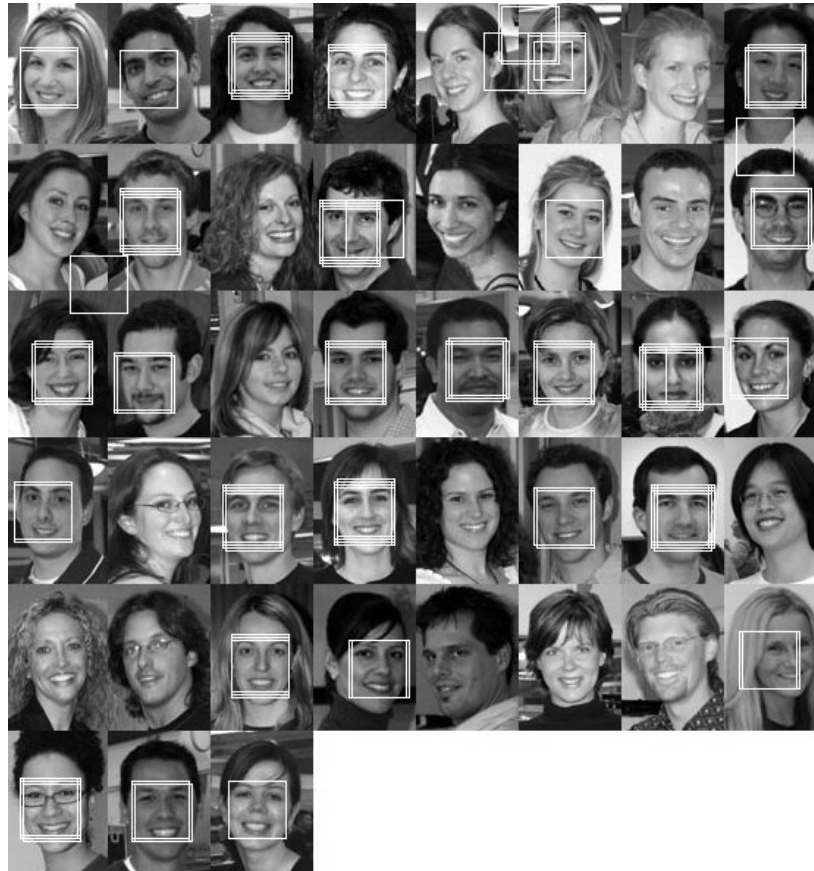


**Figura 16:** Prueba #1

Para la imagen de la figura 16 se obtuvieron los siguientes resultados

Verdaderos Positivos (VP)	11
Falsos Positivos (FP)	0
Verdaderos Negativos (VN)	373971
Falsos Negativos (FN)	13
Sensibilidad	0.458333
Especificidad	1
Valor Predictivo Positivo (VPP)	1
Valor Predictivo Negativo (VPN)	0.999965
Tiempo promedio evaluación por Muestra	0.1212 milisegundos

**Cuadro 7:** Análisis de Resultados de la Prueba # 1



**Figura 17: Prueba #2**

Para la imagen de la figura 17 se obtuvieron los siguientes resultados

Verdaderos Positivos (VP)	28
Falsos Positivos (FP)	7
Verdaderos Negativos (VN)	777445
Falsos Negativos (FN)	15
Sensibilidad	0.651163
Especificidad	0.999991
Valor Predictivo Positivo (VPP)	0.8
Valor Predictivo Negativo (VPN)	0.999981
Tiempo promedio evaluación por Muestra	0.1693 milisegundos

**Cuadro 8:** Análisis de Resultados de la Prueba # 2



**Figura 18:** Prueba #3

Para la imagen de la figura 18 se obtuvieron los siguientes resultados

Verdaderos Positivos (VP)	1
Falsos Positivos (FP)	2
Verdaderos Negativos (VN)	576925
Falsos Negativos (FN)	0
Sensibilidad	1
Especificidad	0.999997
Valor Predictivo Positivo (VPP)	0.333333
Valor Predictivo Negativo (VPN)	1
Tiempo promedio evaluación por Muestra	0.0927 milisegundos

**Cuadro 9:** Análisis de Resultados de la Prueba # 3



**Figura 19:** Prueba #4

Para la imagen de la figura 19 se obtuvieron los siguientes resultados

Verdaderos Positivos (VP)	1
Falsos Positivos (FP)	1
Verdaderos Negativos (VN)	70941
Falsos Negativos (FN)	0
Sensibilidad	1
Especificidad	0.999986
Valor Predictivo Positivo (VPP)	0.5
Valor Predictivo Negativo (VPN)	1
Tiempo promedio evaluación por Muestra	0.2008 milisegundos

**Cuadro 10:** Análisis de Resultados de la Prueba # 4



**Figura 20:** Prueba #5

Para la imagen de la figura 20 se obtuvieron los siguientes resultados

Verdaderos Positivos (VP)	3
Falsos Positivos (FP)	1
Verdaderos Negativos (VN)	255105
Falsos Negativos (FN)	4
Sensibilidad	0.428571
Especificidad	0.999996
Valor Predictivo Positivo (VPP)	0.75
Valor Predictivo Negativo (VPN)	0.999984
Tiempo promedio evaluación por Muestra	0.1441 milisegundos

**Cuadro 11:** Análisis de Resultados de la Prueba # 5

## 7. Discusión de los Resultados

A partir de los resultados obtenidos con el conjunto de ejemplos de prueba de la base de datos CBCL FACEDETECTION #1, se confirma que el detector en su último nivel alcanza una sensibilidad de 0.400424, con un valor predictivo positivo VPP de 0.317114, lo cuál indica que el clasificador no garantiza el hallazgo de rostros dentro de un conjunto de datos, pero garantiza con una especificidad de 0.982735 y un valor predictivo negativo VPN de 0.987931 que rechaza con gran precisión los imágenes que efectivamente no pertenecen al conjunto de los rostros.

La razón por la cual el detector no discrimina eficazmente los conjuntos de datos, en especial los rostros, se debe a que los datos de la CBCL FACEDETECTION # 1 no son en su totalidad representativos para la solución del problema, pero suficientes para alcanzar un generalización aceptable.

Además, con el fin de verificar el funcionamiento del detector en un ambiente diferente al de la base de datos CBCL FACEDETECTION #1, se realizaron pruebas adicionales para observar el grado de generalización del clasificador. Estos resultados expuestos en los cuadros del 7 al 11 y resumidos en el cuadro 12 confirman la capacidad del detector de rechazar eficazmente las imágenes que no son rostros con una especificidad del 0.9998 y un valor predictivo negativo de 0.9998. Adicionalmente se confirma que el detector tiene una capacidad considerable para detectar rostros

	Media	Varianza	Desviación Estandar
Sensibilidad	0.7076	0.0785	0.2803
Especificidad	0.9998	0.0000	0.0000
Valor Predictivo Positivo	0.6767	0.0686	0.2618
Valor Predictivo Negativo	0.9998	0.0000	0.0000
Tiempo promedio Muestra (ms)	0.1456	0.0018	0.0419

**Cuadro 12:** Resumen de los Resultados de las Pruebas

en ambientes heterógeneos con una sensibilidad de  $0.7076 \pm 0.2803$  y un valor predictivo positivo de  $0.6767 \pm 0.2618$ .

Como valor agregado se puede apreciar que el detector tiene un tiempo de respuesta rápido por imagen analizada de  $0.1456 \pm 0.0419$  milisegundos, lo cual indica que el detector aproximadamente verifica 6868.13 imágenes por segundo, donde una imagen es un ejemplo a verificar por el clasificador que puede variar en tamaño ya que el clasificador tiene la capacidad de analizar muestras a diferentes resoluciones indicando si son o no un rostro.

# 8. Conclusiones y Recomendaciones

## 8.1. Conclusiones

La implementación de un detector de rostros en posición frontal mediante el framework de detección rápida de objetos es una buena alternativa para la solución de problemas que impliquen la búsqueda de patrones dentro de un conjunto de datos.

Se pudo observar que debido a su estructura, el detector de rostros en cascada efectivamente acelera el proceso de detección, ya que con un menor número de operaciones rechaza la mayor cantidad de imágenes que no son consideradas rostros, discriminando con mayor precisión únicamente un pequeño número de imágenes donde se tiene alta probabilidad de encontrar un rostro.

La base de datos CBCL FACEDETECTION #1 a pesar de brindar una generalización considerable al detector, no es la mejor alternativa para el entrenamiento de un detector de rostros en posición frontal; esto se debe a que las imágenes de entrenamiento y prueba no presentan condiciones óptimas que permitan la generación de un clasificador de mayor precisión, ya que en muchos casos los rostros eran poco legibles.

El uso de la imagen integral optimiza el tiempo de computo requerido para la eva-

luación de las características de Haar pues el número de operaciones necesarias para evaluar los cinco modelos de referencia es constante e independiente del tamaño de la imagen a evaluar.

Conocer el modelo matemático sobre el que se fundamenta el framework de detección rápida de objetos permite su implementación en hardware dedicado como los DSP(Digital Signal Processors), ya que las librerías disponibles actualmente para detección de rostros no son compatibles con estas arquitecturas.

## 8.2. Recomendaciones

A partir de los resultados obtenidos del estudio se recomienda hacer verificación de los algoritmos de entrenamiento con otras bases de datos que permitan realizar una mejor generalización de los casos de estudio vinculados al problema.

Si se estudia en detalle los algoritmos de entrenamiento se puede realizar una optimización en el tiempo de entrenamiento mediante paralelización del algoritmo de búsqueda del mejor clasificador débil, el cual consume el mayor tiempo en el periodo de entrenamiento del detector en cascada de rostros.

Un trabajo complementario al actual sería la implementación de una técnica de búsqueda de rostros sobre una imagen de dimensiones considerables, pues el recorrido sobre imágenes no era una finalidad del trabajo. Esta proposición nace a partir del gran número de muestras que se deben tomar de una imagen para detectar los rostros que se encuentran inmersos.

Una implementación del algoritmo sobre un DSP permitirá la detección de rostros

en tiempo real, ya que presenta un arquitectura de hardware dedicada con canales de entrada y salida de video, que permiten captar las imágenes directamente de la fuente y su correspondiente retrasmisión con los resultados obtenidos.

# Bibliografía

- [1] Cbcl face database no. 1. *MIT Center For Biological and Computation Learning*, <http://www.ai.mit.edu/projects/cbcl>, 2000.
- [2] Vladimir Vapnik Corinna Cortes. Support-vector networks. *Machine Learning*, 20:273–297, 1995.
- [3] Shape Constraints David, David Cristinacce, and Tim Cootes. Facial feature detection using adaboost with shape constraints. pages 231–240, 2003.
- [4] Y. Freund and Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *In Computational Learning Theory: Eurocolt 95*, Springer-Verlag, page 23–37, 1995.
- [5] Yoav Freund. An adaptive version of the boost by majority algorithm. pages 102–113, 2000.
- [6] Duy-Dinh Le and Shin'ichi Satoh. Feature selection by adaboost for efficient svm-based face detection. *Information Technology Letters*, 3:183–186, 2004.
- [7] J. Ng and S.G. Gong. Composite support vector machines for detection of faces across views and pose estimation. 20(5-6):359–368, April 2002.
- [8] M. Papageorgiou, C. Oren and T. Poggio. A general framework for object detection. *In International Conference on Computer Vision*, 1998.

- [9] A. Saffari, M. Godec, T. Pock, C. Leistner, and H. Bischof. Online multi-class lpboost. pages 3570–3577, june 2010.
- [10] C.C. Tsai, W.C. Cheng, J.S. Taur, and C.W. Tao. Face detection using eigenface and neural network. 5:4343–4347, Oct. 2006.
- [11] Paul Viola and Michael J. Jones. Robust real-time object detection. *Second International Workshop on Statistical and Computational Theories of Vision - Modeling, Learning, Computing, and Sampling*, 2001.
- [12] Paul Viola and Michael J. Jones. Rapid object detection using a boosted cascade of simple features. 2004.
- [13] Paul Viola and Michael J. Jones. Robust real-time face detection. *Int. J. Comput. Vision*, 57(2):137–154, 2004.
- [14] Manfred K. Warmuth, Karen A. Glocer, and S. V. N. Vishwanathan. Entropy regularized lpboost. 2008.
- [15] Phillip Ian Wilson and John Fernandez. Facial feature detection using haar classifiers. *J. Comput. Small Coll.*, 21(4):127–133, 2006.
- [16] M.-H. Yang and N. Ahuja. Face detection and hand gesture recognition for human-computer interaction. *Kluwer Academic Publishers*, 2001.
- [17] Mian Zhou and Hong Wei. Face verification using gaborwavelets and adaboost. pages 404–407, 2006.