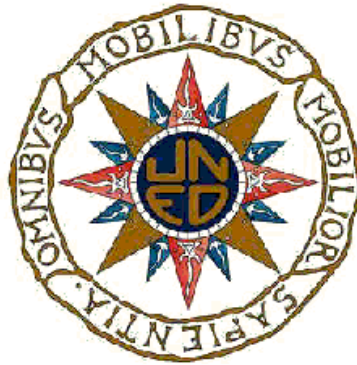


# **MÁSTER UNIVERSITARIO**

## **DE INVESTIGACIÓN EN INGENIERÍA DEL SOFTWARE Y SISTEMAS INFORMÁTICOS**



### **TRABAJO FIN DE MÁSTER**

#### **DESARROLLO Y EVALUACIÓN DE TÉCNICAS DE VISIÓN ARTIFICIAL APLICADAS A UN PROBLEMA DE RECONOCIMIENTO FACIAL**

**AUTOR: ALBERTO BRAVO PÉREZ**

**DIRECTOR: CARLOS CERRADA SOMOLINOS**

Mayo 2018



Trabajo Fin de Máster

**Desarrollo y evaluación de técnicas de visión artificial aplicadas a un  
problema de reconocimiento facial**

AUTOR:

Alberto Bravo Pérez

DIRECTOR:

Carlos Cerrada Somolinos

La defensa del presente Trabajo Fin de Máster se realizó el día XX de XXXX de  
XXXX y fue evaluada por el siguiente tribunal:

PRESIDENTE:

SECRETARIO:

VOCAL:

---



# **DECLARACIÓN JURADA DE AUTORÍA DEL TRABAJO CIENTÍFICO, PARA LA DEFENSA DEL TRABAJO FIN DE MASTER**

Fecha: 05/06/2018

Quién suscribe:

Autor(a): Alberto Bravo Pérez

D.N.I./N.I.E./Pasaporte.: 03895185C

Hace constar que es el autor(a) del trabajo:

**Desarrollo y evaluación de técnicas de visión artificial aplicadas a  
un problema de reconocimiento facial**

En tal sentido, manifiesto la originalidad de la conceptualización del trabajo, interpretación de datos y la elaboración de las conclusiones, dejando establecido que aquellos aportes intelectuales de otros autores, se han referenciado debidamente en el texto de dicho trabajo.

## **DECLARACIÓN:**

- ✓ Garantizo que el trabajo que remito es un documento original y no ha sido publicado, total ni parcialmente por otros autores, en soporte papel ni en formato digital.
- ✓ Certifico que he contribuido directamente al contenido intelectual de este manuscrito, a la génesis y análisis de sus datos, por lo cual estoy en condiciones de hacerme públicamente responsable de él.
- ✓ No he incurrido en fraude científico, plagio o vicios de autoría; en caso contrario, aceptaré las medidas disciplinarias sancionadoras que correspondan.

Fdo.







IMPRESO TFdMo5\_AUTOR  
AUTORIZACION DE PUBLICACION  
CON FINES ACADEMICOS



Impreso TFdMo5\_Autor. Autorización de publicación  
y difusión del TFdM para fines académicos

## Autorización

Autorizo/amos a la Universidad Nacional de Educación a Distancia a difundir y utilizar, con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la memoria de este Trabajo Fin de Máster, como el código, la documentación y/o el prototipo desarrollado.

Firma del/los Autor/es

Juan del Rosal, 16  
28040, Madrid

Tel: 91 398 89 10  
Fax: 91 398 89 09

[www.issl.uned.es](http://www.issl.uned.es)





## AGRADECIMIENTOS

Parece que fuera ayer cuando estaba aquí sentado, nuevamente, sobre esta misma página en blanco. Buscando las palabras que mejor la puedan decorar, que expliquen el largo camino recorrido cuando se estudia mientras se trabaja para llegar hasta este punto.

El tren recorrió el trayecto más lento de lo habitual, algunos viajeros se bajaron en estaciones cercanas al fin de línea, otras simplemente aguantaron hasta el final. Quizás estuvieran castigados, quizás simplemente no podían bajarse.

En primer lugar, me gustaría agradecer a mis abuelos. A mi abuela Carmen, por demostrarme que la vida consiste en reír, por siempre estar dispuesta a escuchar, por tener un corazón tan grande... por dejarme mirar atrás sin parar de sonreír. A mi abuelo Goyo, por enseñarme a no darse por vencido, que hay que echar azúcar para que los monitos vengan, por demostrarme que el silencio reconforta... por saber el momento oportuno de escuchar y hacer reír. En definitiva, gracias a ambos por compartir vuestro tiempo con nosotros. Ojalá pudierais leerlo.

A mi padre, por demostrarme que las personas cambian y que ser moderno es cumplir años. A mi madre, por corregirme nuevamente la memoria (prometo que no habrá más) y llamarme todos los miércoles. A mi hermana, por traer a Marce y saber que nunca tendré que ir a un concierto solo. A mi hermano, por compartir mis gustos y hacer que escucha cuando le cuento chorradas.

No puedo olvidarme de mi abuelo Eugenio, aún sigo sonriendo al recordar. A mi abuela Margarita, prometo visitarte más.

Para terminar, agradecer a todas aquellas personas que me han ayudado directa o indirectamente para que pueda estar hoy aquí, escribiendo estas palabras.

A todos, GRACIAS

Alberto



*A mis Abuelas*

*Y Abuelos*



*“Cualquier tecnología suficientemente avanzada es indistinguible de la magia”*

*Arthur C. Clarke (1917-2008)*  
*Escritor y científico británico*

*“No es el conocimiento sino el acto de conocer, no es la posición, sino el acto de llegar allí lo que produce el mayor gozo”*

*Karl Friedrich Gauss (1777-1855)*  
*Matemático, astrónomo y físico alemán*

*“Por muy lentamente que os parezca que pasan las horas, os parecerán cortas si pensáis que nunca más han de volver a pasar”.*

*Aldous Huxley (1894-1963)*  
*Escritor inglés*

*“La teoría es asesinada tarde o temprano por la experiencia”*

*Albert Einstein (1879-1955)*  
*Físico alemán*



# RESUMEN

En los últimos años hemos asistido a un aumento exponencial de aplicaciones que utilizan cualquier tipo de reconocimiento biométrico. Con la salida de los nuevos smartphones que casi todos incluyen reconocimiento facial y por huella dactilar, han puesto al mundo biométrico en la palestra de la innovación. Augurando un futuro más que prometedor, aunque como siempre, llegamos tarde...

El reconocimiento biométrico ha sido una parte importante en la creación de la sociedad. Desde el comienzo de la civilización se ha utilizado el reconocimiento biométrico para establecer clases y grupos sociales, estatus personales y de seguridad... en definitiva, para la estructuración de la sociedad.

Realmente el primer párrafo no es cierto del todo, antes de la existencia de los smartphones hubo otro acontecimiento que disparó la industria biométrica con el fin de ofrecer una mayor seguridad tanto en lugares públicos como privados. A partir del 2001, la industria biométrica ha producido un incremento en la demanda de sistemas de identificación humana fiables. Este crecimiento puede atribuirse a una serie de situaciones civiles y militares, otras surgen de la necesidad conjunta de seguridad requerida por la población y por supuesto, por una demanda de mejoras a nivel de consumidor.

Como todo avance tecnológico debe venir acompañado por un incremento en la eficiencia de los métodos de reconocimiento biométrico. A lo largo de este documento se presentará una evaluación de algunos de los algoritmos más importantes utilizados en reconocimiento biométrico. Se pretende acercar al lector una idea general sobre la industria biométrica y sobre todo del funcionamiento de las técnicas de reconocimiento facial.





# INDICE DE CONTENIDO

|                                                        |             |
|--------------------------------------------------------|-------------|
| <b>AGRADECIMIENTOS .....</b>                           | <b>VIII</b> |
| <b>RESUMEN .....</b>                                   | <b>XIV</b>  |
| <b>CAPITULO 1: INTRODUCCIÓN .....</b>                  | <b>211</b>  |
| <b>CAPITULO 2: REVISIÓN DEL MUNDO BIOMÉTRICO .....</b> | <b>23</b>   |
| 2.1 INTRODUCCIÓN .....                                 | 23          |
| 2.2 SISTEMAS BIOMÉTRICOS.....                          | 26          |
| 2.2.1 Tipos de operaciones .....                       | 27          |
| 2.2.2 Procesado de datos biométricos.....              | 27          |
| 2.3 EVALUACIÓN DE LOS SISTEMAS BIOMÉTRICOS .....       | 29          |
| 2.3.1 Competiciones.....                               | 29          |
| 2.3.2 Tipos de Evaluaciones .....                      | 30          |
| 2.3.3 Tipos de Errores .....                           | 32          |
| 2.4 BIOMETRÍA UNIMODAL.....                            | 35          |
| 2.4.1 Firma manuscrita .....                           | 35          |
| 2.4.2 Geometría Facial .....                           | 36          |
| 2.4.3 Huella Dactilar .....                            | 37          |
| 2.4.4 Iris.....                                        | 38          |
| 2.4.5 DNA .....                                        | 39          |
| 2.4.6 Huella de la palma de la mano .....              | 39          |
| 2.4.7 Dinámica del Tecleo .....                        | 40          |
| 2.4.8 Movimiento (caminar) .....                       | 40          |
| 2.4.9 Voz.....                                         | 41          |
| 2.5 BIOMETRÍA MULTIMODAL .....                         | 42          |
| <b>CAPITULO 3: RECONOCIMIENTO FACIAL .....</b>         | <b>45</b>   |
| 3.1 INTRODUCCIÓN .....                                 | 45          |
| 3.2 DETECCIÓN DE ROSTROS .....                         | 47          |
| 3.2.1 Neural Networks (NN).....                        | 49          |
| 3.2.2 Support Vector Machines (SVM).....               | 51          |
| 3.2.3 Algoritmo Viola-Jones .....                      | 52          |
| 3.3 RECONOCIMIENTO DE ROSTROS.....                     | 55          |
| 3.3.1 EigenFaces.....                                  | 57          |

|                                                          |                                              |           |
|----------------------------------------------------------|----------------------------------------------|-----------|
| 3.3.2                                                    | FisherFaces .....                            | 60        |
| 3.3.3                                                    | Local Binary Patterns Histograms (LBPH)..... | 62        |
| <b>CAPITULO 4: DISEÑO Y DESARROLLO .....</b>             |                                              | <b>64</b> |
| 4.1                                                      | INTRODUCCIÓN.....                            | 64        |
| 4.2                                                      | BASE DE DATOS .....                          | 66        |
| 4.3                                                      | ENTORNO DE TRABAJO .....                     | 67        |
| 4.3.1                                                    | OpenCV .....                                 | 68        |
| 4.3.2                                                    | Visual Studio 2015 Enterprise .....          | 69        |
| 4.3.3                                                    | Microsoft Access 2016 .....                  | 69        |
| 4.4                                                      | LINEA BASE .....                             | 70        |
| 4.4.1                                                    | Entrenamiento .....                          | 72        |
| 4.4.2                                                    | Test.....                                    | 76        |
| 4.5                                                      | ALTERNATIVA .....                            | 78        |
| <b>CAPITULO 5: RESULTADOS .....</b>                      |                                              | <b>84</b> |
| 5.1                                                      | INTRODUCCIÓN.....                            | 84        |
| 5.2                                                      | RESULTADOS LINEA BASE.....                   | 85        |
| 5.2.1                                                    | Método EigenFaces .....                      | 85        |
| 5.2.2                                                    | Método FisherFaces .....                     | 86        |
| 5.2.3                                                    | Método LBPH .....                            | 87        |
| 5.3                                                      | RESULTADOS ALTERNATIVA .....                 | 89        |
| 5.3.1                                                    | Método EigenFaces .....                      | 89        |
| 5.3.2                                                    | Método FisherFaces .....                     | 91        |
| 5.3.3                                                    | Método LBPH .....                            | 92        |
| <b>CAPITULO 6: CONCLUSIONES Y TRABAJOS FUTUROS .....</b> |                                              | <b>94</b> |
| 6.1                                                      | CONCLUSIONES .....                           | 94        |
| 6.2                                                      | TRABAJOS FUTUROS.....                        | 96        |

# INDICE DE FIGURAS

|                                                                                                              |    |
|--------------------------------------------------------------------------------------------------------------|----|
| FIGURA 1: SISTEMA BIOMÉTRICO .....                                                                           | 26 |
| FIGURA 2: DIAGRAMA DE FLUJO DEL PROCESADO BIOMÉTRICO .....                                                   | 28 |
| FIGURA 3: DIAGRAMA (RECURSOS/COSTES) Vs (VERACIDAD) DE LOS TEST OFF-LINE.....                                | 31 |
| FIGURA 4: UMBRAL DE DECISIÓN .....                                                                           | 33 |
| FIGURA 5: REPRESENTACIÓN DE FRR, FAR Y ERR .....                                                             | 34 |
| FIGURA 6: CURVA ROC .....                                                                                    | 34 |
| FIGURA 7: CURVA DET PARA DIFERENTES RASGOS BIOMÉTRICOS .....                                                 | 34 |
| FIGURA 8: ARQUITECTURA TÍPICA DE VERIFICACIÓN ON-LINE .....                                                  | 35 |
| FIGURA 9: EXTRACCIÓN DE CARACTERÍSTICAS DE LA BIOMETRÍA FACIAL .....                                         | 36 |
| FIGURA 10: EXTRACCIÓN DE CARACTERÍSTICAS DE LA HUELLA DACTILAR .....                                         | 37 |
| FIGURA 11: FASES DEL PROCESO DE RECONOCIMIENTO DE IRIS .....                                                 | 38 |
| FIGURA 12: EXTRACCIÓN DE CARACTERÍSTICAS DE LA PALMA DE LA MANO.....                                         | 39 |
| FIGURA 13: CARACTERÍSTICAS DE LA DINÁMICA DEL TECLEO .....                                                   | 40 |
| FIGURA 14: EXTRACCIÓN DE CARACTERÍSTICAS DE LA SILUETA.....                                                  | 40 |
| FIGURA 15: EXTRACCIÓN DE LAS CARACTERÍSTICAS DE LA VOZ .....                                                 | 41 |
| FIGURA 16: VARIOS ESCENARIOS DE UN SISTEMA BIOMÉTRICO MULTIMODAL .....                                       | 44 |
| FIGURA 17: DIAGRAMA DE RED NEURONAL UNITARIA Y MULTICAPA .....                                               | 49 |
| FIGURA 18: HIPERPLANO SEPARADOR DE CLASES .....                                                              | 51 |
| FIGURA 19: HAARS-LIKE FEATURES: (A) 2-RECTANGLE FEATURE (B) 3-RECTANGLE FEATURE (C) 4-RECTANGLE FEATURE..... | 52 |
| FIGURA 20: HAAR-LIKE FEATURES EXTENDIDAS POR LIENHART Y MAYDT .....                                          | 53 |
| FIGURA 21: CÁLCULO DE LA REGIÓN D CON VALORES DE IMAGEN INTEGRAL.....                                        | 54 |
| FIGURA 22: CLASIFICADOR EN CASCADA.....                                                                      | 54 |
| FIGURA 23: PRINCIPALES MÉTODOS EN RECONOCIMIENTO FACIAL.....                                                 | 56 |
| FIGURA 24: EIGENFACES.....                                                                                   | 59 |
| FIGURA 25: FISHERFACES.....                                                                                  | 61 |
| FIGURA 26: OBTENCIÓN DE LOS PARÁMETROS LBP.....                                                              | 63 |
| FIGURA 27: OPERADOR LBPH SOBRE IMAGEN .....                                                                  | 63 |
| FIGURA 28: METODOLOGÍA DE TRABAJO.....                                                                       | 65 |
| FIGURA 29: AT&T DATABASE .....                                                                               | 66 |
| FIGURA 30: ESTRUCTURA FICHERO ENTRADA .....                                                                  | 71 |
| FIGURA 31: ESTRUCTURA FICHERO SALIDA.....                                                                    | 71 |
| FIGURA 32: DIAGRAMA DE ENTRENAMIENTO .....                                                                   | 72 |
| FIGURA 33: DIAGRAMA DE TEST .....                                                                            | 76 |
| FIGURA 34: (A) ROSTRO DETECTADO; (B) ROSTRO RECORTADO; (C) ROSTRO RECORTADO EN B/N .....                     | 80 |
| FIGURA 35: DETECCIÓN ROSTRO ANTE COMPONENTES ESTRUCTURALES .....                                             | 80 |
| FIGURA 36: VARIACIONES LUMINOSIDAD EN ROSTRO .....                                                           | 81 |
| FIGURA 37: DETECCIÓN MÚLTIPLE DE ROSTROS .....                                                               | 83 |
| FIGURA 38: DETECCIÓN DE OJOS .....                                                                           | 83 |
| FIGURA 39: VARIACIÓN FRR Y FAR EN EFR .....                                                                  | 86 |
| FIGURA 40: VARIACIÓN FRR Y FAR EN FFR .....                                                                  | 87 |
| FIGURA 41: VARIACIÓN FRR Y FAR CON LBPH .....                                                                | 88 |
| FIGURA 42: VARIACIÓN FRR Y FAR EN EFR CON ALTERNATIVA.....                                                   | 90 |
| FIGURA 43: VARIACIÓN FRR Y FAR EN FFR CON ALTERNATIVA.....                                                   | 91 |
| FIGURA 44: VARIACIÓN FRR Y FAR EN LBPH CON ALTERNATIVA .....                                                 | 93 |

## INDICE DE ECUACIONES

|                  |    |
|------------------|----|
| ECUACIÓN 1.....  | 53 |
| ECUACIÓN 2.....  | 58 |
| ECUACIÓN 3.....  | 58 |
| ECUACIÓN 4.....  | 58 |
| ECUACIÓN 5.....  | 58 |
| ECUACIÓN 6.....  | 58 |
| ECUACIÓN 7.....  | 59 |
| ECUACIÓN 8.....  | 60 |
| ECUACIÓN 9.....  | 60 |
| ECUACIÓN 10..... | 60 |
| ECUACIÓN 11..... | 62 |
| ECUACIÓN 12..... | 62 |
| ECUACIÓN 13..... | 63 |
| ECUACIÓN 14..... | 63 |

## INDICE DE TABLAS

|                                                                                            |    |
|--------------------------------------------------------------------------------------------|----|
| TABLA 1: COMPARACIÓN DE VARIAS CARACTERÍSTICAS BIOMÉTRICAS (A=ALTO, M=MEDIO, B=BAJO) ..... | 25 |
| TABLA 2: RESULTADOS LB CON EIGENFACES.....                                                 | 85 |
| TABLA 3: RESULTADOS LB CON FISHERFACES.....                                                | 86 |
| TABLA 4: RESULTADOS LB CON LBPH .....                                                      | 88 |
| TABLA 5: RESULTADOS ALTERNATIVA CON EFR .....                                              | 89 |
| TABLA 6: COMPARATIVA LÍNEA BASE/ALTERNATIVA EN EFR.....                                    | 90 |
| TABLA 7: RESULTADOS ALTERNATIVA CON FFR.....                                               | 91 |
| TABLA 8: COMPARATIVA LÍNEA BASE/ALTERNATIVA EN FFR.....                                    | 92 |
| TABLA 9: RESULTADOS ALTERNATIVA CON LBPH .....                                             | 92 |
| TABLA 10: COMPARATIVA LÍNEA BASE/ALTERNATIVA EN LBPH .....                                 | 93 |



# CAPÍTULO 1

## INTRODUCCIÓN

Hoy en día la seguridad es una parte importante en nuestras vidas. Los mecanismos de seguridad están tan asumidos por la sociedad que los utilizamos continuamente sin extrañarnos por ello, como si fuera normal. Desde encender un ordenador hasta conducir un coche o desbloquear un teléfono móvil. Se utilizan prácticamente en cualquier ámbito y lugar: coches, casas, oficinas de trabajo, edificios gubernamentales, museos, aeropuertos, teléfonos móviles... Pero no siempre fue así, todo avance viene precedido de un desencadenante. Hubo un punto de inflexión que ayudó a intensificar los mecanismos de seguridad, consiguiendo así reflotar y ampliar el ámbito de uso de una industria en concreto: “La industria biométrica”.

Los atentados de las Torres Gemelas, los del 11M... ayudaron a la implantación definitiva de la industria biométrica en el mercado actual. A partir de ese momento las exigencias de seguridad se fueron intensificando gradualmente. Las tarjetas de identidad y los pasaportes ya no eran suficientes debido a su fácil falsificación. Se necesitaba identificar individuos con participación tanto activa como pasiva de los mismos. Por lo tanto, se necesitaban procedimientos nuevos e infalsificables. Convirtiendo a la biometría en una creciente línea de investigación tecnológica y convirtiéndose así en una industria con mucho futuro.

A raíz de aquellos acontecimientos, y los que desgraciadamente les han ido siguiendo, se generó una sensación de inseguridad pública en aeropuertos, metros, sitios públicos y estaciones de tren. Probablemente la identificación de individuos es el

cometido más importante para aumentar la seguridad en las zonas de acceso público o de tránsito masivo de personas. Detectar sujetos potencialmente peligrosos antes de su actuación es un “plus” añadido a la seguridad personal. La introducción de la biometría, y más concretamente la identificación biométrica, consiguió disminuir esa sensación de inseguridad generada.

La biometría ofrece múltiples formas de reconocimiento de personas, y mediante la tecnología ofrecida por la industria biométrica se pretende evitar nuevos desastres. Este tipo de reconocimiento basado en rasgos intrínsecos a la persona lleva el hándicap añadido de no unicidad. Bajo esa característica no existen dos personas con todos los rasgos biométricos iguales. El reconocimiento biométrico de personas ofrece dos alternativas: la verificación y la identificación. Con la verificación se pretende corroborar la autenticidad de la persona, mientras que con la identificación se pretende reconocer a una persona.

Para realizar una identificación biométrica se debe comparar la muestra biométrica con las existentes en la base de datos. Cabe resaltar que, por lo tanto, se necesita un elevado número de comparaciones hasta conseguir una identidad, en el caso de que estuviera registrada. Si los rasgos biométricos de la persona no fueron previamente almacenados en la base de datos, el reconocimiento biométrico es ineficaz.

El objetivo del proyecto es desarrollar y evaluar técnicas de visión artificial aplicadas a un problema de reconocimiento facial. A lo largo del documento se desarrollarán y evaluarán diferentes técnicas de reconocimiento facial.

Este documento se estructura de la siguiente manera: En primer lugar, se describen las motivaciones y objetivos del proyecto. Se continua con una revisión del mundo biométrico actual. Se prosigue con una revisión del estado del arte en el reconocimiento facial. En el siguiente capítulo se expone la metodología y técnicas de evaluación más importantes, así como una descripción de los algoritmos más importantes e implementados en el proyecto. Posteriormente, se evaluarán los resultados obtenidos tanto en la línea base como en la alternativa propuesta. Se concluye la memoria con el apartado de conclusiones y futuras vías de investigación.

# CAPÍTULO 2

---

## REVISIÓN DEL MUNDO BIOMÉTRICO

En este capítulo se ofrece una visión resumida de la biometría. En primer lugar, se revisa el concepto de biometría para continuar con la descripción de un sistema biométrico y sus operaciones. Se prosigue analizando los sistemas de experimentación y las medidas de evaluación de los mismos. Y se termina con la enumeración de las tecnologías biométricas más utilizadas y los sistemas multimodales.

### 2.1 INTRODUCCIÓN

Desde el comienzo de la civilización la identificación de humanos ha sido crucial para la construcción de una sociedad. Desde hace cientos de años se fueron utilizando las características de nuestro propio cuerpo para reconocernos entre nuestros semejantes [Anil K. Jain et al. 2004a]. Entre ellas la cara, la voz, el modo de caminar...

Biometría se refiere al concepto de reconocimiento de individuos mediante algunos rasgos o características personales (biológicas, físicas, psicológicas o de comportamiento) y unívocas [Anil K. Jain et al. 2004a]. Según [Dessimoz et al, 2005] estas características biológicas deben de ser mensurables o estadísticamente medibles.

El santo y seña de los siglos pasados será en el siglo XXI el propio cuerpo y sus características físicas, únicas y distintas a las de cualquier otro ser humano. La industria



biométrica abre un amplio abanico de posibilidades tanto en accesibilidad como en seguridad, está en continuo desarrollo y en los últimos años se han producido importantes avances tecnológicos. El reconocimiento biométrico es un concepto antiguo, véase [Raphael et al, 1974], pero se podría decir que los sistemas biométricos y el cuerpo humano serán la llave maestra que nos abrirá muchas puertas en el futuro próximo.

Para que un rasgo humano sea útil desde el punto de vista biométrico debe de ser parametrizable o mensurable. Además de ello, para que ese mismo rasgo pueda ser utilizado como característica biométrica debe cumplir una serie de requisitos que aparecen en [Anil K. Jain et al. 2004a]:

- ✓ *Universalidad*: El rasgo biométrico debería existir para todas las personas de una población.
- ✓ *Distintivos o inconfundibles*: Dos personas deben ser suficientemente diferentes en términos de rasgos.
- ✓ *Permanentes*: Las características deben de ser suficientemente invariantes con el tiempo dentro de un sujeto.
- ✓ *Accesibles*: Las características deben ser cuantificadas mediante una operación moderadamente sencilla.

A menudo, a la hora de elegir qué rasgos biométricos son los más adecuados para una aplicación concreta existen otros tipos de requisitos que deberían ser tomados en cuenta:

- ✓ *Rendimiento*: Hace referencia a la precisión y velocidad del proceso de reconocimiento como a los recursos necesarios para conseguir los niveles de rendimiento deseados.
- ✓ *Aceptabilidad*: Representa el grado de aceptación que la sociedad presenta ante que un determinado rasgo sea capturado.
- ✓ *Invulnerabilidad*: Refiere a la robustez que ofrece el rasgo al sistema frente a métodos falsificables.

Una única característica biométrica no es la herramienta de reconocimiento final y la elección de la misma dependerá de la aplicación. En la (Tabla 1: Comparación de varias técnicas biométricas) extraída de [Anil K. Jain et al. 2004a] se muestra las características biométricas más comunes en los sistemas biométricos actuales, así como el grado de cumplimiento de los requisitos anteriores.

Aunque en el ámbito forense (judicial, policial y pericial) el análisis científico de evidencias biométricas se ha venido utilizando desde hace más de un siglo, el reconocimiento biométrico automático como medio de autenticación personal en

aplicaciones comerciales o civiles, es un área de investigación y desarrollo bastante más reciente y en pleno crecimiento.

| TIPO           | Rasgos Biométricos          | Universalidad | Distintivos o Inconfundibles | Permanentes | Accesibles | Rendimiento | Aceptabilidad | Invulnerabilidad |
|----------------|-----------------------------|---------------|------------------------------|-------------|------------|-------------|---------------|------------------|
| Fisiológicas   | Voz <sup>1</sup>            | M             | B                            | B           | M          | B           | A             | A                |
|                | Huella dactilar             | M             | A                            | A           | M          | A           | M             | M                |
|                | Iris                        | A             | A                            | A           | M          | A           | B             | B                |
|                | Retina                      | A             | A                            | M           | B          | A           | B             | B                |
|                | Geometría facial            | A             | B                            | M           | A          | B           | A             | A                |
|                | Venas del dorso de la mano  | M             | M                            | M           | M          | M           | M             | B                |
|                | Geometría de la palma de la | M             | M                            | M           | A          | M           | M             | M                |
|                | Huella de la palma de la    | M             | A                            | A           | M          | A           | M             | M                |
|                | Geometría de la oreja       | M             | M                            | A           | M          | M           | A             | M                |
|                | Olor corporal               | A             | A                            | A           | B          | B           | M             | B                |
| Comportamiento | Voz <sup>1</sup>            |               |                              |             |            |             |               |                  |
|                | Firma                       | B             | B                            | B           | A          | B           | A             | A                |
|                | Dinámica de tecleo          | B             | B                            | B           | M          | B           | M             | M                |
|                | Movimiento y/o cadencia del | M             | B                            | B           | A          | B           | A             | M                |
|                | DNA                         | A             | A                            | A           | B          | A           | B             | B                |

**Tabla 1: Comparación de varias características biométricas (A=Alto, M=Medio, B=Bajo)**

<sup>1</sup>La voz encierra tanto componentes físicas como componentes psicológicas y de comportamiento

## 2.2 SISTEMAS BIOMÉTRICOS

En esencia el núcleo del sistema biométrico es un sistema de reconocimiento de patrones [Anil K. Jain et al, 2000]. De forma vaga, [Watanabe, 1985] un patrón biométrico es “una entidad a la que se le podría adjudicar un nombre”. Por tanto, un sistema biométrico es un sistema que reconoce individuos mediante un vector de características derivado de algún rasgo físico o de comportamiento que el individuo posee [Anil K. Jain et al, 2004a].

Los sistemas biométricos se caracterizan por realizar su trabajo en tres fases. [Anil K. Jain et al, 2000] En una primera fase, se obtienen muestras biométricas de un individuo mediante algún sensor o sensores específicos. Seguidamente se extrae el conjunto de valores correspondientes a las características relevantes de la muestra (vector de características) adquirida que se pueda manipular y medir mediante algún algoritmo o fórmula matemática. Por último, se compara los datos originados anteriormente con los almacenados y se decide la aceptación o rechazo del individuo (**¡Error! La autoreferencia al marcador no es válida.**).

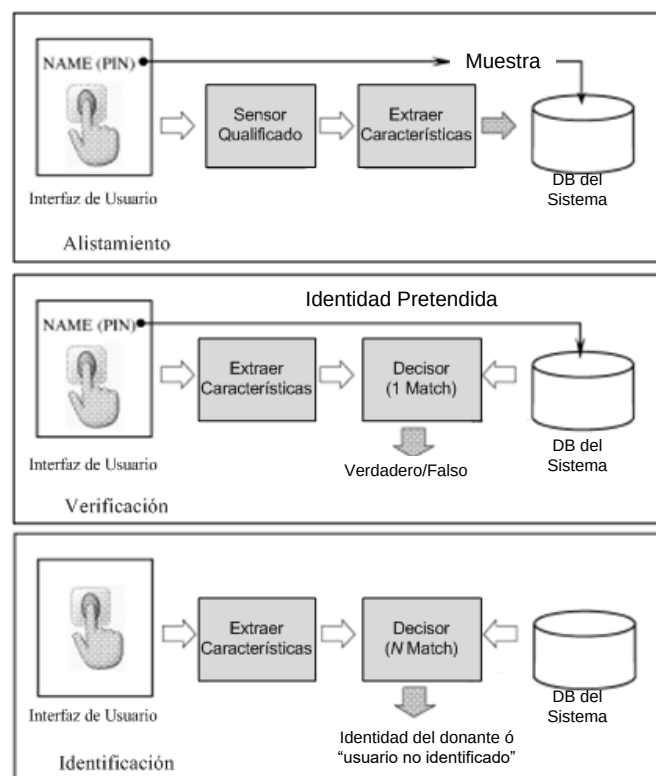


Figura 1: Sistema Biométrico, extraída de [Anil K. Jain et al. 2004a]

### 2.2.1 Tipos de operaciones

Los pasos comunes explicados anteriormente son utilizados en una serie de operaciones biométricas de alto nivel [D. Dessimoz et al, 2005] [Anil K.Jain et al, 2004a]: Alistamiento, Verificación e Identificación.

- *Alistamiento*: Con esta operación se añade un nuevo usuario al sistema. Mediante un cierto número de representaciones biométricas se entrena un modelo para el usuario y se adapta el modelo de los usuarios de fondo en caso de ser necesario (**¡Error! La autoreferencia al marcador no es válida.**). También se pueden dar de baja a usuarios que ya no pertenecen al sistema biométrico.
- *Verificación*: La verificación consiste en comparar la representación biométrica de un usuario donante frente al modelo de un usuario pretendido. Una vez contrastada la información con el modelo se obtiene una puntuación que se coteja con un umbral para aceptar o denegar la identidad (**¡Error! La autoreferencia al marcador no es válida.**) [J. L. Wayman, 2001].
- *Identificación*: Mediante esta operación se busca la representación biométrica más parecida en el sistema a la muestra de entrada (**¡Error! La autoreferencia al marcador no es válida.**). El sistema reconoce a un individuo mediante la comparación de sus características biométricas con los modelos de todos los usuarios almacenados en el sistema. A partir de las puntuaciones obtenidas para cada modelo se establece la identidad [Karthik Nandakumar et al. 2009]. La identificación es un proceso relativamente lento debido al elevado número de comparaciones que se deben realizar para dar con una identidad.

### 2.2.2 Procesado de datos biométricos

El procesado de datos biométricos es una sucesión lineal de operaciones. Para cualquier modalidad biométrica, los datos serán transformados siguiendo una serie de pasos que se pueden observar de manera esquematizada en la Figura 2 tomada de [D. Dessimoz et al. 2005]:

- ✓ *Captura*: La información es digitalizada por un sensor específico y almacenada.
- ✓ *Pre-Procesado*: Se realiza una normalización de la información en el dominio de la señal con el fin de eliminar información corrupta. Quedando la información preparada para la posterior extracción de características.
- ✓ *Extracción de características*: De la señal digitalizada y pre-procesada se extrae la información más relevante consiguiendo una nueva representación de la señal en un nuevo dominio. De esta forma, se reduce la dimensionalidad de la señal de entrada y se crea una representación a nivel de características del patrón de

entrada. Esta nueva señal se emplea en el posterior reconocimiento de patrones y se la conocerá en el documento como vector de características.

- ✓ *Post-Procesado:* Las características obtenidas en el paso anterior se vuelven a normalizar para eliminar el posible sesgo, o bien para conseguir una mejor adaptación de las mismas al clasificador.
- ✓ *Creación del modelo:* Una vez se extrajo la información relevante de la señal y se obtuvo la representación en el nuevo dominio, se procede a la generación del modelo para dicho usuario. La generación del modelo no es más que una representación matemática y genérica de un usuario a partir de su vector de características. Este modelo será usado para futuras comparaciones.
- ✓ *Creación del modelo de fondo:* También llamado modelo del mundo. Algunos sistemas necesitan de la creación de un modelo para normalizar las puntuaciones. Este modelo representa una “media” de los usuarios con respecto la población del sistema.
- ✓ *Almacenamiento de modelos:* Una vez estimados los datos que definen la identidad de un usuario los modelos son almacenados para su futuro uso en operaciones biométricas.
- ✓ *Comparación de modelos:* Se compara un conjunto de vectores de características con el modelo biométrico almacenado de un usuario. Esta comparación da como resultado una puntuación que indica cuanto se asemejan los datos biométricos de entrada con las identidades de los modelos biométricos que han sido cotejadas. En ocasiones, usar el modelo del mundo en la comparación mejora los resultados.
- ✓ *Establecimiento de umbrales:* A partir de los datos de usuarios reales e impostores se calcula un umbral de decisión que marca la aceptación de la identidad en el sistema.

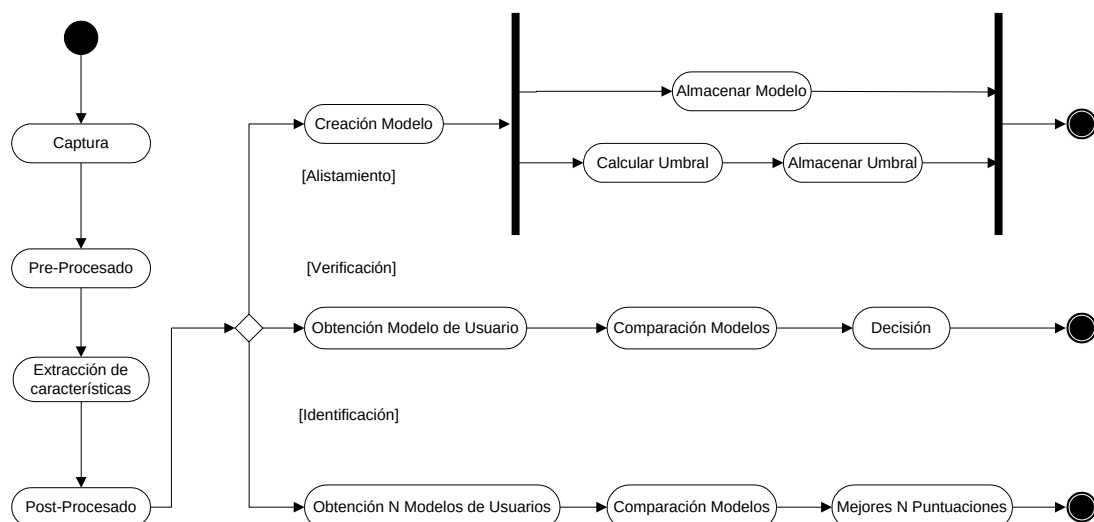


Figura 2: Diagrama de flujo del Procesado Biométrico

## 2.3 EVALUACIÓN DE LOS SISTEMAS BIOMÉTRICOS

Durante años se han ido introduciendo multitud de avances en los sistemas biométricos, por parte de la comunidad científica y organizaciones comerciales. Como bien describe [R. Capelli et al, 2006], las prestaciones estaban siendo medidas mediante protocolos de evaluación no estandarizados y bases de datos propias. Obteniendo así un conjunto de datos no susceptibles de ser comparados y de entredicha importancia.

Un protocolo de evaluación definido por [P.J. Phillips et al, 2000] determina tanto la forma en la que se testea un sistema y se miden sus prestaciones como la información utilizada para evaluarlo. Para que un sistema de evaluación sea aceptado debe de cumplir una serie de requisitos y someterse a una metodología de evaluación. Todo este trabajo ha sido realizado por diferentes organismos nacionales y finalmente se aunaron en la Common Criteria (CC) <sup>2</sup> para posteriormente ser estandarizada por la ISO 15408 [R. S. Reillo et al, 2007]. Para que un protocolo de evaluación sea aceptado por la comunidad científica, los detalles de sus procedimientos han de estar publicados junto con las medidas de prestaciones propuestas y algunos ejemplos representativos del conjunto de datos.

### 2.3.1 Competiciones

Durante el siglo XXI se han desarrollado multitud de iniciativas con el claro fin de poder establecer una medida de evaluación única y fiable. Las iniciativas más importantes son:

- ✓ La Universidad de Bolonia desarrolla desde el año 2000 una competición bianual, la Fingerprint Verification Competition<sup>3</sup>, para evaluar el Estado del Arte de los sistemas de verificación basados en huella dactilar.
- ✓ El Instituto Nacional de Estándares y Tecnología (NIST)<sup>4</sup> desarrolla desde el año 2005 una competición para evaluar el estado del arte de los sistemas de reconocimiento de iris.
- ✓ El Instituto de Ciencias de la Academia China de las Ciencias realiza desde el año 2004 competiciones para evaluar las prestaciones de los sistemas de verificación de identidad basados en iris. Así mismo, desarrolla la Signature Verification Competition para valorar los sistemas de verificación de identidad basados en firma on-line.
- ✓ El Face Recognition Vendor Test<sup>5</sup> desarrolla competiciones para la evaluación de

<sup>2</sup> <http://www.commoncriteriaportal.org/>

<sup>3</sup> <http://bias.csr.unibo.it/fvc2006/>

<sup>4</sup> <http://iris.nist.gov/ice/>

<sup>5</sup> <http://www.frvt.org/>

sistemas comerciales basados en geometría facial desde el año 2000. En sus inicios llamado FERET [P.J. Phillips et al, 2000b].

- ✓ El Instituto Nacional de Estándares y Tecnología (NIST)<sup>6</sup> desarrolla desde el año 2005 una competición para evaluar el estado del arte de los sistemas de reconocimiento de la cara.

### 2.3.2 Tipos de Evaluaciones

La evaluación de los sistemas biométricos es una parte muy importante del reconocimiento biométrico. Una evaluación es correcta cuando las puntuaciones obtenidas permiten distinguir entre los distintos enfoques y tecnologías existentes. Las fortalezas y debilidades encontradas en el proceso de evaluación indican las aplicaciones para las que las tecnologías son más adecuadas. Se describen a continuación los distintos tipos de evaluaciones existentes según [P. J. Phillips et al, 2000] [A. J. Mansfield et al, 2002]:

- *Evaluación de Tecnología:* Es la tecnología de evaluación más generalizada. Se comparan los diferentes algoritmos existentes y prototipos de laboratorio para una misma tecnología. Todos los algoritmos son testeados con una base de datos estandarizada y recogida mediante un sensor universal. Se realiza para determinar el estado del arte de una tecnología e identificar los progresos tecnológicos.
- *Evaluación de Escenario:* Valora las prestaciones globales de los sistemas para un prototipo de escenario o aplicación simulada. El propósito principal de este tipo de evaluación es determinar si una tecnología biométrica es lo suficientemente madura para alcanzar los requisitos de prestaciones en un tipo de aplicaciones. Se suele evaluar el comportamiento de los diferentes algoritmos y combinación de sensores bajo condiciones reales.
- *Evaluación Operacional:* Aunque es similar a la evaluación de escenario su diferencia radica en que valora las prestaciones de un algoritmo específico para un tipo de aplicación específica. El propósito principal de la evaluación operacional es determinar si un sistema biométrico alcanza los requisitos para una aplicación específica.

La evaluación de un sistema biométrico requiere una captura de información [R. Capelli et al, 2006]. Esta información se utiliza tanto para el entrenamiento del sistema biométrico, como para la obtención de las puntuaciones en la etapa de reconocimiento [Anil K. Jain et al, 2004a]. Esa información puede ser utilizada en tiempo real o por el contrario se puede almacenar para un uso posterior. Dependiendo de la manera de utilizar

---

<sup>6</sup> <http://face.nist.gov/frgc/>

la información según [A. J. Mansfield et al, 2002] [Capelli et al, 2006] se pueden clasificar los test de evaluación en dos grupos:

- *On-line*: Este tipo de evaluación se realiza con la presencia del usuario. El alistamiento y el cálculo de las puntuaciones se realizan inmediatamente después de obtener la información biométrica. Posee la ventaja de que sólo se almacena la información estrictamente necesaria debido a que la muestra se elimina tras su utilización. Desafortunadamente, una misma operación no puede ser repetida por falta de la muestra descartada.
- *Off-line*: Este tipo de evaluación se basa en una colección de datos capturados y almacenados previamente necesitando así un mayor almacenamiento de información. Este tipo de evaluación ofrece un mayor control sobre qué información será utilizada en cada operación. Permitiendo reproducir un mismo test bajo las mismas condiciones.

A su vez, los test de evaluación off-line se subdividen en diferentes tipos como bien recoge [D. Carrero et al., 2009]. No todos ellos consumen los mismos recursos computacionales (Figura 3), extraída de [Capelli et al, 2006]:

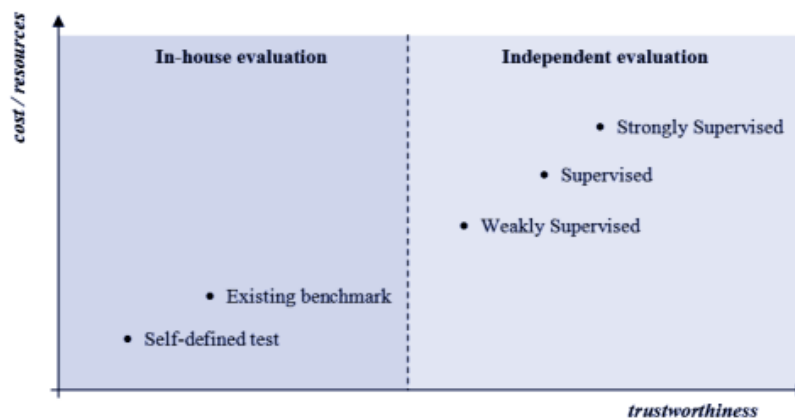


Figura 3: Diagrama (recursos/costes) Vs (veracidad) de los test off-line

- ✓ *In-house – self defined test*: La recolección de la información biométrica y la definición del protocolo de prueba es definido por el investigador. Como consecuencia, no permite ningún tipo de comparación debido a que los datos de test no pueden ser reproducidos por terceras personas.
- ✓ *In-house – existing Benchmark*: El test se realiza sobre una base de datos de carácter público y de acuerdo con los protocolos existentes. En este caso, los resultados obtenidos son comparables con otros obtenidos mediante el mismo protocolo y la misma base de datos. El principal inconveniente reside en la imposibilidad de predecir las verdaderas prestaciones del sistema en aplicaciones reales debido al posible sobreajuste realizado sobre los datos utilizados.



- ✓ *Independent – weakly supervised*: La base de datos no se hace pública hasta el comienzo de la evaluación. Por un lado, el test se realiza en el entorno del investigador mediante datos sin etiquetar y con limitaciones temporales. Por otro lado, las prestaciones del sistema se determinan por evaluadores independientes a partir de las puntuaciones obtenidas por el investigador durante la prueba del sistema.
- ✓ *Independent – supervised*: Este enfoque es muy similar al anterior con la salvedad de que el test se realiza en la localización del evaluador sobre el hardware del investigador. En este test el evaluador puede mejorar el control de la evaluación pero no hay forma de comparar la eficiencia computacional, algunas estadísticas de interés no se obtienen y no se puede impedir la normalización de las puntuaciones.
- ✓ *Independent – strongly supervised*: La base de datos es aislada y no se comparte hasta la conclusión del test. Los componentes software se amoldan para dar unos protocolos de entrada/salida que se testean en la zona el evaluador sobre su hardware. Los algoritmos son testeados sobre un entorno completamente controlado donde las operaciones de entrada/salida son estrictamente monitorizadas.

### 2.3.3 Tipos de Errores

No existen sistemas biométricos que realicen perfectas. El rendimiento de los sistemas biométricos también es medido, cuantificado y expresado mediante una serie de indicadores [Dessimoz et al, 2005]. Muchos de estos indicadores son utilizados para medir el rendimiento de las operaciones de los sistemas biométricos (véase 2.2.1).

Durante el proceso de alistamiento pueden aparecer una serie de dificultades a la hora de capturar los datos biométricos. Dificultando la creación de modelos de usuarios en el sistema biométrico. Las medidas de rendimiento que reflejan estos problemas son [Anil k. Jain et al, 2004a]:

- ✓ *FTC (Failure to Capture) o FTA (Failure to Acquisition)*: Es el porcentaje de veces que el sistema no puede obtener una muestra biométrica de suficiente calidad.
- ✓ *FTE (Failure to Enroll) o FER (Failure to Enroll Rate)*: Se define como el porcentaje de veces que el sistema no ha podido crear un modelo de usuario de suficiente calidad.
- ✓ *TTE (Time to Enroll)*: Es definido como el tiempo transcurrido desde que se capturan la información biométrica hasta que se crea el modelo.

La captura de los datos se ve influenciada por la calidad del sensor, cambios en las características psíquicas y de comportamiento del individuo, las condiciones ambientales

en el momento de la captura. Todos estos factores pueden llevar a errores en la identificación o reconocimiento de un individuo. Debido a esta peculiaridad, se establece un umbral que cuantifica la similitud entre los datos de entrada y el modelo de usuario almacenado (Figura 4), extraída de [Anil k. Jain et al, 2004a]. Como consecuencia, se producen los siguientes tipos de errores de decisión:

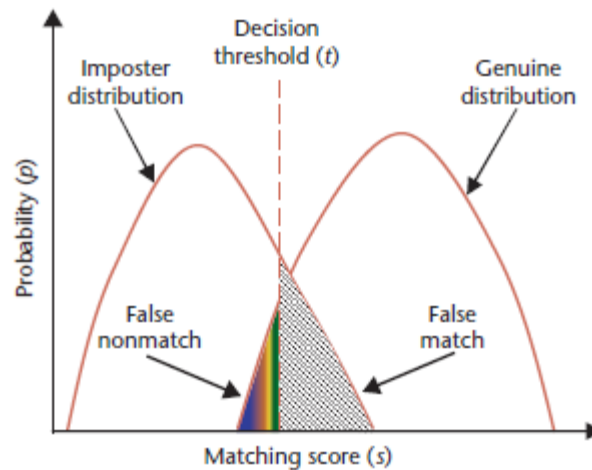


Figura 4: Umbral de decisión

- ✓ *FRR (False Rejection Rate) o Tasa de Error de Tipo I*: Se define como la probabilidad de que se rechace a un usuario legítimo por ser el sistema incapaz de identificarle.
- ✓ *FAR (False Acceptance Rate) o Tasa de Error de Tipo II*: Es la probabilidad de que se autentifique a un usuario ilegítimo.

Estos dos errores dan lugar a la aparición de estos otros errores de comparación:

- ✓ *FMR (False Match Rate) o Falso Positivo*: Se define como la frecuencia con la que se produce FAR.
- ✓ *FNMR (False Non-Match Rate) o Falso Negativo*: Es la frecuencia con la que se produce FRR.

Existen otras medidas que pueden ser usadas como una medida general de las prestaciones de un sistema biométrico:

- ✓ *EER (Equal Error Rate)*: Hace referencia al valor del umbral de decisión para el que los valores de FAR y FRR son iguales (Figura 5 extraída de [Anil k. Jain et al, 2004a]).
- ✓ *HTER (Half Total Error Rate)*: Es la semisuma de los valores de FAR y FRR.
- ✓ *Coste*: Es la suma ponderada de los valores de FAR y FRR. En este cálculo se asignan costes a los diferentes tipos de errores del sistema.

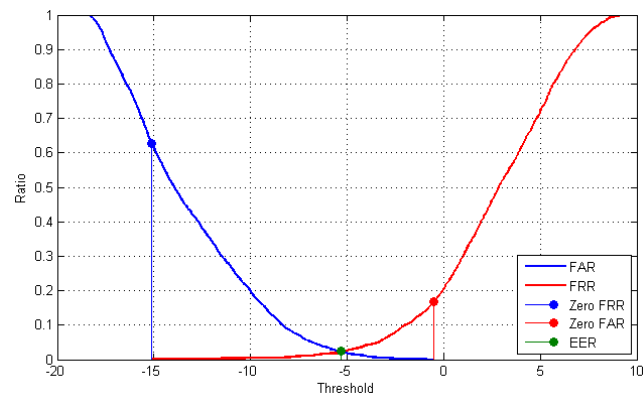


Figura 5: Representación de FRR, FAR y ERR

Una forma alternativa de evaluar las prestaciones de un sistema biométrico es mediante la curva ROC (Receiver Operating Curve) [Fogarty et al, 2005] o mediante la curva DET (Detection Error Trade-Off). La curva ROC (Figura 6 extraída de [Fogarty et al, 2005]) consiste en la representación gráfica del ratio de falsa aceptación (FAR) frente al ratio de correcta aceptación ( $1 - \text{FRR}$ ) en función del umbral de decisión. Cuanto mayor sea el área por debajo de la curva ROC mayor confianza posee el sistema biométrico, lo ideal es que el área sea igual a la unidad. Por otro lado, la curva DET (Figura 7 extraída de [Fogarty et al, 2005]) consiste en la representación gráfica de FNMR frente a FMR. Una de las principales ventajas de estas curvas es la posibilidad de contrastar las prestaciones de diferentes sistemas en un solo gráfico.

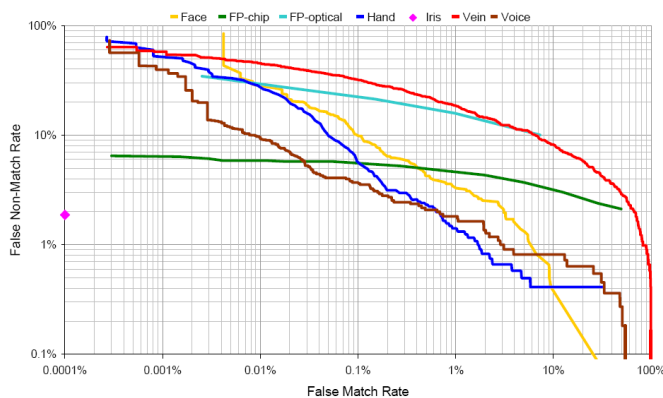


Figura 7: Curva DET para diferentes rasgos biométricos

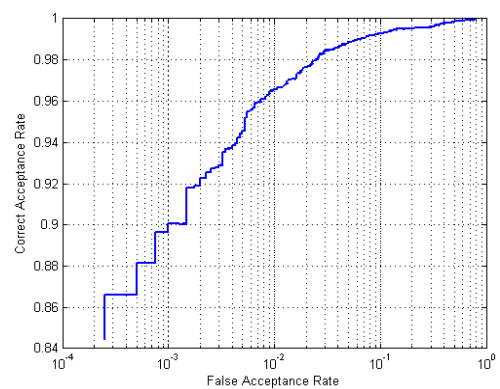


Figura 6: Curva ROC

## 2.4 BIOMÉTRIA UNIMODAL

El mundo de la biometría abarca un amplio abanico de tecnologías diferentes. En este apartado se intentará estudiar algunas de las tecnologías biométricas unimodales más conocidas.

### 2.4.1 Firma manuscrita

La firma manuscrita es un rasgo que plasma el comportamiento de una persona. Está caracterizado por movimientos rápidos de la mano debidos a la actividad del sistema neuromuscular. Se puede trabajar con una imagen previamente guardada, llamada firma estática u off-line. O, por el contrario, con la información instantánea (trayectoria, presión, velocidad, fuerza) recogida mediante algún sensor (tableta de firmas), llamada firma dinámica u on-line, ambas recogidas en [V. S. Nalwa, 1997]. La Figura 8 recoge el proceso de verificación on-line, extraída de [V. S. Nalwa, 1997].

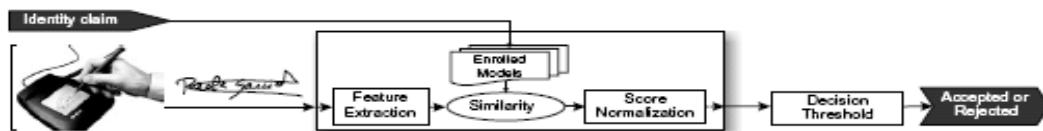


Figura 8: Arquitectura típica de verificación on-line

Los principales métodos para realizar la verificación de firma on-line se basan en los algoritmos Dynamic Time Wrapping (DTW) que se pueden encontrar en [Hao Feng et al, 2003], Hidden Markov Model (HMM) [J. Fierrez et al, 2007] con distribuciones de probabilidad gaussiana (GMM) [J. Richiardi et al, 2003] y redes neuronales (NN) descritas en [H. D. Chang et al, 1993]. También se han utilizado, aunque con peores resultados que los anteriores, la distancia Euclídea de la manera descrita en [T. Rhee et al, 2001] y la correlación global y local [V. S. Nalwa, 1997]. Los sistemas que trabajan con firma offline pueden extraer las características estáticas mediante la geometría de la imagen y verificar usando NN [Huang et al, 1997]. También pueden utilizar HMM y la Discrete Radon Transform (DRT) tal como describe [J. Coetzer et al, 2004]. El inconveniente de este tipo de biometría es la variabilidad temporal entre firmas y se pueden producir falsificaciones entrenadas.

### 2.4.2 Geometría Facial

Probablemente, el reconocimiento facial sea el método natural más utilizado con el que identificamos a las personas. Se pueden obtener características biométricas sin que el individuo se percate de ello (método no intrusivo). En la actualidad, la mayoría de los sistemas actuales de reconocimiento facial determinan la apariencia de un sujeto a través de la obtención de puntos especialmente significativos de la cara (llamados puntos nodales) mediante información 2D o 3D. A partir de estos puntos, como describe [W. Zhao et al, 2003] se determina la distancia entre los ojos, la anchura de la nariz, la distancia del ojo a la boca o la longitud de la línea de la mandíbula. La Figura 9 (extraída de [C. Conde et al, 2008]) recoge el proceso de extracción de características de la biometría facial.

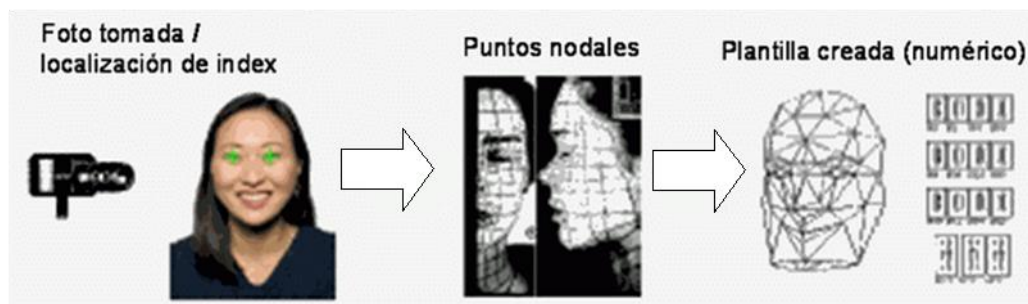


Figura 9: Extracción de Características de la Biometría Facial

Existen múltiples formas de realizar reconocimiento facial pero las más implantadas se basan en la obtención de características geométricas de la cara. Uno de los métodos más populares (Land Marks) se basa en la localización y forma de distintas características biométricas de la cara como la distancia entre los ojos o la anchura de la nariz. Los métodos basados en la correlación espacial establecen un indicador de parecido mediante la comparación de la correlación entre dos imágenes. Otros métodos, los descritos en [M. Padilla et al, 2003], se basan solo en el uso de características estáticas de la imagen (tamaño, color, forma...). Finalmente, la técnica de Eigen Faces codifica una imagen facial de tal forma que puede ser representada como la combinación lineal de otras imágenes [L. Sirovich et al, 1987]. Esta técnica puede ir aplicada con el algoritmo Hill-Climbing (descrito en [J. Galbally et al, 2009]) mejorando el resultado.

Mientras que las técnicas de correlación imponen un elevado coste computacional, Eigen Faces permite optimizar el sistema y aumenta la velocidad de procesamiento. Como desventaja resulta complicado reconocer una cara en una imagen tomada en diferentes puntos de vista, también es importante la iluminación con que es tomada la imagen y el fondo [Anil K. Jain et al, 2004a].

### 2.4.3 Huella Dactilar

El reconocimiento biométrico por huella dactilar es la tecnología más implantada y evolucionada. Aunque tienen un amplio abanico de uso, tradicionalmente se utiliza en la ciencia forense. Este tipo de tecnología posee una mayor aceptabilidad entre la población. Está basada en los estudios realizados por Faulds, Herchel y Galton durante el siglo XVIII. Estos métodos se basan en la comparación de unas características extraídas de las huellas llamadas minucias. Una huella dactilar está formada por un conjunto de crestas y valles característicos para cada individuo. Las minucias son el resultado del análisis de las crestas y valles que conforman la huella. Por tanto, para cada individuo solo se almacena su patrón de puntos de minucias. Previa a la identificación de usuarios se realiza una preclasificación de las huellas con el fin de reducir el número de búsquedas exhaustivas (diferente clasificación dependiendo del país, todas ellas recogidas en [F. M. Rodríguez et al, 2004]). La Figura 10, extraída de [F. M. Rodríguez et al, 2004], ofrece de manera abreviada el proceso de extracción de características.

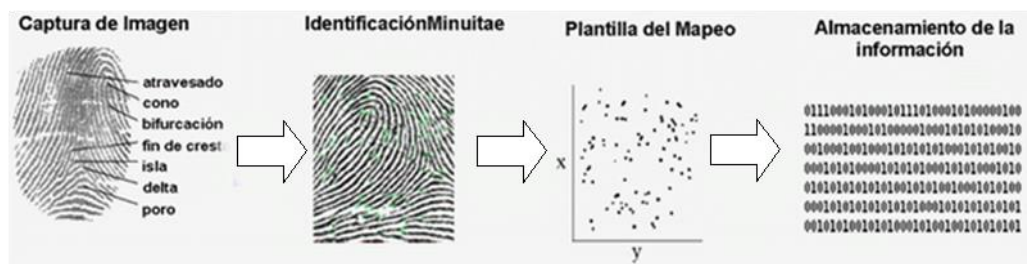


Figura 10: Extracción de Características de la Huella Dactilar

La extracción de características se basa en el estudio de una imagen en escala de grises. Para este estudio existen dos procedimientos: Por un lado, está el descrito en [K. C. Chan et al, 2003] para sistemas basados en correlación, donde se superponen dos imágenes de la huella dactilar y se calcula la correlación entre ambas para diferentes posiciones espaciales mediante rotación y translación de las mismas [Anil K. Jain et al, 2000]. Por otro lado, se encuentra el descrito en [N. Yager et al, 2004] para los sistemas basados en la extracción de puntos característicos. En ellos se segmenta la imagen (principalmente por filtros Gabor descritos en [F. M. Rodríguez et al, 2004] y DFT descrita en [Koichi Ito et al, 2005]) y se localizan las minucias (bifurcaciones, fines de cresta...) cuya disposición espacial ofrece una elevada capacidad de diferenciación. En ambos métodos, según [J. González et al, 2004], la decisión se toma mediante aproximaciones Bayesianas.

### 2.4.4 Iris

El iris es la región anular del ojo que se encuentra limitada por la pupila en la parte interior y la esclera o esclerótica por el exterior. Al igual que la huella, sus patrones se forman durante el desarrollo del feto y se estabilizan a los dos años. La compleja textura del iris ofrece una información muy discriminativa de cada individuo. Aunque es una tecnología moderna [J. Daugman, 1992] ya existen varias aproximaciones al problema del reconocimiento por iris. Todas ellas se basan en la extracción de características globales del iris o bien de ciertas regiones locales.

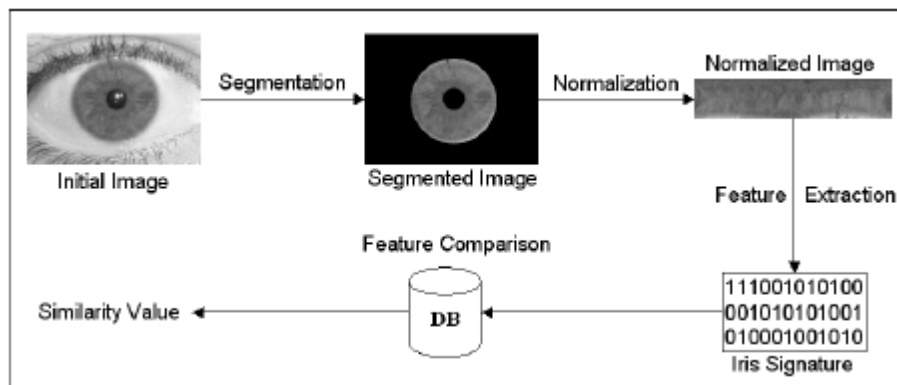


Figura 11: Fases del Proceso de Reconocimiento de Iris

El reconocimiento biométrico a través del iris según [Shrikanth Mohan, 2004] se realiza mediante una serie de pasos: la adquisición de la imagen, el pre-procesado de la misma, la localización del iris, normalización de la imagen, extracción de características y la comparación (Figura 11). Algunos sistemas, según [Hasan Demirel et al, 2009] utilizan los métodos de ajuste basados en modelos de histogramas. Estos procesan la imagen del iris mediante filtros paso-banda de dos dimensiones y utilizan el discriminante lineal de Fisher para realizar el proceso de comparación. Otros sistemas emplean la transformada de Hough para localizar el iris y mediante la transformada de Hilbert [C. L. Tisse, 2003] o con una Laplaciana piramidal [R. P. Wildes, 1994] obtienen las características de éste. Otros sistemas realizan un cambio a coordenadas polares y mediante el algoritmo del gradiente obtienen información discriminante de los límites del iris (radio y centroides). El último tipo de sistemas utilizan transformadas de Hough circulares para localizar el iris y filtros de Gabor para extraer el conjunto de características y realizan la comparación mediante el cálculo de la distancia de Hamming (HD) [C. L. Tisse et al, 2002].

### 2.4.5 DNA

El DNA (DeoxyriboNucleic Acid) podría ser el código único definitivo para cada individuo si no existiera la posibilidad de gemelos idénticos, los cuales poseen idénticos patrones de DNA. Es una característica unidimensional y es mayoritariamente usada en el contexto forense. Según [Anil K. Jain et al, 2004a] existen problemas que limitan el uso de esta biometría de aplicación. Por un lado, es muy fácil robar una muestra de DNA de un sujeto concreto y suplantar así su identidad. Para realizar una autenticación se deben de realizar complicados métodos químicos limitando su identificación en tiempo real. Por otro lado, no conviene que el código genético sea de dominio público.

### 2.4.6 Huella de la palma de la mano

La palma de la mano contiene una gran cantidad de crestas y valles al igual que las huellas dactilares. La zona de la palma de la mano es mucho mayor que la de los dedos por lo que puede ofrecer mayor potencial discriminativo. El reconocimiento biométrico basado en la huella de la palma de la mano es muy similar al de huella dactilar. Se apoya igualmente en la información local o global representada por los valles y crestas de la superficie de la palma, como bien recoge [José García-Hernández et al, 2005]. La Figura 12, extraída de [José García-Hernández et al, 2005], recoge de manera global la extracción de características descrito anteriormente. Por otro lado, también se puede obtener la información biométrica mediante el método utilizado por [Guangming Lu et al, 2002] llamado Eigen Palms y es similar al Eigen Faces (véase sección 2.4.2). Este método consiste en la representación de una imagen como combinación lineal de otras.

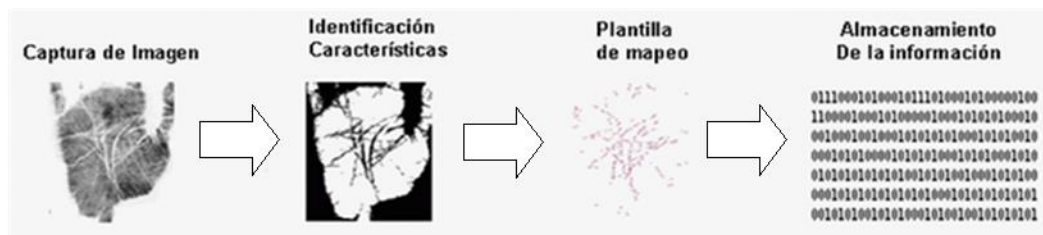


Figura 12: Extracción de Características de la Palma de la Mano



### 2.4.7 Dinámica del Tecleo

Es un tipo de rasgo biométrico conductual empleado en la verificación de la identidad de un individuo mediante su cadencia de escritura en un teclado. Esta tecnología se sostiene sobre la premisa de que cada individuo exhibe un patrón y una cadencia distintivos cuando pulsa una combinación de teclas [F. Monroe et al, 2000].

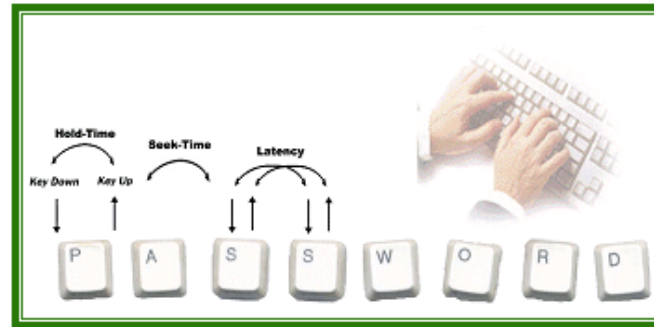


Figura 13: Características de la Dinámica del Tecleo, extraída de [F. Monroe et al, 2000]

En la mayoría de los sistemas se emplea la latencia (tiempo transcurrido entre la pulsación de dos teclas) entre pulsaciones como característica; sin embargo, como dice [O. Coltell et al, 2006] otros utilizan también el tiempo que permanecen la teclas presionadas. Esta tecnología no requiere hardware o dispositivos adicionales ya que se soporta sobre software de captura. Los principales sistemas utilizan clasificadores bayesianos, redes neuronales y lógica difusa.

### 2.4.8 Movimiento (caminar)

Es un tipo de rasgo biométrico conductual. Se emplea en la verificación de un individuo mediante un examen espacio temporal de su patrón de marcha (Figura 14). Los sistemas se pueden clasificar en estáticos (basados en la silueta humana [Jeff P. Foster et al, 2003]) o dinámicos (basados en el movimiento del individuo [Liang Wang et al, 2003]).



Figura 14: Extracción de Características de la Silueta

### 2.4.9 Voz

El reconocimiento biométrico mediante voz se basa en unas características tanto físicas del individuo como de comportamiento del mismo. Utilizando la clasificación de [D. A. Reynold et al, 1995] existen dos tipos de reconocimiento de locutor: unos, basados en el conocimiento de una frase preestablecida (text-dependent). Otros, no imponen ningún tipo de restricción de texto (text-independent). La Figura 15 muestra una visión general del proceso de extracción de características.

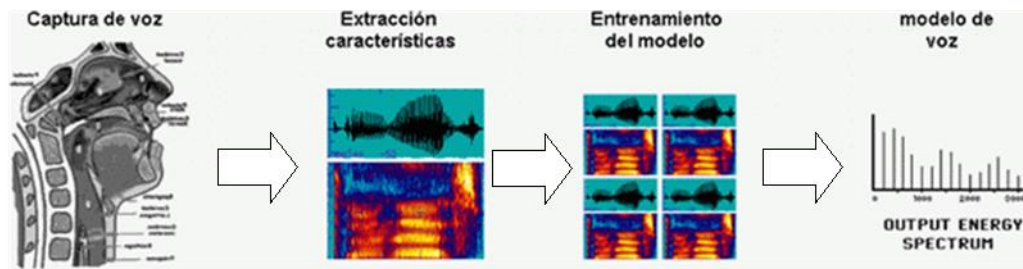


Figura 15: Extracción de las Características de la Voz

A lo largo de los años se ha ido mejorando la técnica para realizar el reconocimiento de locutores mediante la introducción de diferentes algoritmos [Dessimoz et al, 2005]. Se han ido utilizando los métodos probabilísticos para modelar el comportamiento de la señal de voz, entre ellos se encuentran el HMM y el GMM [D. A. Reynold et al, 1995]. Otros métodos de clasificación utilizados para el reconocimiento de locutor son NN [Anil K. Jain et al, 2004b], el SVM [M. Schmidt et al, 1996], DTW [M. Pandit et al, 1998] y el perceptrón multicapa (MLP) [C. E. Vivaracho et al, 2006]. O combinación de clasificadores y modeladores en la etapa de entrenamiento [Todor Ganchev et al. 2002].

## 2.5 BIOMETRÍA MULTIMODAL

Según [Anil K. Jain et al, 2004a] los sistemas biométricos que trabajan con un sólo rasgo biométrico (unimodales, véase sección 2.4) poseen las siguientes limitaciones:

- ✓ *Ruido*: La información capturada puede incorporar ruido o algún tipo de distorsión. Estos hechos son el resultado de la utilización de sensores defectuosos y de condiciones ambientales adversas.
- ✓ *Variación intra-clase*: Los datos capturados de un individuo en la operación de autenticación pueden ser muy diferentes de los empleados en la generación de su modelo durante la operación de alistamiento.
- ✓ *Variación inter-clase*: La escasa variación de un rasgo biométrico entre los sujetos de una población limita su capacidad de discriminación.
- ✓ *Universalidad*: Es posible que un determinado subconjunto de usuarios no posea un rasgo biométrico particular.
- ✓ *Ataques de impostores*: Un usuario ilegítimo puede falsificar el rasgo biométrico de otro legítimo para eludir el sistema.

Las limitaciones impuestas por los sistemas unimodales pueden atenuarse usando múltiples modalidades biométricas [L. Puente et al, 2008]. Por ejemplo, el requisito de universalidad (véase sección 2.1) ya no es imprescindible. El uso de múltiples modalidades biométricas también puede conseguir un aumento de la seguridad del sistema ante ataques de impostores. En definitiva, tal como describe [Hong et al, 1999] los sistemas biométricos multimodales utilizan varios rasgos biométricos para establecer el reconocimiento de individuos.

Según [Anil K. Jain et al, 2004a] los sistemas biométricos multimodales pueden operar en tres diferentes modos:

- ✓ *Serial Mode (Modo Consecutivo)*: La información de un rasgo biométrico es utilizado para reducir el número de posibles identidades antes de la entrada del nuevo rasgo biométrico.
- ✓ *Parallel Mode (Modo Paralelo)*: La información de múltiples rasgos biométricos es utilizada para realizar el reconocimiento.
- ✓ *Hierarchical Mode (Modo Jerárquico)*: En este modo, se combinan clasificadores individuales en una estructura de árbol.

Estos sistemas biométricos presentan información tomada de múltiples indicadores biométricos. Para conseguir el funcionamiento del sistema en los modos descritos

anteriormente se deben combinar los sistemas biométricos (o fusionar). Existen tres niveles de fusión [Anil K. Jain et al, 2004a]:

- ✓ *Fusión a nivel de extracción de características:* En este nivel se combinan las características extraídas de los rasgos biométricos en un único vector.
- ✓ *Fusión a nivel de puntuaciones:* En este nivel se combinan las puntuaciones que describen las similitudes entre los datos adquiridos y los modelos de cada sistema. Este modo de fusión requiere que las puntuaciones de los subsistemas se encuentren normalizadas en un dominio común [Anil K. Jain et al, 2005]. La fusión a este nivel es la preferida debido a la facilidad de acceso y de combinación que ofrecen las puntuaciones.
- ✓ *Fusión a nivel de decisiones:* En este nivel se combinan las decisiones tomadas por cada sistema con el fin de obtener la decisión final.

Los sistemas biométricos multimodales están diseñados para trabajar de cinco formas diferentes (Figura 16) y aunque algunas de ellas no implican múltiples modalidades, implican la fusión de información en algunos puntos [Anil K. Jain et al, 2004a]. Los esquemas de operación son los siguientes:

- ✓ *Un rasgo, múltiples sensores:* Se adquiere un único rasgo mediante diferentes sensores y se combina para mejorar el proceso de reconocimiento.
- ✓ *Múltiples rasgos:* Se obtienen y combinan diferentes características biométricas de una persona.
- ✓ *Un rasgo, múltiples unidades:* Se adquieren y combinan diferentes unidades del mismo rasgo.
- ✓ *Un rasgo, múltiples representaciones:* Se adquieren múltiples tomas de un mismo rasgo por el mismo sensor.
- ✓ *Un rasgo, múltiples clasificadores:* Se obtiene un rasgo a través de un único sensor y mediante diferentes técnicas de extracción de características y/o de comparación de patrones se realiza el proceso de reconocimiento.

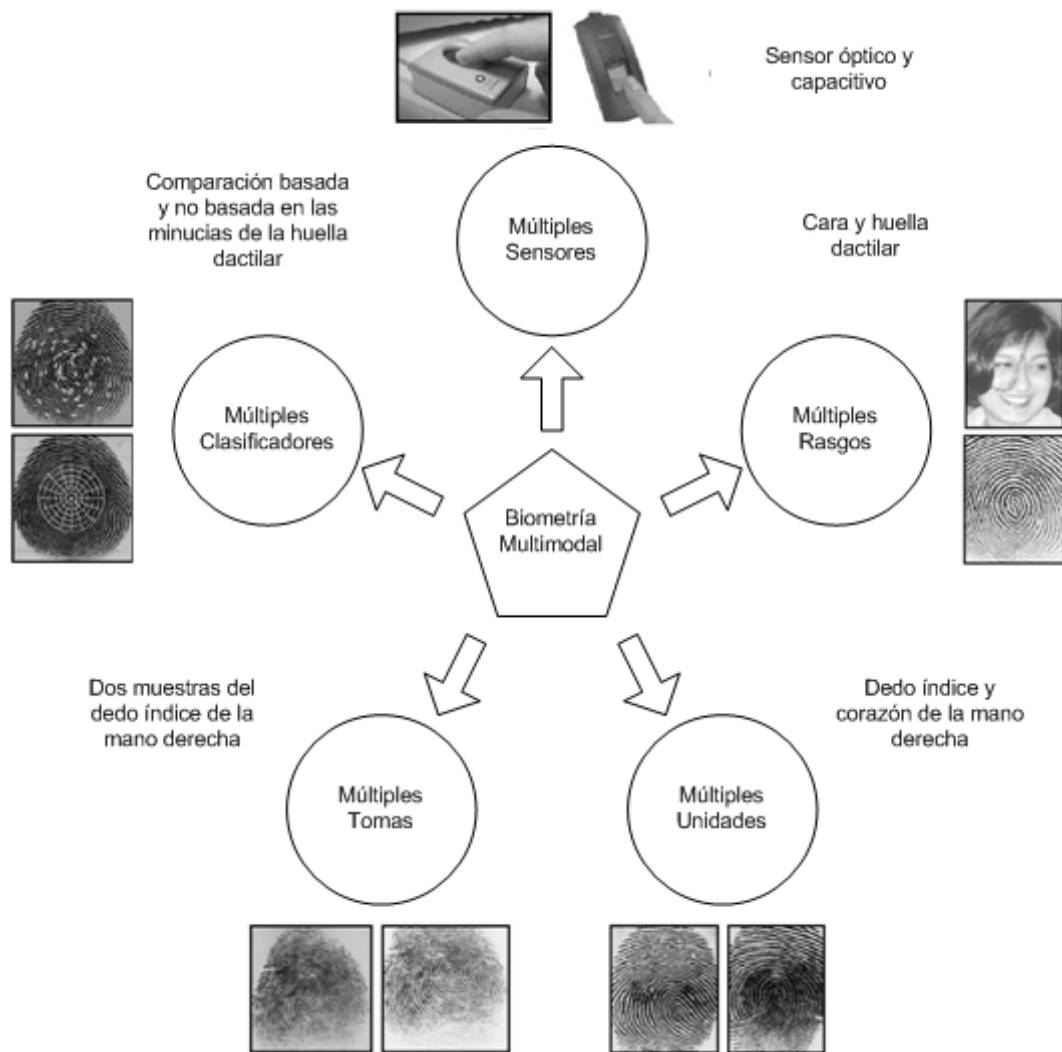


Figura 16: Varios Escenarios de un Sistema Biométrico Multimodal (extraído de [Anil K. Jain et al, 2004a])

# CAPÍTULO 3

---

## RECONOCIMIENTO FACIAL

En este capítulo, se presenta al lector un acercamiento en el proceso de reconocimiento facial y la manera de resolverlo en la actualidad. Se hablará de las diferentes formas de extracción de características, el proceso de detección de rostros y finalmente la toma de decisiones.

### 3.1 INTRODUCCIÓN

Los seres humanos poseen una alta capacidad para reconocer rostros aun en escenarios donde existan altos niveles de variabilidad. La cara es el rasgo biométrico más utilizado por la mayoría de las personas para reconocerse entre sus semejantes. Aunque su característica de unicidad es menor en comparación con la huella dactilar o el iris, ha sido utilizado desde los orígenes de la humanidad para establecer una sociedad.

El reconocimiento facial es el método de reconocimiento más antiguo, aunque no se empezó su desarrollo e investigación hasta comienzos de los años 60. Recientemente esta técnica está recibiendo especial interés, especialmente durante los últimos años con la aparición de los últimos smartphones. Todo este auge queda recogido en la aparición de conferencias sobre reconocimiento facial tales como AFRG [1] y AVBPA [2]. Por otro lado, el incremento de todas las aplicaciones tecnológicas donde se puede aplicar esta

técnica ha aumentado considerablemente durante los últimos 30 años de investigación, desde intereses gubernamentales, militares, privados o comerciales, hasta las aplicaciones forenses o policiales.

En nuestra vida diaria estamos obligados a memorizar una gran cantidad de contraseñas, PIN... necesarios para sacar dinero de un cajero o encender el ordenador. Todo lo memorizado puede ser olvidado con facilidad, por lo que tendemos a repetir los patrones y finalmente utilizamos el mismo password para casi todo. Aun así, el ser humano es olvidadizo por naturaleza, por lo que finalmente se acaban perdiendo u olvidando ¿Por qué utilizar passwords cuando nuestro cuerpo nos ofrece unos unívocos?

En la actualidad existen multitud de métodos fiables de identificación biométrica personal, por ejemplo, la huella dactilar (implementada ya en smartphones), el iris o la retina, todos ellos con una elevada tasa de acierto. Todos ellos necesitan de la cooperación del participante para la extracción de datos, lo cual puede resultar intrusivo. Por el contrario, en el reconocimiento facial no se necesita la acción directa del sujeto para la obtención de datos, está calificado como un método no intrusivo, de ahí su alta aceptación en la sociedad.

La identificación de características faciales [R. and Poggio et al, 1993] ha recibido un fuerte impulso gracias al avance de la tecnología de video multimedia desencadenando así un aumento de cámaras en los lugares de trabajo, hogar y dispositivos móviles. Hoy en día, el reconocimiento facial se puede aplicar en el control de accesos a edificios públicos y privados, cajeros automáticos, laboratorios de investigación, como clave secreta de acceso a ordenadores de carácter personal o terminales móviles de última generación.

Los sistemas biométricos se basan en el reconocimiento de patrones, tal y como se dijo en capítulos anteriores. Un patrón biométrico este compuesto por un conjunto de características fisiológicas y psicológicas que definen un vector asociado unívocamente a una persona. Entonces, utilizando esta información es posible verificar e identificar un vector de características dentro de una base de datos con información de otros individuos.

El proceso de identificación facial o reconocimiento de cara involucra dos etapas: la detección y el reconocimiento [W. Zhao et al, 2003]. En la primera etapa se determina si en la imagen o secuencia de imágenes existen caras, en cuyo caso se procede a la segmentación de las mismas y de la imagen de fondo. En la segunda etapa se extraen las características de las caras obtenidas en la etapa anterior y se procede a la identificación del individuo que se desea reconocer. Ambos procesos no deberían ser totalmente independientes debido a que según la forma en que se detecte una cara puede ser prácticamente imposible su reconocimiento con caras de una base de datos detectadas de manera diferente. De ahí que los sistemas de reconocimiento facial estén fuertemente condicionados por la posición y orientación de la cara del sujeto con respecto a la cámara y las condiciones de iluminación en el momento de realizar la detección.

A continuación, pasaremos a hablar sobre cada una de las etapas más a fondo y de los algoritmos más utilizados en cada una de ellas.

## 3.2 DETECCIÓN DE ROSTROS

La detección de rostros consiste en saber si existe alguna cara en una imagen o secuencia de imágenes dada, además de la localización de las coordenadas de dichos rostros. La localización del rostro es una parte fundamental en el proceso de reconocimiento. La precisión con que se detecten las coordenadas del rostro influye posteriormente en el rendimiento del reconocedor. Puesto que una buena detección de rostro influye en una buena extracción de características que facilitarían el reconocimiento del rostro posteriormente [K.J. Kirchberg et al, 2002].

Entre los principales problemas a los cuales se enfrenta la detección de rostros están: la pose y orientación, tamaño de la cara, expresión de la cara, presencia de componentes estructurales (barba, gorro, gafas...), oclusiones parciales, problemas de iluminación, cantidad desconocida de caras en la imagen, ruido en el proceso de adquisición de las imágenes y pérdida de fidelidad por los métodos de compresión [Xiaoguang Lu, 2005].

La detección de rostros ha avanzado considerablemente en la última década por lo que el estado del arte es sumamente amplio. En [M. H. Yang et al, 2002] se presenta una revisión general de algunos de los métodos más relevantes, así como una clasificación de ellos:

- **Métodos basados en conocimientos:** Se basan en obtener un modelo a partir del conocimiento humano de las características que conforman el rostro. Este conocimiento se usa para establecer reglas que permitan distinguir entre rostros. Por lo que el funcionamiento del método dependerá del conocimiento del investigador y de su experiencia para delimitar reglas. Por ejemplo, la técnica top-down descrita en [G. Yang and T.S. Huang, 1994].
- **Métodos basados en características invariantes:** Se buscan las características del rostro que son persistentes o invariantes a un gran número de situaciones complejas, a diferentes condiciones de iluminación y pose. Por ejemplo, el color de piel, la textura, los bordes.
- **Métodos de correspondencia de plantillas:** Se utilizan varios patrones del rostro que describen completa o parcialmente sus características. Para detectar un rostro se calcula la correlación entre una imagen de entrada y los patrones con los que se cuenta. Se distinguen dos tipos: los que manejan patrones predefinidos y los que permiten patrones deformables.

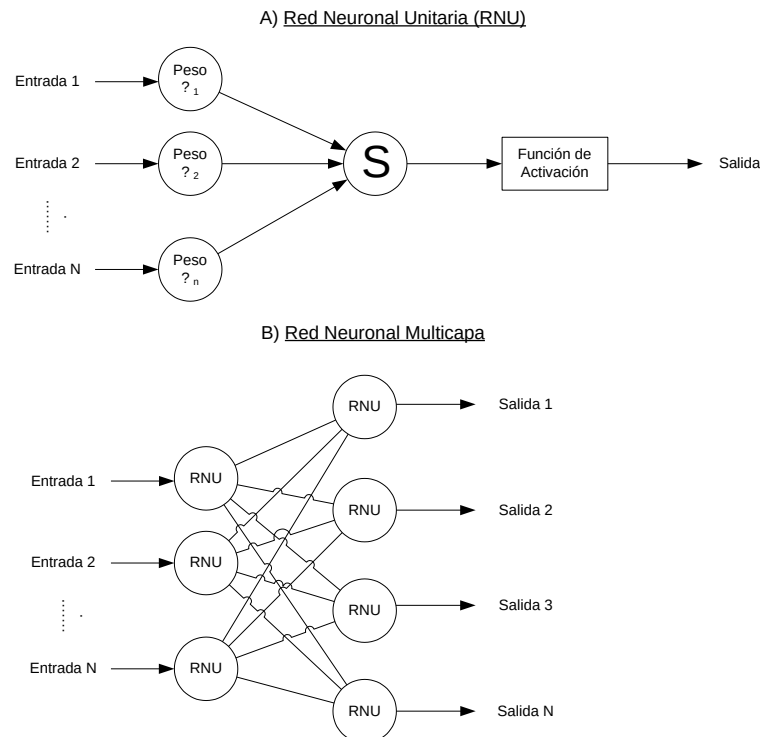


1. *Patrones predefinidos:* El más utilizado es el uso de operadores de bordes, se crean modelos de las imágenes de bordes. Se puede dividir el rostro en bloques, donde cada uno tiene información de luminosidad y nivel de bordes del mismo.
  2. *Patrones deformables:* En este caso se permite cierta flexibilidad en el ajuste del modelo de rostro, para ello se define una función de energía que mide el grado de ajuste en cada momento.
- **Métodos basados en la apariencia:** Se crean patrones o modelos a partir de un conjunto de imágenes de entrenamiento tomando valores de sus píxeles. Por lo general, están divididos en las siguientes etapas:
    1. *Búsqueda multiescala:* A partir de la imagen de entrada, se obtiene una pirámide de imágenes con diferentes resoluciones.
    2. *Pre-procesamiento:* Descartar regiones en las que no existe una cara y mejorar las imágenes de las ventanas de cara a la etapa posterior.
    3. *Clasificación:* Se realiza una clasificación de rostro y no rostro. Para resolverlo se han aplicado distintos métodos de clasificación: Redes Neuronales, discriminantes de Fisher, máquinas de vectores de soporte (SVM), combinación de clasificadores elementales (AdaBoost)...
    4. *Agrupación de candidatos y post-procesamiento:* Al finalizar el proceso se obtiene una división de muchas regiones las cuales pueden estar asociadas a rostros o ser falsas detecciones. En este paso, se agrupan las regiones con un alto solapamiento y se eliminan los candidatos menos fiables.

En los apartados siguientes se hablará de los algoritmos más utilizados en la detección de rostros. Se hace especial hincapié en el algoritmo Viola-Jones debido a su implementación en el proyecto.

### 3.2.1 Neural Networks (NN)

Este clasificador se encuentra dentro de los clasificadores supervisados. La esencia principal del aprendizaje supervisado responde a la existencia de una serie de muestras de entrenamiento, véase [K. Fukushima, 1988]. Estas muestras atraviesan sucesivamente la red neuronal, donde el vector de coeficientes cepstrales se multiplica por los respectivos pesos y los resultados se suman y se evalúan mediante una función de activación del tipo sigmoideal, escalón u otra [A. V. Oppenheim et al, 1998]. La salida de la red neuronal simboliza la pertenencia a una clase u otra. Los valores tomados a la salida pueden variar dependiendo de si es binario (1, -1) o multiclase (1, 0, -1...).



**Figura 17: Diagrama de Red Neuronal Unitaria y Multicapa**

Las muestras de entrenamiento son pasadas por la red neuronal forzando a un valor de salida. En el caso de una red unitaria, si se obtiene en la salida un valor diferente a lo esperado se vuelven a calcular los pesos y se vuelve a entrenar (Figura 17a). Por el contrario, si la red es multicapa se vuelven a calcular los pesos de los nodos de la red neuronal teniendo en cuenta el error del nodo anterior (Figura 17b). Las redes unitarias están formadas por un único perceptrón (MLP), mientras que las redes multicapa se forman con múltiples perceptrones anidados (la salida de uno es la entrada de otro).

La red neuronal multicapa modela una línea de decisión de las clases entrenadas, lo que conlleva que para incorporar una nueva clase se deba entrenar nuevamente toda la red. En el test, el vector de características de la voz desconocida se somete al paso de la red ya entrenada (con los pesos determinados) y a la salida se asociará con alguna de las clases.

### 3.2.2 Support Vector Machines (SVM)

Este método es utilizado para la resolución de problemas de clasificación, regresión y estimación. Está basado en la teoría del aprendizaje descrita en [V. Vapnik, 1995a]. Según [T. Marwala et al, 2005] el objetivo principal es transformar los vectores de entrada (los vectores de características) en otros vectores de dimensión más alta en los que el problema pueda solucionarse linealmente.

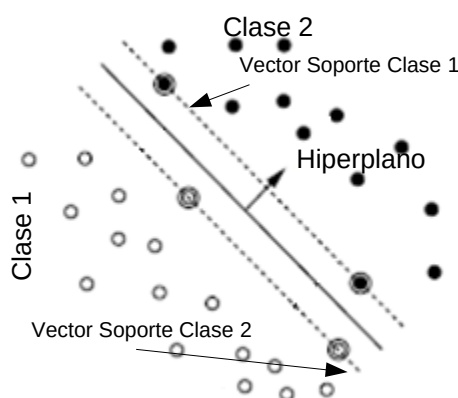


Figura 18: Hiperplano Separador de Clases

La frontera de decisión lineal corresponde a un hiperplano/s separador de la clase/s óptimo/s ([Pai-Hsuen Chen et al, 2003]). El hiperplano elegido es el que maximiza la distancia euclídea a los vectores de características más cercanos de cada clase (Figura 18). Estos vectores se denominan vectores soporte. Cuando las clases no son linealmente separables, se busca el hiperplano que minimice los errores a través de una función de coste. La extensión a límites no lineales se realiza a través de un núcleo (kernel) que satisfaga las condiciones de Mercer recogidas en [J. Mercer, 1909]. Existen distintos tipos de kernel (gaussiano, polinomial, perceptrón, sigmoide)<sup>78</sup> dependiendo de la aproximación que se quiera realizar, en el proyecto se utilizó concretamente uno del tipo gaussiano, que a su vez es el más utilizado según [Baudat G. et al, 2001]. Básicamente, cada vector de características es mapeado a un espacio de alta dimensionalidad en que los datos son separables linealmente. El producto escalar de los vectores transformados se puede escribir como el núcleo, por lo que se necesita trabajar en el espacio extendido.

SVM trabaja en bloque, por lo tanto, un cambio en los datos supone un nuevo entrenamiento. En la etapa de test, SVM obtiene una puntuación que indica cuantos vectores de características de la muestra desconocida pertenecen a una clase y cuantos pertenecen a la otra. También puede devolver directamente la clase a la que pertenece.

<sup>7</sup> [www.kernel-machines.org](http://www.kernel-machines.org)

<sup>8</sup> [www.support-vector.net](http://www.support-vector.net)

### 3.2.3 Algoritmo Viola-Jones

El algoritmo Viola-Jones supone un gran avance dentro de los algoritmos de detección por su bajo coste computacional y su gran rapidez. Este algoritmo permite la detección de rostros en tiempo real de forma robusta y precisa. Todo motivado por que la clasificación se realiza mediante características en vez de pixel a pixel. Este algoritmo está constituido por tres componentes esenciales: Haar-Like feature, Integral Image y AdaBoost [P. Viola and M.J. Jones, 2001].

#### Haar-like feature

Las características Haar (Haar-like feature) son características digitales utilizadas en reconocimiento de objetos. Estas características permiten obtener información de una zona concreta mediante una operación aritmética simple consiguiendo así una gran eficiencia tanto de cálculo como espacial. En imágenes digitales, las características Haar son definidas por la diferencia entre la suma de todos los píxeles de dos regiones.

Las características usadas en el algoritmo son las mismas utilizadas por [Papageorgiou et al, 1998] con la salvedad de que no se trabaja sobre imágenes de intensidad. En dicho estudio se realiza la detección de objetos basándose en una representación wavelet de un objeto. En otras palabras, se aprende una clase de objeto en términos de un subconjunto de características Haar básicas.

Inicialmente se usaron tres características Haar (**¡Error! No se encuentra el origen de la referencia.**):

- *Two-Rectangle feature*: Su valor se calcula con la diferencia entre la suma de los píxeles dentro de las dos regiones rectangulares.
- *Three-Rectangle feature*: Su valor se calcula con la suma de los píxeles dentro de las dos regiones rectangulares exteriores y la resta de los píxeles de la región interior.
- *Four-Rectangle feature*: Su valor se calcula con la diferencia entre las diagonales de los pares de rectángulos.

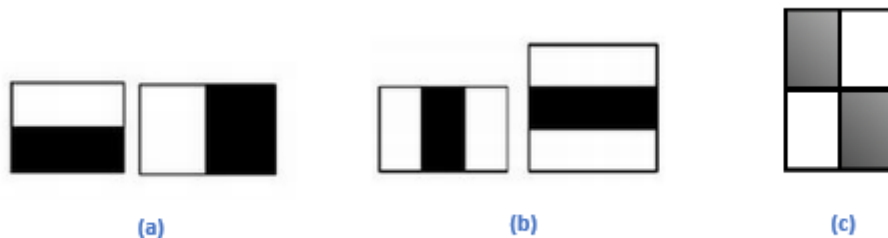


Figura 19: Haars-like features: (a) Two-Rectangle feature, (b) Three-Rectangle Feature, (c) Four-Rectangle Feature, extraído de [P. Viola and M.J. Jones, 2001]

Posteriormente, [R. Lienhart and J. Maydt, 2002] extendieron las características Haar, agregando características rotadas en 45°. Estas últimas son las utilizadas en el detector de objetos de OpenCV (Figura 20: Haar-like features extendidas por Lienhart y Maydt).

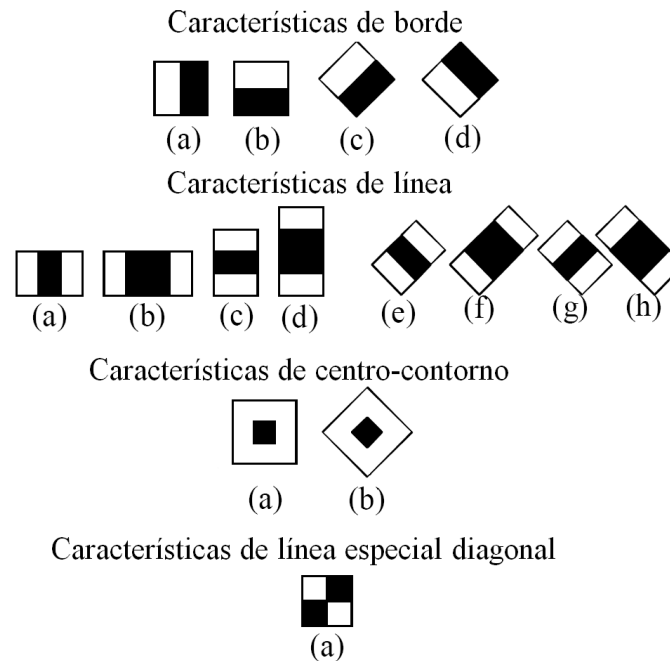


Figura 20: Haar-like features extendidas por Lienhart y Maydt

### Integral Image

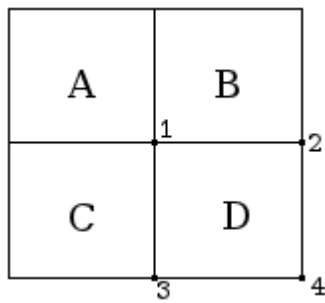
La imagen integral (integral image) es la aportación más importante del trabajo de Viola-Jones, consiste en la representación de imágenes para agilizar el cálculo de las características Haar. La integral de una imagen con respecto a un punto (x,y) consiste en la suma de todos los píxeles arriba y a la izquierda de dichos puntos (x,y) incluidos.

$$ii(x, y) = \sum_{\substack{x' \leq x \\ y' \leq y}} i(x', y')$$

Ecuación 1

Donde  $ii(x,y)$  es la imagen integral e  $i(x,y)$  la imagen original.

La imagen integral permite extraer de forma rápida características a diferentes escalas ya que se trabaja con una imagen acumulativa construida a partir de operaciones básicas, en lugar de trabajar directamente con los valores de intensidad. Así, con este método se logran obtener los valores requeridos para el cálculo de la sumatorio de una región con tan solo tres o cuatro operaciones básicas.



Veamos un ejemplo:

Para obtener el valor de la región D se suma el valor del pixel 4 con el valor del pixel 1 y se resta la suma del pixel 2 con el 3:

$$D = (4 + 1) - (2 + 3)$$

Figura 21: Cálculo de la región D con valores de imagen integral

Por lo que se requieren 6 referencias a la imagen integral para calcular el valor de una característica Haar con dos regiones adyacentes.

### AdaBoost

El algoritmo AdaBoost (Adaptative Boost) es un algoritmo de aprendizaje máquina (boosting) que combina varios clasificadores básicos o débiles de forma secuencial para formar un único clasificador más complejo y preciso.

Dicho algoritmo fue introducido por [Y. Freund and R.E. Schapire, 1995], inicialmente a todas las muestras de entrenamiento se les asigna el mismo peso; sin embargo, cada vez que se genera un clasificador estos pesos se cambian favoreciendo a las muestras mal clasificadas. De esta forma se brinda la capacidad para utilizar el nivel de error de cada algoritmo y poner mayor atención a dichas muestras.

El algoritmo de Viola-Jones utiliza un árbol de decisión en cascada, donde cada nodo es un clasificador obtenido con el algoritmo AdaBoost. En este árbol binario de decisión, el primer clasificador es un clasificador muy simple pero rápido y el último es más complejo pero exacto [Figura 22: Clasificador en cascada]. El primer nodo permite el error máximo que se ha permitido en la detección y a medida que se avanza en el árbol, el criterio de descarte es mucho más estricto. De esta forma se permite descartar en un primer momento las muestras más sencillas de detectar y aplicar un mayor esfuerzo en aquellas cuya clasificación sea más dudosa.

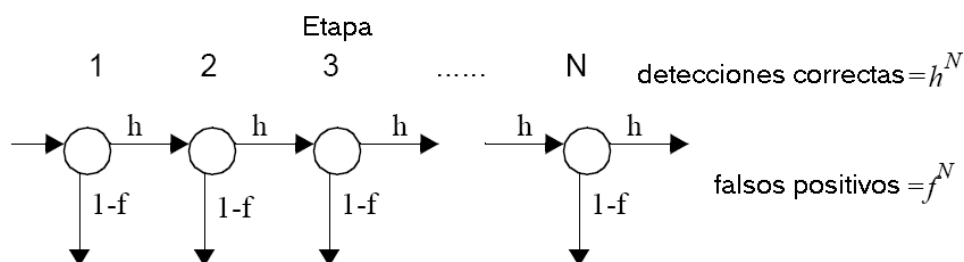


Figura 22: Clasificador en cascada

### 3.3 RECONOCIMIENTO DE ROSTROS

El reconocimiento de rostros en sistemas biométricos consiste en la identificación o verificación de una o más personas en una imagen estática o secuencia de imágenes. Para poder llevar a cabo dicha identificación/verificación el rostro de la persona debe haber sido almacenado y etiquetado previamente en una base de datos. Últimamente, el reconocimiento facial es un tema muy recurrente y en pleno alza con la aparición de los smartphones, donde se ha introducido el desbloqueo del teléfono con reconocimiento facial.

En el ser humano existen estudios psicológicos que indican que el reconocimiento del rostro es un proceso dedicado en el cerebro y difiere del reconocimiento de otros objetos. Así mismo, nuestro cerebro percibe con mayor magnitud ciertas partes del rostro, en especial la parte superior (cerca de la nariz), recordando con mayor precisión los rostros que son considerados atractivos [W.Zhao et al, 2003].

Por identificación se entiende el proceso de asociar una única etiqueta a un individuo. Donde varios factores se deben considerar, uno de los principales es decidir si se trabaja con imágenes estáticas, o bien con video.

Existen tres corrientes o métodos diferentes en la comunidad biométrica [Xiaoguang Lu, 2005]:

- **Métodos basados en la apariencia (Heurísticos):** Una imagen es considerada como un vector de alta dimensionalidad. Por lo que se utilizan diferentes imágenes (perspectivas diferentes) para generar un vector de características para un individuo.
- **Métodos basados en el modelo:** Consiste en generar un modelo 2D o 3D del rostro humano, el cual es capaz de capturar las variaciones en la cara. Donde se establecen puntos de control que identifican posiciones específicas del rostro
- **Métodos basados en características invariantes:** Se utiliza la información intrínseca de la imagen (color, textura, forma...). El objetivo es detectar patrones en la imagen para segmentar los rostros contenidos. Muchas veces es necesario transformar la imagen a otros espacios como YCbCr o HSV.

Evidentemente, se pueden combinar los diferentes métodos para obtener mejores resultados. En estos casos se llaman métodos híbridos. En la [Figura 23] se puede observar un esquema de los diferentes métodos.



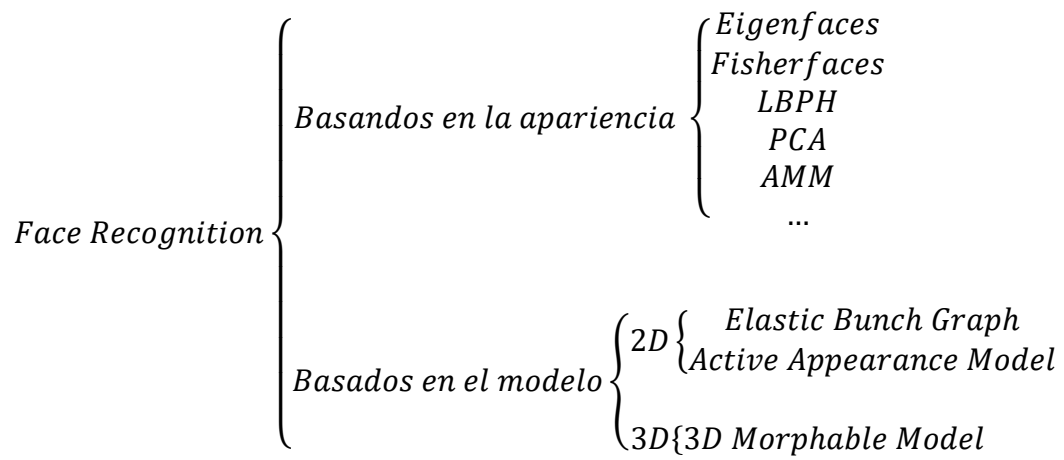


Figura 23: Principales métodos en reconocimiento facial

En los sucesivos apartados se hablará de los métodos basados en apariencia utilizados en el desarrollo del proyecto.

### 3.3.1 EigenFaces

Es una técnica de extracción de características basadas en la apariencia, el trabajo más representativo fue el presentado por [M. Turk and A. Pentland, 1991]. *EigeFaces* permite determinar, mediante ortogonalidad dimensional, qué vectores ofrecen más información a un conjunto de datos de dimensión  $N$ . La generación de estos vectores implica un gasto computacional bastante elevado, por lo tanto, para la obtención de la información  $N$ -dimensional se aplica análisis de componentes principales (PCA) [Moon H. and Phillips J, 1998], el cual toma una cantidad menor de vectores por imagen, pero con la información necesaria para la reconstrucción de los rostros. Por el contrario, PCA retiene alguna información no deseada debida a la iluminación y a la expresión facial, ya que ambas no las tiene en cuenta.

Cuando se proyecta la imagen de un rostro sobre el subespacio creado con PCA (las *eigenfaces*), la imagen no cambia mucho; pero cuando se proyectan imágenes sin rostros, estas varían considerablemente. Por tanto, para realizar la detección de rostros se calcula la distancia entre la región a evaluar y el subespacio de rostros. La distancia será menor para regiones con rostros, que para aquellas en las que no se encuentre ninguno.

Los *eigenfaces* son un conjunto de vectores que conforman el mapa de las variaciones entre imágenes. Estos vectores son el resultado de aplicar PCA a la matriz de covarianzas de un conjunto de imágenes, los cuales se llaman *eigenvectors*, siendo la imagen tratada como un vector en el espacio multidimensional. Además, cada cara individual puede ser representada exactamente en términos de una combinación lineal de las *eigenfaces*, la cual puede ser aproximada utilizando solo las mejores *eigenfaces*. Por tanto, los  $M$  mejores *eigenfaces* conforman un subespacio  $M$ -dimensional de las caras de todas las posibles imágenes.

El reconocimiento mediante *Eigenfaces* realiza las siguientes operaciones:

1. Adquisición de una serie de imágenes de rostros iniciales.
2. Calculo de los *eigenfaces* del conjunto de entrenamiento, almacenando únicamente las  $M$  dimensiones que correspondan con los mayores eigenvalores.
3. Calculo de un conjunto de pesos basados en la imagen de entrada y las  $M$ -*eigenfaces* mediante la proyección de la imagen de entrada sobre cada una de las *eigenfaces*.
4. Determinar si la imagen de la cara pertenece o no al conjunto de entrenamiento, por medio de la distancia euclídea.

Para el desarrollo matemático se define una imagen  $I(x,y)$  como una matriz bidimensional de  $N \times N$ , de 8 bits y valores de intensidad de 0 a 255 (correspondientes a una imagen en escala de grises). Cada vector es de longitud  $N_2$  y es una combinación lineal de la imagen de una cara original.

El conjunto de imágenes de caras para la etapa de entrenamiento es  $\Gamma_1, \Gamma_2, \Gamma_3, \dots, \Gamma_M$ . A continuación, se obtiene la cara promedio:

$$\psi = \frac{1}{M} \sum_{n=1}^M \Gamma_n$$

(Ecuación 2)

Se subtrae el promedio de cada rostro:

$$\Phi_i = \Gamma_i - \psi$$

(Ecuación 3)

A continuación, se calcula la matriz de covarianza:

$$C = \frac{1}{M} \sum_{n=1}^M \Phi_n \Phi_n^T = AA^T$$

(Ecuación 4)

Donde A es la matriz  $A = [\Phi_1, \Phi_2, \dots, \Phi_n]$  y la matriz C es de dimensión  $N_2 \times N_2$ .

Se calculan los *eigenvectors*  $\mu_i$  de  $C = AA^T$ , con ayuda de los autovalores de  $v_i$  y solo nos quedamos con los K *eigenvector* con valores más elevados.

$$\mu_i = Av_i$$

(Ecuación 5)

Para realizar el reconocimiento de la imagen de entrada, se compara ésta con la imagen media y se multiplica su diferencia por cada vector propio de la matriz L:

$$\omega_k = u_k^T (\Gamma - \Psi) \quad \text{para } k = 1, \dots, M$$

(Ecuación 6)

Con esto se obtiene un conjunto de pesos que conforman el vector  $\Omega^T = [\omega_1, \omega_2, \dots, \omega_M]$  que describe la contribución de cada *eigenface* en la representación de la imagen de entrada [Figura 24: Eigenfaces]. Por lo que este vector puede ser utilizado para evaluar la distancia entre vectores y mediante umbralización, determinar si la imagen de entrada puede considerarse que pertenece al espacio de imágenes de entrenamiento. Este valor será el que minimiza la distancia euclidiana:

$$\varepsilon_k^2 = \underset{\Omega_k}{\operatorname{argmin}} \|\Omega - \Omega_k\|$$

(Ecuación 7)

Donde  $\Omega_k$  es el vector que describe la k-ésima cara.

Como todo método de reconocimiento tiene ciertas limitaciones, en este caso si la imagen esta girada, cambios en la iluminación, variación del fondo de la imagen... todo ello repercute en el reconocimiento del rostro.

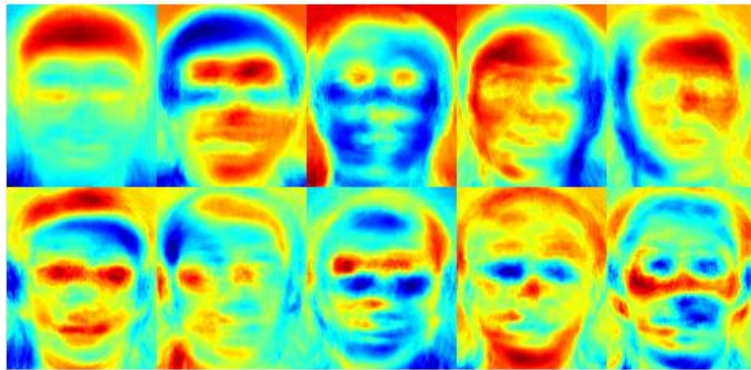


Figura 24: Eigenfaces

### 3.3.2 FisherFaces

Al contrario que *EigenFaces*, el método de reconocimiento de rostros basado en *Fisherface* tiene en cuenta la intensidad de luz y las expresiones faciales. Este método se encarga de clasificar y reducir las dimensiones de las caras usando el método FLD (Fisher's Linear Discriminant) [R. Fisher, 1936]. Por lo que el análisis de Fisher intenta proyectar los datos de manera que su nueva dispersión sea la óptima para la clasificación.

Mientras PCA reduce los datos buscando los vectores que mejor describen los mismos, LDA (Discriminant Lineal Analysis) [M. Welling, 2009] busca los vectores que proporcionan mejor discriminación entre clases después de la proyección, con el fin de que la separabilidad entre clases sea lo mayor posible.

Por tanto, *Fisherfaces* realiza un LDA sobre la clasificación de las imágenes de entrenamiento buscando aprovechar la información disponible para buscar una proyección que maximice la separación entre imágenes de diferentes personas y minimice la distancia entre imágenes de una misma clase. Realiza una agrupación entre clases como bien explican en [P. N. Belhumeur et al, 1997].

Con respecto a la formulación, se definen dos matrices: de varianza entre clases (imágenes de personas distintas) y de varianza dentro de la clase (imágenes de una misma persona).

$$S_B = \sum_{i=1}^c |X_i|(\mu_i - \mu)(\mu_i - \mu)^T; \quad \text{matriz de varianza entre clases}$$

(Ecuación 8)

$$S_W = \sum_{i=1}^c \sum_{X_k \in X_i} (X_k - \mu_i)(X_k - \mu_i)^T; \quad \text{matriz de varianza dentro de clase}$$

(Ecuación 9)

Donde  $\mu_i$  corresponde a la imagen promedio de la clase  $X_i$ ,  $|X_i|$  al numero de puntos dentro de la clase  $X_i$  y  $\mu$  es el promedio de todas las imágenes.

Luego se debe encontrar la matriz de proyección  $W \in R^{(m \times n)}$  resolviendo el problema de optimización, dando lugar a la siguiente ecuación:

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

(Ecuación 10)

Para esto se debe reducir la dimensionalidad de las imágenes a no más de un número de imágenes – número de clases. Luego se puede aplicar LDA para reducir a número de clases menos un elemento y agruparlas según la clase a la que pertenece. Si la primera

reducción se realiza utilizando *Eigenfaces*, al método se le llama *Fisherfaces* [Figura 25: Fisherfaces].

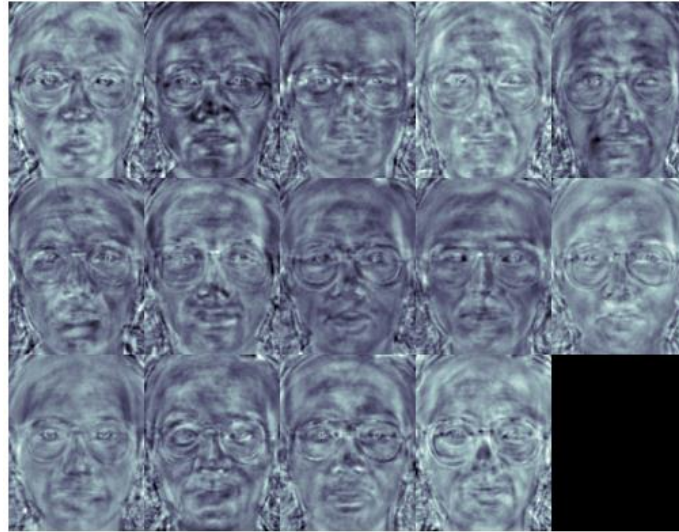


Figura 25: Fisherfaces

Estos métodos han mostrado gran eficiencia y popularidad debido a su relativa simpleza. En contra tienen la cantidad de cálculos y elevados requerimientos de memoria debido a la dimensionalidad de las imágenes.

### 3.3.3 Local Binary Patterns Histograms (LBPH)

Inicialmente fue diseñado para la descripción de texturas [T. Ahonen et al, 2004]. Los descriptores de texturas aplicados localmente en algunas regiones del rostro aportan más información que otras. Por lo que los descriptores de texturas tienden a promediar la información que describen, lo cual no es conveniente al describir rostros debido a que mantener la información de las relaciones espaciales es importante.

Para formar la descripción global, la imagen del rostro es dividida en diferentes regiones, a cada una se le aplica un histograma con el que se obtiene el operador LBPH que describe información independiente por región. Cada una de estas descripciones locales son entonces concatenadas para construir una descripción global del rostro.

En resumen, el método LBPH asigna etiquetas a cada uno de los píxeles de la imagen tomando en cuenta la distribución de los vecinos. Para el reconocimiento de imágenes sigue los siguientes pasos.

- La imagen es recorrida iterativamente por una máscara de tamaño determinado (8x8) seleccionando cada vez un píxel central y sus vecinos.
- El píxel central se compara con cada uno de sus vecinos de forma ordenada. De forma que se asigna un 1 cada vez que el píxel central sea menor que el píxel comparado y un 0 en el caso contrario.

$$LBP(x_c, y_c) = \sum_{p=0}^{p-1} 2^p s(i_p - i_c)$$

(Ecuación 11)

Con  $(x_c, y_c)$  como el píxel central con intensidad  $i_c$ ;  $i_p$  la intensidad del píxel vecino;  $s(x)$  como la función señal definida por:

$$s(x) = \begin{cases} 1, & \text{si } x \geq 0 \\ 0, & \text{otro valor} \end{cases}$$

(Ecuación 12)

- El número binario resultante [Figura 26: Obtención de los parámetros LBP] se convierte a un número decimal que es contado en el histograma para formar la descripción. Posteriormente, el histograma de las etiquetas de todos los píxeles es utilizado como una descripción de la textura de la imagen. El vector de características mejorado espacialmente se obtiene concatenando los histogramas locales (no fusionándolos) [T. Ahonen et al, 2004].

$$H_i = \sum_{xy} I[LBP(x, y) = i], \quad i = 0, \dots, n - 1$$

(Ecuación 13)

$$I(x) = \begin{cases} 1, & \text{si } x \text{ es verdadero} \\ 0, & \text{otro valor} \end{cases}$$

(Ecuación 14)

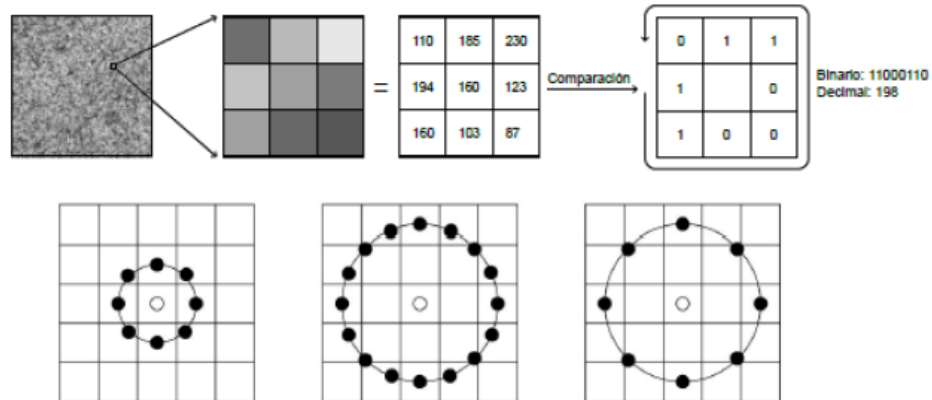


Figura 26: Obtención de los parámetros LBP, extraído de [T. Ahonen et al, 2004]

El LBP es un operador robusto contra transformaciones de imagen en escala de grises. En la siguiente imagen [Figura 27: Operador LBPH sobre imagen] se puede observar el funcionamiento del algoritmo LBPH y la transformación que realiza en la imagen.

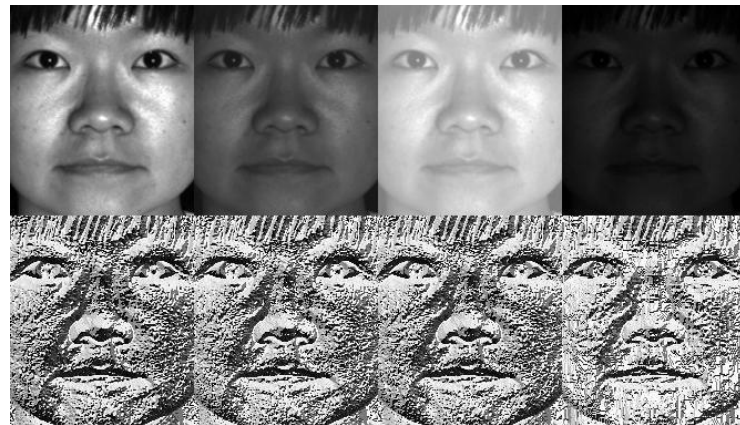


Figura 27: Operador LBPH sobre imagen



# CAPÍTULO 4

## DISEÑO Y DESARROLLO

En este capítulo, se presenta al lector el diseño, el desarrollo y el lenguaje utilizado para la implementación de la aplicación. También se hablará de cada uno de los procesos y métodos realizados hasta la obtención de la solución final... el reconocimiento facial.

### 4.1 INTRODUCCIÓN

Hoy en día, el reconocimiento facial es un método biométrico que está tan implantado en la sociedad que se ha creído oportuno realizar un trabajo evaluando los métodos o técnicas de reconocimiento facial que se encuentran a nuestro alcance. Para ello nos hemos apoyado de diferentes librerías, aplicaciones y recursos que se encuentran actualmente en el mercado *opensource*.

Por lo general la realización de cualquier proyecto es una ardua tarea, puede llegar a ser complicada si no se establecen ciertas directrices el trabajo puede convertirse infinitamente complicado. El conjunto de directrices o pasos que se utilizaron en este proyecto conforman la metodología de trabajo que se ha utilizado. En la (Figura 28) se describe de manera esquematizada la metodología de trabajo que se ha seguido.

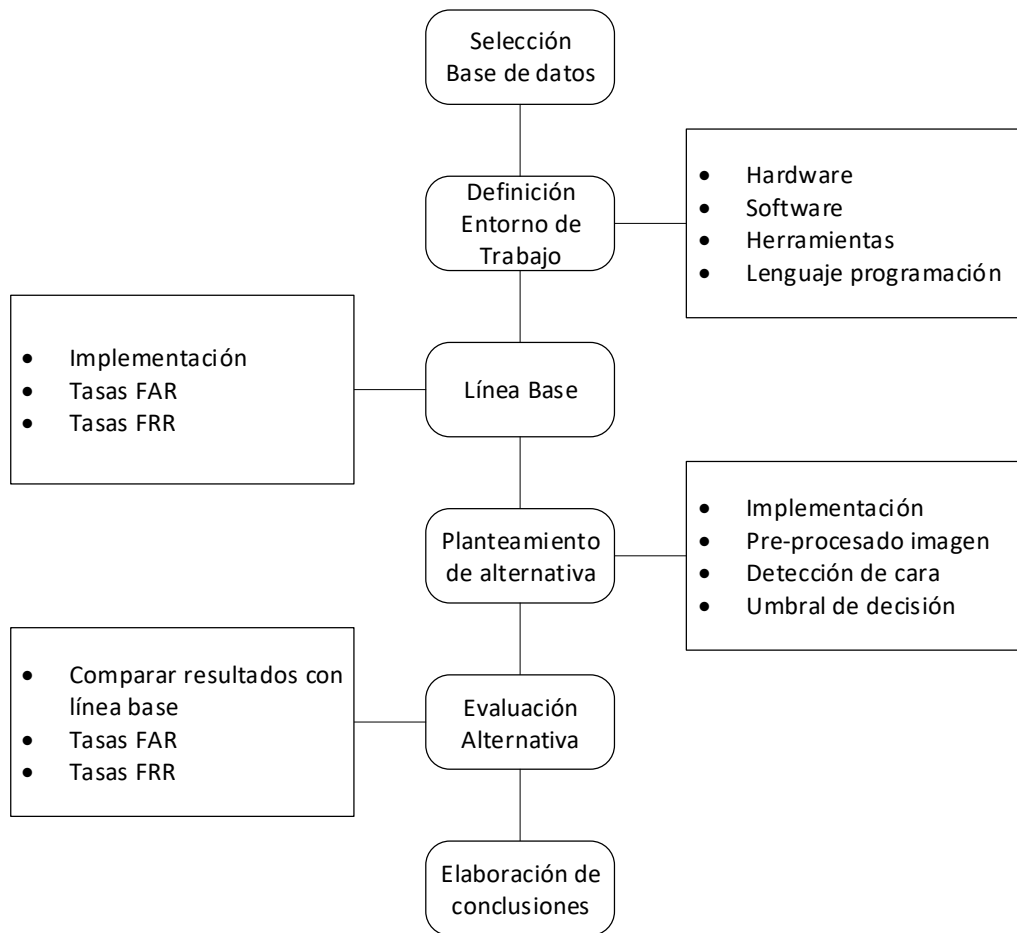


Figura 28: Metodología de trabajo

A lo largo del capítulo veremos la elección de la base de datos, así como todo el equipo tanto hardware como software utilizadas. Esta es una parte importante debido a que una buena selección del equipo puede conllevar a la obtención de resultados diferentes.

Una vez sabido el equipo del que se dispone, se pasara a la descripción del entorno de trabajo, así como las etapas del desarrollo en el lenguaje seleccionado y su implementación. Allí se verán los diferentes métodos implementados, el porqué de su aplicación y el objetivo perseguido.

El objetivo primordial del proyecto es la evaluación de algunas de las técnicas utilizadas en reconocimiento facial, por lo que a continuación se evaluarán los resultados obtenidos partiendo de una línea base de evaluación y se concluirá con la elaboración de conclusiones.

Se ha creído oportuno que tanto la evaluación de resultados como la elaboración de conclusiones tengan su propio capítulo por ser el objetivo primordial del proyecto. A continuación, se hablará sobre cada uno de los apartados de la figura anterior.

## 4.2 BASE DE DATOS

La base de datos con la que se ha decidido trabajar es la AT&T Laboratories<sup>9</sup> realizada por la Universidad de Cambridge para su departamento de Ingeniería, en concreto para el grupo de Visión y Robótica.

Está formada por diez imágenes de cada uno de los 40 diferentes sujetos tomadas entre abril de 1992 y abril de 1994. Para algunos sujetos fueron tomadas en diferentes momentos, variaciones de luz, expresiones faciales (ojos abiertos/cerrados, sonriendo/sin sonreír) y componentes estructurales (gafas/sin gafas). Todas las imágenes fueron tomadas sobre un fondo oscuro y con mirada al frente (permitiendo cierta movilidad lateral), como se puede observar en la Figura 29: AT&T Database.



Figura 29: AT&T Database

Las imágenes fueron almacenadas en formato PGM, el tamaño de cada imagen es 92x112 píxeles, con 256 niveles de gris por píxel. Las imágenes están organizadas en 40 directorios (uno por sujeto), con nombres de la forma sX, donde X indica el número del sujeto que se encuentra entre 1 y 40. En cada uno de los directorios hay diez imágenes diferentes del sujeto, donde están nombradas de la forma Y.pgm, donde Y es el número de la imagen para dicho sujeto, numeradas del 1 al 10.

<sup>9</sup> <http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>

Para dividir entre imágenes de entrenamiento e imágenes de prueba se ha establecido un 40% para entrenamiento y un 60% para test. Por lo que, 4 de cada 10 imágenes de cada individuo serán dedicadas a entrenar los algoritmos (160 imágenes). Las restantes serán usada en la etapa de test (240 imágenes).

### 4.3 ENTORNO DE TRABAJO

Para la realización de todo proyecto es necesario la aportación de ciertos requisitos software y hardware para la consecución del objetivo. Los diferentes requisitos utilizados en la elaboración del trabajo serán descritos a continuación.

Hardware:

- Procesador Intel ® Core ™ i7-4770 CPU a 3,40GHz
- 6 GB de RAM DDR3
- Disco duro SSD de 250GB
- Grafica AMD Radeon HD 5700 Series

Software:

- Windows 10 Pro 64-bit
- Visual Studio 2015 x86/x64
- Lenguaje C++ .Net
- Microsoft Access 2016 (aunque posteriormente fue sustituido por CSV)
- Librería OpenCV 2.4.13

En cuanto a los requisitos hardware, hubiera sido conveniente duplicar los recursos RAM en algunos casos, debido que ha habido cierta limitación causada por los elevados requisitos en los cálculos realizados por los algoritmos, en algunos casos consumiendo toda la memoria y finalizando el proceso.

El equipamiento hardware debe venir acompañado de un buen surtido de equipamiento software, en los siguientes apartados se ha creído conveniente una breve descripción de las herramientas utilizadas y su motivación para aplicarlas en el estudio.

### 4.3.1 OpenCV



OpenCV (Open Source Computer Vision Library) es una biblioteca libre de visión artificial desarrollada originalmente por Intel. OpenCV fue desarrollada para establecer una estructura común para las aplicaciones de visión artificial (computacional) y acelerar el uso de percepción maquina en los productos comerciales. Está implementado en diferentes lenguajes como Java, C++, python y

Matlab con lo que facilita su utilización.

Desde su aparición en 1999, se ha utilizado en infinidad de aplicaciones desde sistemas de seguridad con detección de movimiento, hasta aplicaciones de control de procesos donde se requiere reconocimiento de objetos. Este se debe a que su publicación se da bajo licencia BSD (Berkeley Software Distribution), que permite su utilización libre para propósitos comerciales, de investigación y modificación de código<sup>10</sup>.

OpenCV posee una estructura modular, donde cada módulo incluye varias librerías enfocadas a un objetivo concreto:

- ❖ **Core:** Es el módulo que define las estructuras básicas y funciones que usarán otros módulos.
- ❖ **Imgproc:** Dedicado al procesado de imagen, posee todo tipo de transformadas lineales o no lineales, geométricas, histogramas...
- ❖ **Video:** Dedicado al procesado de video, como seguimiento de movimiento, extracción del fondo y algoritmos de seguimiento.
- ❖ **Objdetect:** Dedicado a la detección de objetos. Incluye clases predefinidas (ojos, boca...)
- ❖ **Highgui:** Añade una interfaz sencilla a las aplicaciones de imagen y video (botones, barra de desplazamiento...)

La multitud de clases dedicadas a video y audio que posee OpenCV facilita la tarea del procesado de imágenes. En el apartado de implementación de la línea base se verán las más utilizadas.

---

<sup>10</sup> Wikipedia, OpenCV [<http://es.wikipedia.org/wiki/OpenCV>]

### 4.3.2 Visual Studio 2015 Enterprise



Una parte muy importante es la elección del entorno de desarrollo, en nuestro caso vamos a desarrollar la aplicación sobre Windows 10 Pro.

Al decidir utilizar también la librería OpenCV como librería de procesamiento de imagen, se creyó conveniente utilizar Visual Studio como IDE (Integrated Development Environment) debido a la amplia información que existe desde la propia página del proyecto<sup>11</sup>, así como en numerosas páginas de colaboración de programadores. De ahí que se decidiera también la utilización de lenguaje de programación C++. Net, que es para el que existe más ayuda en la página propia de OpenCV y en diferentes foros. Además de ser un lenguaje orientado a objetos muy similar a Java y de fácil comprensión.

La versión utilizada en el presente proyecto es Microsoft Visual Studio 2015 Enterprise. Aunque en la página de la propia universidad existe ya la versión 2017, se optó por la anterior por estar previamente instalada.

### 4.3.3 Microsoft Access 2016



Microsoft Access es un sistema de gestión de base de datos incluido en el paquete Office. Está diseñado para poder consultar los datos mediante consultas utilizando el lenguaje MySQL.

En un principio se implementó el código para usar una base de datos como medio de gestión de las imágenes, así de anotar los resultados. De esta forma tendríamos todo organizado en un único fichero, en tablas diferentes, tanto la obtención de los datos como los resultados de ejecución.

Finalmente se descartó esta opción (aunque los métodos están implementados) debido a que los accesos a la base de datos ralentizan la ejecución de procesos y consume más memoria que la lectura y escritura de archivos .csv.

<sup>11</sup> [https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec\\_api.html#FaceRecognizer](https://docs.opencv.org/2.4/modules/contrib/doc/facerec/facerec_api.html#FaceRecognizer)

## 4.4 LINEA BASE

En este capítulo se describe la línea base que sirvió como punto de partida para el proyecto. Esta línea base ha sido utilizada para comparar los resultados obtenidos con los sucesivos experimentos, así como servirá de referencia para comprobar la eficiencia del proceso de mejora propuesto.

Se ha creído conveniente dividir el apartado en fase de entrenamiento y fase de test para agilizar su comprensión. A continuación, se detallarán la algoritmia implementada para cada una de las fases, con sus distintas clases y métodos.

Pero antes hablaremos de una clase genérica utilizada en ambas etapas, esta clase es DB, esta clase tiene implementados los siguientes métodos y variables:

```
DataTable ^Pop_Data(String ^datos);
void Push_Result(int label, int predictedLabel, double time_elapsed);
void closeConnection();
void openConnection();
ArrayList ^TableToArrayImages(DataTable ^table);
void writeCSV(int label, int predicted_label, double time_elapsed, String ^nombre, double umbral);
ArrayList ^readCSV(String ^database);
```

Como se puede observar existen diferentes métodos implementados para el acceso y modificación de resultados en la base de datos (Access), que por los elevados tiempos de acceso fueron sustituidos por archivos csv. Empecemos por describir cada uno de los métodos:

- ✓ *openConnection ()*: Se encarga de establecer la conexión con la base de datos.
- ✓ *PopData ()*: En el string datos se le indica el tipo de datos a obtener (test o train). La lectura en Access devuelve un objeto DataTable que contiene la lectura de la consulta.
- ✓ *PushResult ()*: Se almacena en la base de datos el valor de la etiqueta original, la etiqueta predicha y el tiempo transcurrido.
- ✓ *TableToArrayImages ()*: Se convierte el objeto DataTable obtenido en PopData en una lista de Imágenes.
- ✓ *closeConnection ()*: Se encarga de cerrar la conexión con la base de datos una vez se termina de trabajar en ella.
- ✓ *writeCSV()*: Se encarga de escribir en un fichero .csv la etiqueta original, la etiqueta predicha, el tiempo transcurrido y se guarda en con el nombre “Resultados-nombre\_umbral.csv”. Donde nombre hace referencia al algoritmo utilizado.
- ✓ *readCSV()*: Se encarga de leer el fichero .csv y guardar los datos en una lista de objetos Imagen, dicha clase hablaremos más adelante sobre ella.

Al descartarse utilizar la base de datos de Access no tiene cabida hablar de la estructuración del fichero, por el contrario, sí es importante conocer la estructuración del fichero de entrada de datos y el fichero de salida en formato CSV.

El fichero CSV de entrada, que posee los datos de las imágenes tanto de entrenamiento como de test, posee la siguiente estructura:

*id; ruta de almacenamiento; etiqueta*

Como bien se puede observar en la Figura 30: Estructura fichero, donde *id* es solo el identificador de posición que ocupa en la lista, *ruta de almacenamiento* es el lugar donde se encuentra ubicada la imagen y *etiqueta* hace referencia a la etiqueta del individuo.

| A  | B                                                                                                                               | C  |
|----|---------------------------------------------------------------------------------------------------------------------------------|----|
| 1  | G:\Almacenamiento\Estudios\Master en Investigacion de Ingenieria de Software y Sistemas Informaticos\Faces\att_faces\s1\1.pgm   | 1  |
| 2  | G:\Almacenamiento\Estudios\Master en Investigacion de Ingenieria de Software y Sistemas Informaticos\Faces\att_faces\s1\10.pgm  | 1  |
| 3  | G:\Almacenamiento\Estudios\Master en Investigacion de Ingenieria de Software y Sistemas Informaticos\Faces\att_faces\s1\2.pgm   | 1  |
| 4  | G:\Almacenamiento\Estudios\Master en Investigacion de Ingenieria de Software y Sistemas Informaticos\Faces\att_faces\s1\3.pgm   | 1  |
| 5  | G:\Almacenamiento\Estudios\Master en Investigacion de Ingenieria de Software y Sistemas Informaticos\Faces\att_faces\s1\4.pgm   | 1  |
| 6  | G:\Almacenamiento\Estudios\Master en Investigacion de Ingenieria de Software y Sistemas Informaticos\Faces\att_faces\s10\1.pgm  | 10 |
| 7  | G:\Almacenamiento\Estudios\Master en Investigacion de Ingenieria de Software y Sistemas Informaticos\Faces\att_faces\s10\10.pgm | 10 |
| 8  | G:\Almacenamiento\Estudios\Master en Investigacion de Ingenieria de Software y Sistemas Informaticos\Faces\att_faces\s10\2.pgm  | 10 |
| 9  | G:\Almacenamiento\Estudios\Master en Investigacion de Ingenieria de Software y Sistemas Informaticos\Faces\att_faces\s10\3.pgm  | 10 |
| 10 | G:\Almacenamiento\Estudios\Master en Investigacion de Ingenieria de Software y Sistemas Informaticos\Faces\att_faces\s10\4.pgm  | 10 |
| 11 | G:\Almacenamiento\Estudios\Master en Investigacion de Ingenieria de Software y Sistemas Informaticos\Faces\att_faces\s11\1.pgm  | 11 |
| 12 | G:\Almacenamiento\Estudios\Master en Investigacion de Ingenieria de Software y Sistemas Informaticos\Faces\att_faces\s11\10.pgm | 11 |

Figura 30: Estructura fichero entrada

El fichero CSV de salida, el que posee los resultados de la evaluación de los algoritmos posee la siguiente estructura:

*etiqueta original; etiqueta predicha; tiempo transcurrido*

Como observamos en la Figura 31: Estructura fichero salida, la *etiqueta original* corresponde al individuo al que pertenece la imagen, *etiqueta predicha* es el sujeto identificado y *tiempo transcurrido* es el tiempo que ha pasado entre que la imagen entra en el sistema hasta que se produce una identificación.

|    | A  | B         | C         |
|----|----|-----------|-----------|
| 1  | 1  | NOT MATCH | 0,0156145 |
| 2  | 1  | NOT MATCH | 0,0156304 |
| 3  | 1  | 1         | 0         |
| 4  | 1  | 24        | 0,0156163 |
| 5  | 1  | NOT MATCH | 0,0156295 |
| 6  | 10 | 10        | 0         |
| 7  | 10 | 10        | 0,0156111 |
| 8  | 10 | 10        | 0,0156319 |
| 9  | 10 | 10        | 0         |
| 10 | 10 | 10        | 0,0156293 |

Figura 31: Estructura fichero salida



#### 4.4.1 Entrenamiento

El proceso seguido para la etapa de entrenamiento se describe en la (Figura 32), cada uno de los apartados han sido implementados en C++ .Net. A lo largo del apartado se explicarán las clases construidas con sus respectivos métodos, así como el funcionamiento de cada uno de ellos.

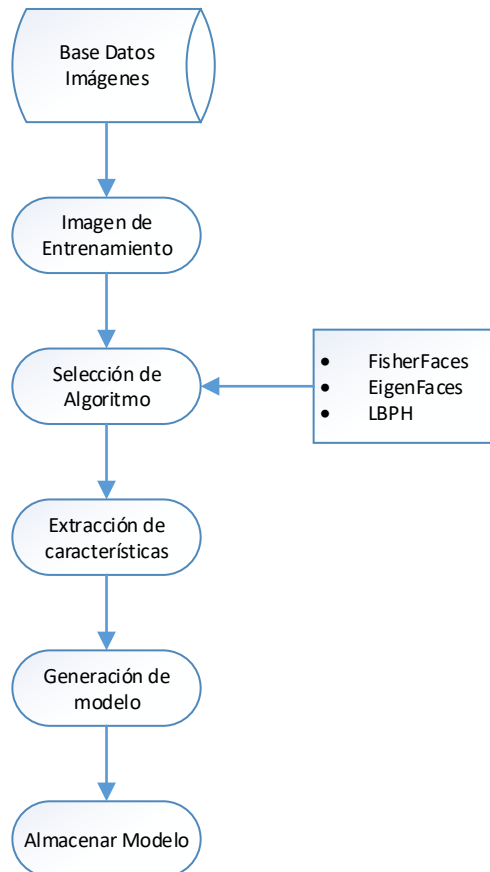


Figura 32: Diagrama de entrenamiento

Para poder manejar toda la información relacionada con las imágenes, tal como la ruta de almacenamiento, la etiqueta asignada, la etiqueta que se le ha predicho... así como si la imagen se ha almacenado nuevamente en blanco y negro. En definitiva, cualquier cambio o modificación que afecte a la imagen. Esta clase se llama *Images* y está formada por:

```

public:
//Método constructor
Images();
Images(int id, int label, String ^ filename);

private:
//Variables
String ^filename, ^grayFilename;
  
```

```

int id, label, predictedLabel;

public:
// Guardamos la ruta de la imagen
void setFilename(String ^filename);
// Guardamos la ruta de la imagen en escala de grises
void setGrayFilename(String ^grayFilename);
// Guardamos la etiqueta asignada al filename
void setLabel(int label);
// Guardamos la etiqueta predicha
void setPredictedLabel(int predictedLabel);

// Métodos get para devolver los datos anteriores
String ^getFilename();
String ^getGrayFilename();
int getLabel();
int getPredictedLabel();
int getId();

```

El funcionamiento de cada uno de los métodos viene descrito en verde, donde los métodos *set* son para guardar o actualizar los valores. Por el contrario, los métodos *get* son para poder usar esos valores posteriormente.

Una vez conocemos el funcionamiento de la clase *Images* podemos continuar hablando del método que se encarga del entrenamiento. Este método se llama *train* y dependiendo del algoritmo que se le pasa se crea uno u otro de la siguiente manera:

```

// Creamos el modelo utilizando el algoritmo EigenFaces
Ptr<FaceRecognizer> model = createEigenFaceRecognizer();

// Creamos el modelo utilizando el algoritmo Fisherfaces
Ptr<FaceRecognizer> model = createFisherFaceRecognizer();

// Creamos el modelo utilizando el algoritmo LBPH
Ptr<FaceRecognizer> model = createLBPHFaceRecognizer();

```

Para el algoritmo *Eigenfaces* puedes indicar cuantos *eigenfaces* quieres utilizar para la creación del modelo y también puedes indicarle un umbral al inicializarle. Por ejemplo, si queremos utilizar solo 12 *eigenfaces* se inicializaría de la siguiente manera:

```
Ptr<FaceRecognizer> model = createEigenFaceRecognizer(12);
```

Por el contrario, si queremos utilizar 12 *eigenface* (mantener 12 componentes principales) y establecer un umbral de confianza de 200 sería de la siguiente forma:

```
Ptr<FaceRecognizer> model = createEigenFaceRecognizer(12, 200.0);
```

El umbral le puedes establecer en la etapa de entrenamiento o bien, como se ha hecho en el trabajo, en la etapa de test. Por último, si queremos utilizar todas las *eigenfaces* posibles solo debemos establecer un 0 en lugar del 12.

Para el caso de *Fisherfaces* es prácticamente similar, en lugar de elegir cuantas *eigenfaces* elegimos cuantas *fisherfaces*. El procedimiento es igual al descrito anteriormente.

El algoritmo LBPH sufre ligeras variaciones, ya que es un algoritmo basado en vecindad el uso de componentes principales no tiene cabida, por el contrario, si podemos establecer el radio de vecindad. Por defecto se establece un radio de 1-8 vecinos, para modificar ese valor de 2 a 16 sería:

```
Ptr<FaceRecognizer> model = createLBPHFaceRecognizer(2, 16);
```

Una vez elegido el algoritmo se pasa a procesar el objeto *Imagen*, de ahí obtenemos la ruta de la imagen y la cargamos en un objeto *Mat* (ver Anexo). A su vez, obtenemos su etiqueta y la almacenamos como entero *int*. Cada imagen leída será almacenada en un vector de objetos *Mat* y cada etiqueta será almacenada a su vez en un vector de enteros, de forma que conceptualmente sería de la siguiente forma:

```
// Creamos los vectores donde almacenaremos las imagenes de entrenamiento y su label
vector<Mat> images;
vector<int> labels;
```

Visualmente quedaría:

|   | A       | B   |
|---|---------|-----|
| 1 | Imagen1 | 1   |
| 2 | Imagen1 | 1   |
| 3 | Imagen1 | 1   |
| 4 | Imagen1 | 1   |
| 5 | Imagen2 | 2   |
| 6 | Imagen2 | 2   |
| 7 | Imagen2 | 2   |
| 8 | ...     | ... |

Todo lo descrito anteriormente en lenguaje C++ .Net quedaría de la siguiente manera:

```
// Obtenemos objeto imagen de la lista de entrenamiento
Images ^im_tr = safe_cast<Images^> (tr->Current);

// Leemos la imagen para procesarla
processedImage = imread(toStandardString(im_tr->getFilename()), CV_LOAD_IMAGE_GRAYSCALE);

// Leemos la imagen y la almacenamos en el vector con su correspondiente label
images.push_back(processedImage);
labels.push_back(im_tr->getLabel());

// Guardamos el ancho y el alto
setHeight(processedImage.rows);
setWidth(processedImage.cols);
```

El método *toStandardString* se utiliza para convertir un objeto *System::String* en *std::string*, que es el manejado por la librería OpenCV. Y se cargan en blanco y negro. También guardamos el ancho y el alto de las imágenes para realizar un redimensionado de las imágenes en la etapa de test. Para que todas las imágenes que se procesan tengan las mismas dimensiones, que es requisito indispensable para que los algoritmos funcionen.

Como se puede observar, las imágenes son leídas directamente en blanco y negro. Si no la leyéramos en B/N posteriormente tendríamos que convertirla para poder realizar transformaciones de histograma. Es de agradecer que OpenCV nos permite realizarlo directamente en la lectura ahorrándonos un paso posterior.

Por último, se realiza el entrenamiento del modelo pasándole el vector de las imágenes (images) y el vector de las etiquetas correspondientes (labels). Este es el proceso computacional más costoso, por lo que nos llevará un tiempo obtener el modelo. Una vez terminado, almacenamos el modelo en formato .yaml. El formato YAML es estándar para OpenCV.

```
// Entrenamos el modelo
model->train(images, labels);

// Guardamos el modelo
modelStore = toStandardString(modelSave);
model->save(modelStore + algoritmo + ".yaml");
```

Hasta aquí la etapa de entrenamiento de la línea base, continuamos con la etapa de test.

### 4.4.2 Test

Esta es la etapa de validación del modelo generado anteriormente, los pasos establecidos son ligeramente diferentes (Figura 33). Al igual que en apartado anterior, explicaremos cada uno de los pasos implementados con su respectivo código.

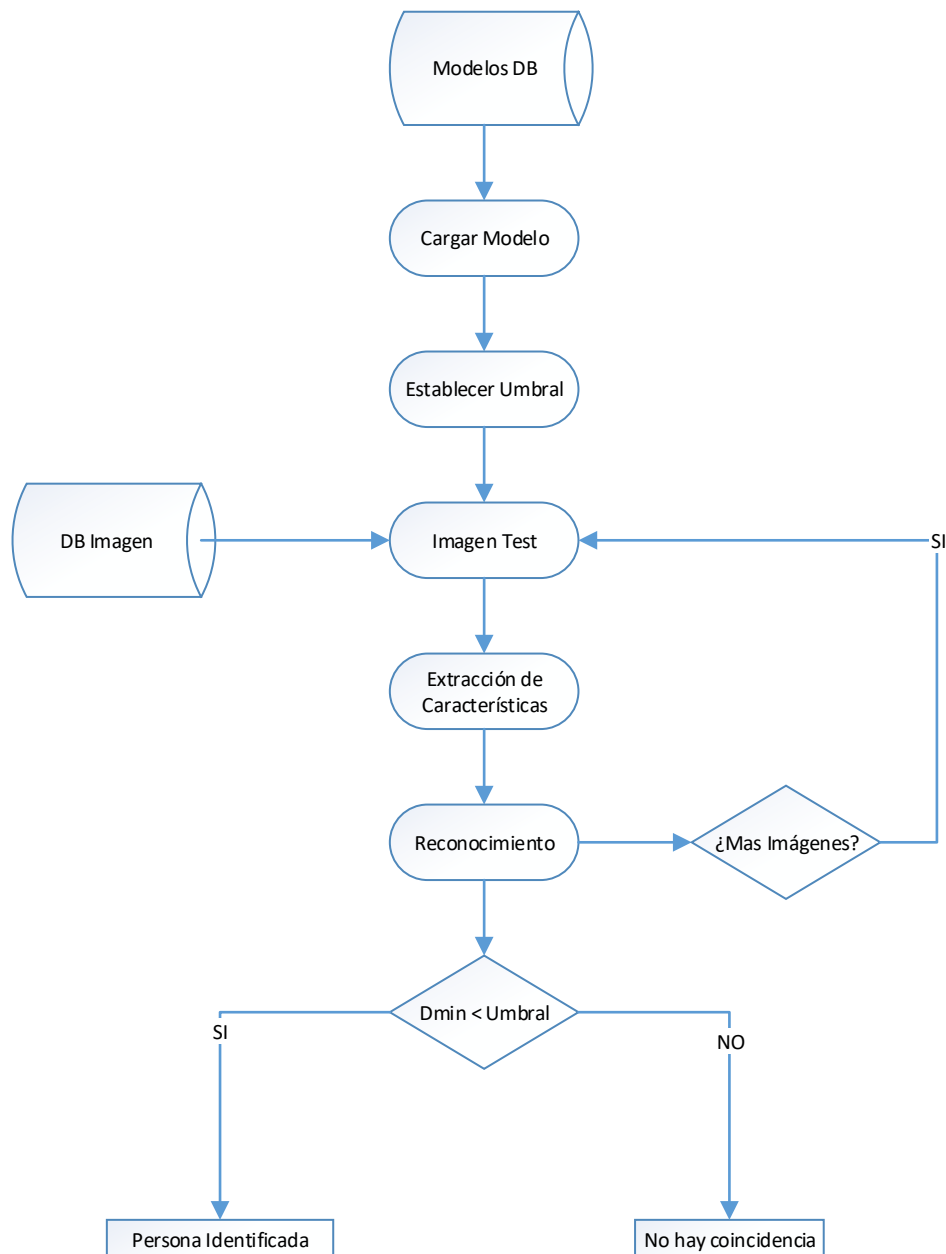


Figura 33: Diagrama de Test

En el primer paso se establece con qué modelos queremos realizar las pruebas (FFR, EFR o LBPH) para cargar posteriormente el algoritmo almacenado en la etapa de entrenamiento. Una vez que hemos establecido el umbral de decisión (distancia mínima

con la que establecemos una identificación correcta) cargamos el modelo a evaluar, como podemos observar en las siguientes líneas de código.

```
// Establecemos algoritmo
modelTest = createEigenFaceRecognizer();
// Establecemos umbral
modelTest->set("threshold", getThreshold());
// Cargamos el modelo almacenado
modelTest->load(modelStore + algoritmo + ".yml");
```

A continuación, se lee la imagen destinada a test y se comprueba que posea las mismas dimensiones que las imágenes de entrenamiento. En caso contrario se redimensionaría al tamaño de las imágenes de entrenamiento almacenado previamente. Una vez que tenemos todo en su dimensión correcta procedemos a la identificación, si la distancia mínima es menor que el umbral establecido nos devolverá la etiqueta a la que pertenece la imagen, en caso contrario nos devolverá -1 indicando que no existe coincidencia en la base de datos.

```
// Comprobamos que posee la misma dimensión que las imágenes de entrenamiento
if ((processedImage.rows != getHeight()) && (processedImage.cols != getWidth()))
{
    // Cambiamos tamaño de la imagen
    resize(processedImage, resized, Size(getWidth(), getHeight()), 1, 1, INTER_CUBIC);
}
// Predecimos la etiqueta de la imagen
int predictedLabel = modelTest->predict(processedImage);
```

Cada predicción realizada se almacena al instante en el fichero CSV, cuando no quedan ninguna imagen por predecir el algoritmo termina y cierra el fichero CSV con los resultados obtenidos para su posterior evaluación.

## 4.5 ALTERNATIVA

La alternativa presentada consiste en un pre-procesado de la imagen previo a la generación del modelo. Este pre-procesado consiste en un tratamiento sobre la imagen para eliminar ruido y luego se realiza una detección del rostro sobre la imagen. Una vez detectado el rostro se recorta dicha detección, de esta forma se pretende introducir menos información redundante a la hora de generar el modelo.

El método que se encarga del pre-procesado de la imagen se llama *cleanNoise()*, donde se realiza un realzado de la imagen. El realzado de la imagen se suele utilizar para que la imagen de salida sea más fácil de interpretar. Se puede hacer que un detalle de la imagen sea visible más fácilmente o bien reducir el ruido presente. La forma de hacerlo es pasar sobre la imagen un filtro de medianas y realizar una igualación de histograma. Su implementación en OpenCV sería:

```
// Pasamos filtro de medianas 5x5 para eliminar ruido sal y pimienta
medianBlur(image, image, 5);

// Ecuálizamos la imagen para eliminar sombras
equalizeHist(image, equalGray);
```

Con el filtro de medianas lo que conseguimos es eliminar el molesto ruido sal y pimienta. En nuestro caso utilizamos una máscara de 5x5 que se irá aplicando a todos los píxeles de la imagen, donde se reemplaza por la mediana de los píxeles vecinos. La mediana se calcula ordenando los valores de los píxeles vecinos y seleccionando el que queda en medio.

La igualación de histograma es una técnica clásica que se puede utilizar para reducir el contraste en la imagen o bien se utiliza para reducir el número de niveles de gris en una imagen. El contraste de una imagen viene dado por la variabilidad, dispersión o esparcimiento de los tonos de gris de los diferentes píxeles de la imagen. Con ello se consigue una mejora de la imagen, ya que se obtiene una imagen con más contraste y los objetos aparecen más resaltados.

Una vez limpiado la imagen, el siguiente paso en nuestra identificación es detectar la cara y recortarla. De esta forma introducimos menos información redundante a la hora de generar el modelo, tales como información de fondo, información redundante sobre morfología del pelo...

Para la detección de caras hemos implementado dos métodos: *faceDetection* y *facesDetection*. El primero solo identifica una cara, como nuestra base de datos está formada por una única cara no tenía sentido que identificáramos más de un rostro. El segundo, como su nombre indica, identifica todos los rostros que existen en una imagen. Este método está pensado para trabajos futuros de identificación múltiple, que una vez puesto no nos ha costado implementarle. Pasamos a comentar cada uno de los métodos, donde primero se expone el código y luego se explicará su funcionamiento.

### *faceDetection*

El método *faceDetection* se encarga de detectar las coordenadas del rostro en la imagen, posteriormente recorta el rostro y lo redimensiona a los valores de la imagen de entrada. En primer lugar, se definen ciertas variables necesarias, como un vector de objetos Rect, las coordenadas de los rostros detectados son almacenados en objetos Rect de ahí que necesitemos dicho vector (*faces*). También se necesita el propio *CascadeClassifier* nombrado por la variable *haars*, que es el clasificador en cascada propiamente dicho. Por último, las dimensiones de la imagen de entrada, un objeto rect para almacenar las coordenadas de la imagen detectada (paso que se podría obviar, pero se incluyó para hacer más legible el código). Veamos el código.

```
// Vector que almacena la posicion de cada una de las caras detectadas
vector<Rect> faces;
// Clasificador en cascada
CascadeClassifier haars;
// Obtenemos ancho y alto de la imagen para redimensionar
int height = getHeight();
int width = getWidth();
// Coordenadas de la cara detectada
Rect faceDetected;
// Imagenes
Mat face, face_resized, frame_gray;
```

El siguiente paso es cargar el fichero XML que indica los pasos por defecto para la detección de las imágenes, dichos ficheros son administrados por OpenCV y se pueden encontrar en la ruta `opencv\\sources\\data\\haarcascades\\`.

El método *detectMultiScale* se encarga de detectar todos los rostros existentes en la imagen, guarda las coordenadas de las regiones de los rostros detectados en objetos de tipo Rect que después los agrega al vector *faces*. Las regiones detectadas tienen forma de rectángulo, por este motivo las regiones se almacenan en objetos de tipo *Rect*. Estos objetos guardan las coordenadas (x,y) de la esquina superior derecha, el ancho y alto del rectángulo. Si no se detecta ningún rostro el vector *faces* tendrá longitud 0, en este caso se devolvería la propia imagen.

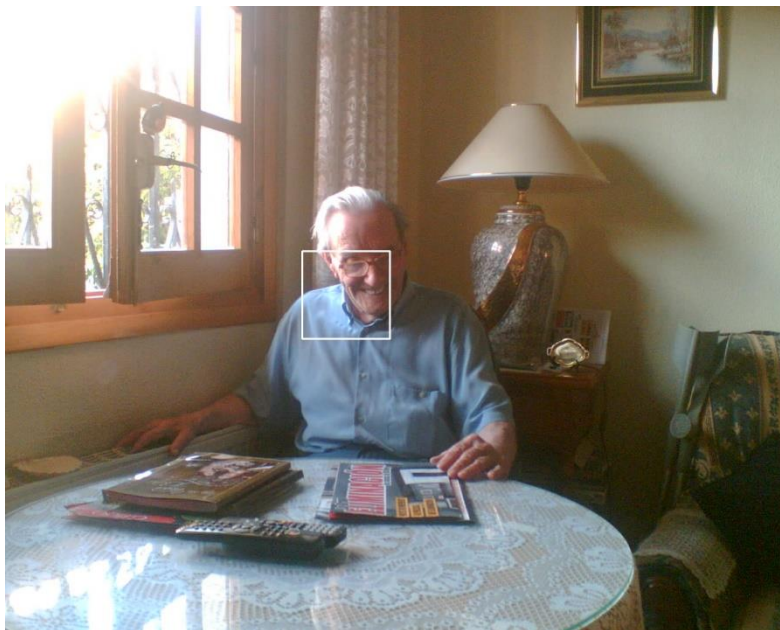
Una vez obtenida la región del rostro se procede a dibujar dicha imagen, al tener forma de rectángulo con el método *rectangle* se dibuja un rectángulo sobre el rostro detectado (Figura 34: (a) Rostro detectado; (b) Rostro recortado; (c) Rostro recortado en B/N (a)). A continuación, se recorta dicho rostro y se guarda en la variable *faces* (Figura 34: (a) Rostro detectado; (b) Rostro recortado; (c) Rostro recortado en B/N (b), (c)). Se redimensiona la nueva imagen al tamaño de la imagen de entrada y se realiza una igualación de histograma antes de devolver la imagen.





**Figura 34: (a) Rostro detectado; (b) Rostro recortado; (c) Rostro recortado en B/N**

Es importante que las imágenes sean tomadas de frente y con el menor ruido de fondo posible. Como se puede observar en la (Figura 35: Detección rostro ante componentes estructurales) la información redundante de fondo, el ladeo de la cabeza y la presencia de componentes estructurales influyen en la detección del rostro.



**Figura 35: Detección rostro ante componentes estructurales**

**facesDetection**

Este método detecta múltiples caras en una misma imagen, el funcionamiento es similar al anterior con la salvedad de que devuelve un vector de objetos Mat con todas las caras detectadas ya recortadas.

La salvedad con el método anterior reside en recorrer el vector con las coordenadas de las caras detectadas por el método *detecMultiScale* y recortar dicha parte sobre la imagen de entrada. Así, cada cara será almacenada en el vector caras creado para dicho fin. A continuación, se muestran solo las partes de código que difieren del método anterior:

```
// Vector que almacena cada una de las caras detectadas
vector<Mat> caras;

//Se recorre el vector con las coordenadas de las caras
for (size_t i = 0; i < faces.size(); i++)
{
    //Se dibuja un rectángulo en la cara i-esima detectada
    rectangle(m, faces[i], CV_RGB(255, 255, 255), 1);
    //Se recorta la cara i-esima
    Mat fa = m(faces[i]);
    //Se almacena en el vector de caras
    caras.push_back(fa);
}
```

La variable *cara* es la que se devuelve, dicha variable contendrá cada una de las caras recortadas existentes en la imagen. Como se puede observar en la (Figura 36: Variación de luminosidad en rostro) las variaciones de luminosidad debidas al flash cerca del rostro influyen negativamente en la detección.



Figura 36: Variación de luminosidad en rostro

Aunque en otros casos simplemente no detecta algún rostro en condiciones favorables y asume como tales objetos de la escena (Figura 37: Detección múltiple de rostros).



(a)



(b)

**Figura 37: Detección múltiple de rostros**

Cabe resaltar que en estos últimos ejemplos el detector de rostros ha sido llevado al extremo, donde se trata de identificar rostros sobre un fondo complicado lleno de información redundante. Aun así, se puede concluir que el detector es efectivo aun en condiciones desfavorables.

**Figura 38: Detección de ojos**

Para terminar con el apartado de implementación, comentaremos un último método implementado que se llama *eyeDetection* y detecta los ojos sobre cada uno de los rostros detectados (Figura 38: Detección de ojos). El funcionamiento es similar al anterior pero una vez detectada una cara se produce la detección de los ojos, se pensó para posibles mejoras futuras por lo que no se explicará su implementación en este apartado.

# CAPÍTULO 5

## RESULTADOS

En este capítulo, se presenta al lector la evaluación de los algoritmos implementados en dicho documento. Así como las tasas FAR y FRR obtenidas en la línea base y posteriormente en la evaluación de las mejoras introducidas.

### 5.1 INTRODUCCIÓN

La evaluación de los resultados es un indicativo de como de buena ha sido nuestra línea de investigación perseguida. En una evaluación los resultados pueden ser alentadores, malos o simplemente no introducen mejoras sustanciales, pero en todos los casos son importantes. Unos malos resultados pueden resultar muchas veces aperturas a nuevas líneas de investigación y mejora que desencadenarán en excelentes resultados.

En los siguientes apartados se realizará la evaluación de los algoritmos *FisherFaces*, *EigenFaces* y *LBPH* para cada uno de los diferentes valores de umbral. Se empezará por la evaluación de la línea base (apartado 4.4) y se concluirá evaluando los resultados obtenidos tras la mejora de pre-procesado de la imagen.



## 5.2 RESULTADOS LINEA BASE

La evaluación de la línea base se ha realizado sobre las imágenes sin realizarse ningún tratamiento sobre ella. Se han realizado los experimentos sobre distintos tipos de umbrales para poder observar la fluctuación de los resultados con la variación del mismo.

### 5.2.1 Método EigenFaces

Se ha probado la efectividad del algoritmo con los valores de umbral comprendidos entre [1000 – 6000], con incrementos de 1000. Valores inferiores de 1000 se obtenía una tasa de FRR del 100%, por tanto, se ha creído conveniente empezar desde 1000 para poder ver una progresión.

| RESULTADOS EIGENFACES |         |         |
|-----------------------|---------|---------|
| UMBRAL                | FRR (%) | FAR (%) |
| 1000                  | 99,00   | 0,00    |
| 2000                  | 67,00   | 0,00    |
| 3000                  | 16,50   | 2,00    |
| 4000                  | 0,00    | 7,50    |
| 5000                  | 0,00    | 7,50    |
| 6000                  | 0,00    | 7,50    |

Tabla 2: Resultados LB con EigenFaces

Como se puede observar en la Tabla 2: Resultados LB con EigenFaces a medida que se aumenta el umbral se va reduciendo la tasa FRR y aumentando FAR. Con umbrales bajos (1000) el sistema es incapaz de identificar ninguna imagen con las almacenadas en la base de datos, esto es debido a que la distancia euclídea establecida entre la imagen de entrada y las almacenadas en la base de datos es superior al umbral establecido. Por tanto, no puede establecer una coincidencia y nos devuelve el sistema un -1, indicando así que la imagen de entrada no estaba previamente almacenada en nuestra base de datos.

Lo ideal para una correcta identificación es que la distancia euclídea entre ambas imágenes (entrada y almacenada) sea exactamente 0. Pero estos casos no siempre suceden ya que a menudo incluso cuando dos imágenes de una misma persona son comprobadas, cualquier variación en las condiciones de iluminación, cualquier mueca... puede generar valores de distancia elevados. De ahí que obtengamos las variaciones observadas en la gráfica de la Figura 39: Variación FRR y FAR, donde podemos observar que FRR disminuye hasta que se obtiene cierto umbral de referencia, que aunque aumente su valor, tanto FRR como FAR se mantienen invariante.

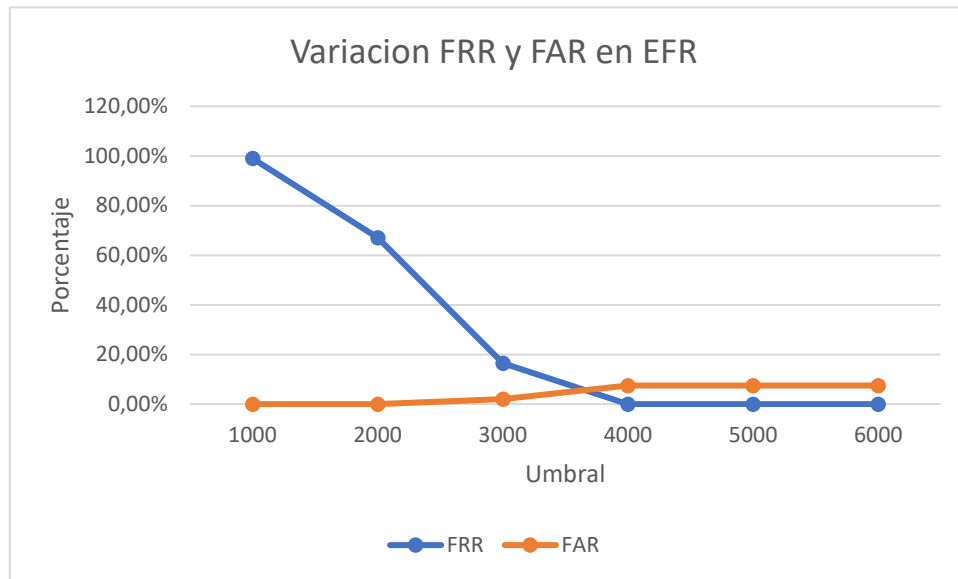


Figura 39: Variación FRR y FAR en EFR

### 5.2.2 Método FisherFaces

Al igual que en el apartado anterior se ha valorado el algoritmo con valores de umbral comprendidos entre [1000-6000] con incrementos de 1000. En la Tabla 3: Resultados LB con FisherFaces se puede observar una variación con respecto a *FisherFaces*, observamos que ya con un valor de umbral de 1000 obtenemos una tasa FRR del 43,5% obteniendo una tasa de acierto superior al 50%. Esto es debido a la robustez añadida que posee el algoritmo ante cambios en la intensidad de la luz y muecas (apartado 3.3.2).

| RESULTADOS FISHERFACES |         |         |
|------------------------|---------|---------|
| UMBRAL                 | FRR (%) | FAR (%) |
| 1000                   | 43,50   | 0,00    |
| 2000                   | 0,00    | 12,00   |
| 3000                   | 0,00    | 12,00   |
| 4000                   | 0,00    | 12,00   |
| 5000                   | 0,00    | 12,00   |
| 6000                   | 0,00    | 12,00   |

Tabla 3: Resultados LB con FisherFaces

A continuación, podemos observar de manera gráfica esa variación, al igual que en el apartado anterior sobre cierto umbral ambas tasas continúan inalterables.

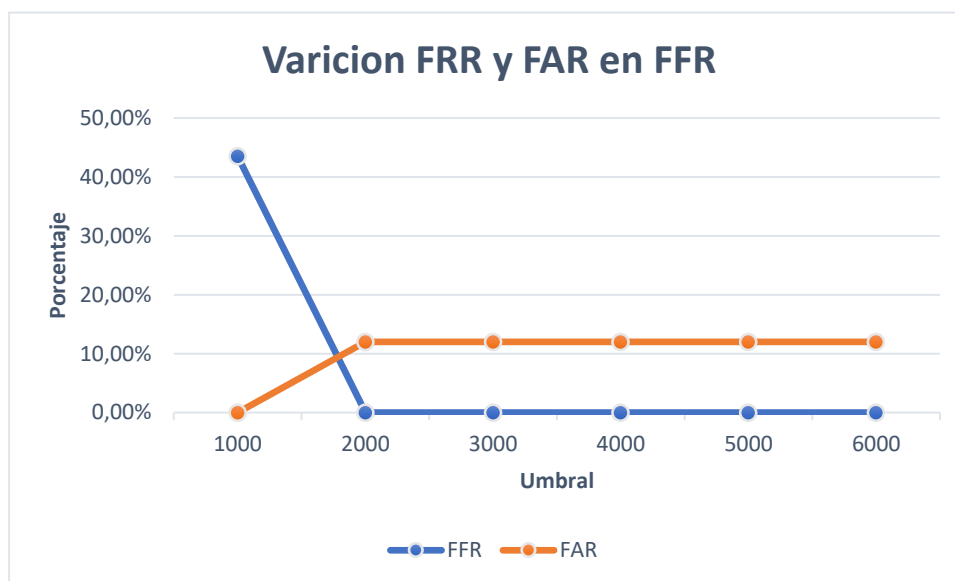


Figura 40: Variación FRR y FAR en FFR

Se puede observar que se obtiene una mejor tasa de FRR ya desde umbrales bajos, a partir de 2000 el valor de FAR y FRR se mantiene constante y empeora con respecto a *EigenFaces*, llegando a ser un 4,5% mayor. Con umbrales bajos conseguimos una mayor significación sobre muestras existentes en la base de datos, pero a medida que se aumenta el umbral obtenemos una mayor tasa de falsa aceptación. En cuestiones de seguridad es preferible que un usuario legítimo no pueda entrar que un usuario ilegítimo pueda acceder, siempre y cuando no estemos en un supuesto ideal y si en el peor de los casos, en este caso una FAR del 12%.

### 5.2.3 Método LBPH

Debido a la robustez mostrada por el algoritmo su evaluación ha sufrido ligeras modificaciones con respecto a las anteriores. Se han tenido que usar valores de umbral inferiores a 1000 para poder observar una variación significativa del algoritmo a medida que el umbral variaba su valor. Se ha utilizado los valores de umbral 20, 50, 100, 500 y [1000 - 6000], los resultados obtenidos se pueden observar en la Tabla 4: Resultados LB con LBPH. Como podemos observar, con un umbral de 50 obtenemos una mejora del 19,5% en FRR con respecto a *Fisherfaces* para un umbral de 1000, si comparamos al mismo umbral la mejora es del 43,5% en FRR, peor en cambio la tasa FAR se incrementa en un 1,5% con respecto a *Fisherfaces*. Con respecto a *Eigenfaces* obtenemos una mejoría para el umbral de 1000 del 99%. Por el contrario, se produce un incremento de la tasa FAR del 11,5%.



| RESULTADOS LBPH |         |         |
|-----------------|---------|---------|
| UMBRAL          | FRR (%) | FAR (%) |
| 20              | 100     | 0,00    |
| 50              | 24,00   | 1,50    |
| 100             | 0,00    | 11,50   |
| 500             | 0,00    | 11,50   |
| 1000            | 0,00    | 11,50   |
| 2000            | 0,00    | 11,50   |
| 3000            | 0,00    | 11,50   |
| 4000            | 0,00    | 11,50   |
| 5000            | 0,00    | 11,50   |
| 6000            | 0,00    | 11,50   |

Tabla 4: Resultados LB con LBPH

Las variaciones anteriormente comentadas se pueden observar de manera gráfica en la Figura 41: Variación FRR y FAR con LBPH, donde se observa que para valores de umbral superiores a 100 las tasas se mantienen inalterables.

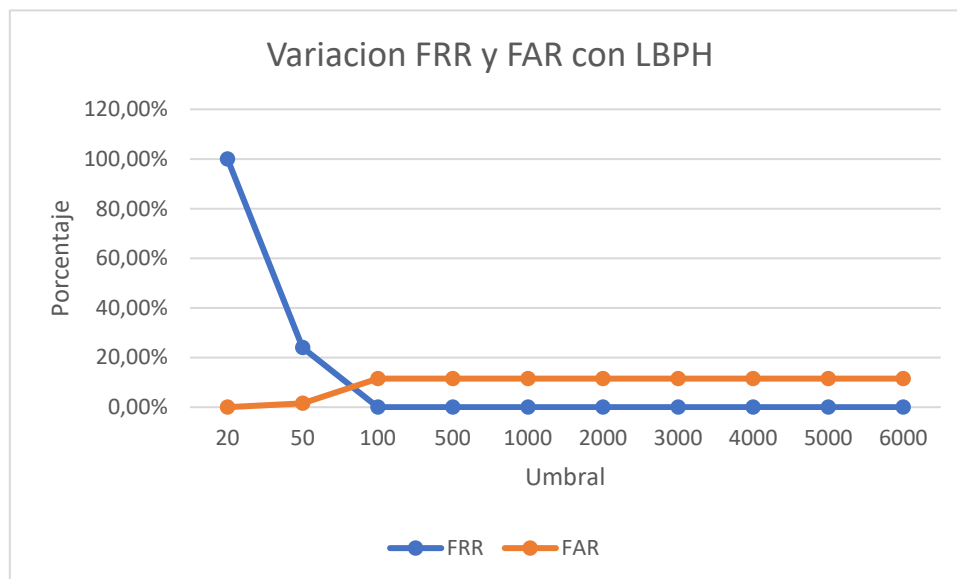


Figura 41: Variación FRR y FAR con LBPH

### 5.3 RESULTADOS ALTERNATIVA

En este apartado se van a comparar los resultados obtenidos con la alternativa planteada en el capítulo 4. Donde se realiza un pre-procesado aplicado a las imágenes con los resultados evaluados en el apartado anterior. Recordamos que a las imágenes se les aplica un filtro de medianas (eliminar el ruido sal y pimienta), luego se procede a una igualación de histograma (mejora de contraste) y por último se recorta el rostro detectado (eliminar información redundante).

#### 5.3.1 Método EigenFaces

Los resultados obtenidos tras todo el pre-procesado mencionado anteriormente se observan en la Tabla 5: Resultados alternativa con EFR . Donde encontramos dos valores de umbral que no estaban en la evaluación de la línea base, estos valores han sido utilizados para evaluar el comportamiento del algoritmo. Se observa que a valores por encima de 6000 ambas tasas permanecen constantes sin introducir apenas mejora en ninguna de ambas tasas.

En la Figura 42: Variación FRR y FAR en EFR con podemos ver que la pendiente es menos pronunciada en el caso de FRR con respecto a la línea base, pero también los resultados son peores para los mismos valores de umbrales. También se puede ver que para el caso de FAR ocurre lo contrario, se consigue mantener más baja para umbrales inferiores a 3000 y superiores a 6000.

| RESULTADOS EFR |         |         |
|----------------|---------|---------|
| UMBRAL         | FRR (%) | FAR (%) |
| 1000           | 99,50%  | 0,00%   |
| 2000           | 89,00%  | 0,00%   |
| 3000           | 58,50%  | 0,50%   |
| 4000           | 33,00%  | 6,00%   |
| 5000           | 13,50%  | 16,50%  |
| 6000           | 22,50%  | 4,00%   |
| 9000           | 26,00%  | 0,00%   |
| 11000          | 26,00%  | 0,00%   |

Tabla 5: Resultados alternativa con EFR

Se puede observar que el umbral optimo estaría comprendido entre 5000 y 6000, donde encontraríamos cierto balance entre ambas tasas. Entraríamos en la disyuntiva de qué es más importante, si una persona que no pertenece al sistema sea reconocida como

tal o que una persona existente no pueda ser reconocida. Valorando esas premisas deberíamos ajustar nuestro umbral.

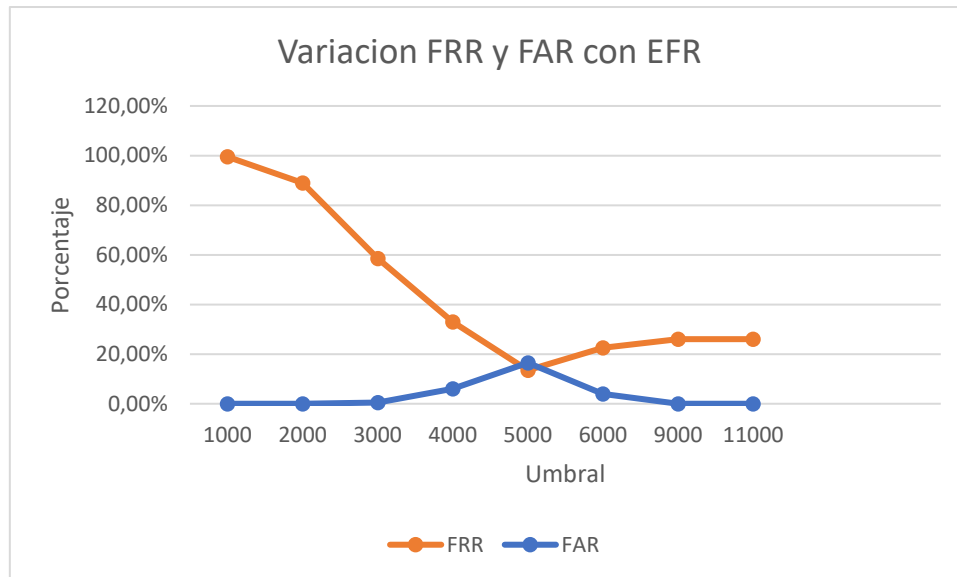


Figura 42: Variación FRR y FAR en EFR con alternativa

A continuación, se puede observar en la Tabla 6: Comparativa línea base/alternativa en EFR la comparación de los resultados obtenidos con la imagen sin procesar. Se ha marcado en verde los mejores resultados obtenidos para ambos casos

| UMBRAL | LINEA BASE |         | ALTERNATIVA |         |
|--------|------------|---------|-------------|---------|
|        | FRR (%)    | FAR (%) | FRR (%)     | FAR (%) |
| 1000   | 99,00      | 0,00    | 99,50%      | 0,00%   |
| 2000   | 67,00      | 0,00    | 89,00%      | 0,00%   |
| 3000   | 16,50      | 2,00    | 58,50%      | 0,50%   |
| 4000   | 0,00       | 7,50    | 33,00%      | 6,00%   |
| 5000   | 0,00       | 7,50    | 13,50%      | 16,50%  |
| 6000   | 0,00       | 7,50    | 22,50%      | 4,00%   |

Tabla 6: Comparativa línea base/alternativa en EFR

### 5.3.2 Método FisherFaces

Los resultados obtenidos se pueden observar en la Tabla 7: Resultados alternativa con FFR, en este caso la variación de los resultados con respecto a la línea base no difiere en más de un 5% de media. A partir del valor 2000 de umbral la tasa FRR se mantiene inmutable en ambas evaluaciones y la tasa FAR se incrementa en un 6,5%. Podríamos decir que el umbral óptimo estaría comprendido entre 2000 y 3000.

| RESULTADOS FISHERFACES |         |         |
|------------------------|---------|---------|
| UMBRAL                 | FRR (%) | FAR (%) |
| 1000                   | 57,00%  | 0,00%   |
| 2000                   | 2,50%   | 16,00%  |
| 3000                   | 0,00%   | 18,50%  |
| 4000                   | 0,00%   | 18,50%  |
| 5000                   | 0,00%   | 18,50%  |
| 6000                   | 0,00%   | 18,50%  |

Tabla 7: Resultados alternativa con FFR

Si nos fijamos en la gráfica (Figura 43: Variación FRR y FAR en FFR con alternativa) la pendiente es prácticamente la misma y su evolución en el tiempo es similar. Pero si nos fijamos en los resultados han empeorado con respecto a nuestra línea base. Un comparativa podemos encontrarla en la Tabla 8: Comparativa línea base/alternativa en FFR, donde se puede observar que la variación de los resultados es leve, también se han marcado en verde los mejores resultados obtenidos.

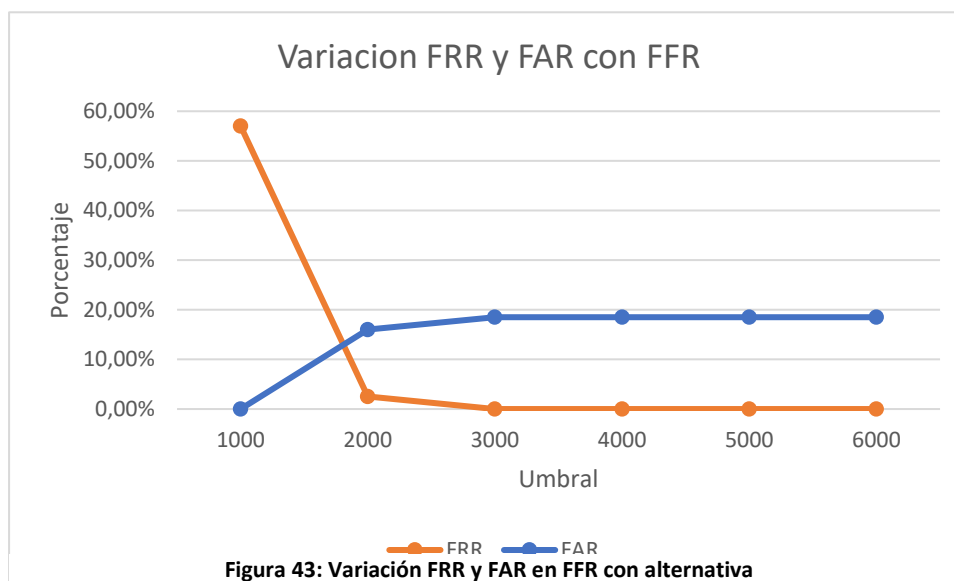


Figura 43: Variación FRR y FAR en FFR con alternativa

|        | LINEA BASE |         | ALTERNATIVA |         |
|--------|------------|---------|-------------|---------|
| UMBRAL | FRR (%)    | FAR (%) | FRR (%)     | FAR (%) |
| 1000   | 43,50%     | 0,00%   | 57,00%      | 0,00%   |
| 2000   | 0,00%      | 12,00%  | 2,50%       | 16,00%  |
| 3000   | 0,00%      | 12,00%  | 0,00%       | 18,50%  |
| 4000   | 0,00%      | 12,00%  | 0,00%       | 18,50%  |
| 5000   | 0,00%      | 12,00%  | 0,00%       | 18,50%  |
| 6000   | 0,00%      | 12,00%  | 0,00%       | 18,50%  |

Tabla 8: Comparativa línea base/alternativa en FFR

### 5.3.3 Método LBPH

Al igual que en apartados anteriores, en la Tabla 9: Resultados alternativa con LBPH podemos encontrar los resultados obtenidos al aplicar la alternativa a la solución. Como calco del apartado anterior los resultados no difieren en demasiado con los obtenidos en la línea base, incluso con un valor de umbral de 100 la tasa FAR ha disminuido levemente, como hándicap FFR se ha incrementado un 7%.

| RESULTADOS LBPH |         |         |
|-----------------|---------|---------|
| UMBRAL          | FRR (%) | FAR (%) |
| 20              | 100,00% | 0,00%   |
| 50              | 52,50%  | 0,00%   |
| 100             | 7,00%   | 11,00%  |
| 500             | 0,00%   | 17,00%  |
| 1000            | 0,00%   | 17,00%  |
| 2000            | 0,00%   | 17,00%  |
| 3000            | 0,00%   | 17,00%  |
| 4000            | 0,00%   | 17,00%  |
| 5000            | 0,00%   | 17,00%  |
| 6000            | 0,00%   | 17,00%  |

Tabla 9: Resultados alternativa con LBPH

La gráfica como era de esperar contiene similitudes con su antecesora, donde a partir de valores de umbral superiores a 100 ambas tasas permanecen invariantes. A las vitas de los resultados el umbral óptimo estaría comprendido entre 100 y 200.

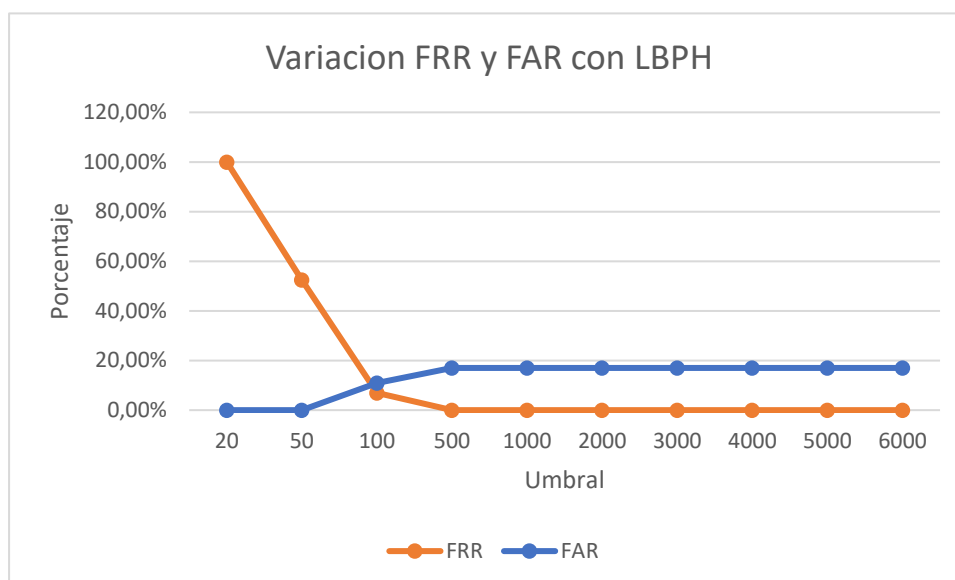


Figura 44: Variación FRR y FAR en LBPH con alternativa

Para una evaluación mejor de los datos obtenidos en la Tabla 10: Comparativa línea base/alternativa en LBPH tenemos una comparativa de los resultados obtenidos con la alternativa y la línea base.

|        | LINEA BASE |         | ALTERNATIVA |         |
|--------|------------|---------|-------------|---------|
| UMBRAL | FRR (%)    | FAR (%) | FRR (%)     | FAR (%) |
| 20     | 100,00%    | 0,00%   | 100,00%     | 0,00%   |
| 50     | 24,00%     | 1,50%   | 52,50%      | 0,00%   |
| 100    | 0,00%      | 11,50%  | 7,00%       | 11,00%  |
| 500    | 0,00%      | 11,50%  | 0,00%       | 17,00%  |
| 1000   | 0,00%      | 11,50%  | 0,00%       | 17,00%  |
| 2000   | 0,00%      | 11,50%  | 0,00%       | 17,00%  |
| 3000   | 0,00%      | 11,50%  | 0,00%       | 17,00%  |
| 4000   | 0,00%      | 11,50%  | 0,00%       | 17,00%  |
| 5000   | 0,00%      | 11,50%  | 0,00%       | 17,00%  |
| 6000   | 0,00%      | 11,50%  | 0,00%       | 17,00%  |

Tabla 10: Comparativa línea base/alternativa en LBPH

# CAPÍTULO 6

## CONCLUSIONES Y TRABAJOS FUTUROS

En este capítulo se presentan los objetivos conseguidos en la consecución del proyecto y los compara con los planteados inicialmente. Así mismo se presentan a su vez las posibles vías futuras de investigación.

### 6.1 CONCLUSIONES

Finalmente llegamos al apartado más complicado de todos los elaborados, el apartado de conclusiones. Preguntareis porqué es complicado, lo es porque tienes que enfrenar todo el optimismo con el que se empieza todo nuevo proyecto contra la cruda realidad de los resultados obtenidos, que pueden ser buenos, malos o simplemente iguales. Este enfrentamiento entre optimismo y realidad debe de quedar plasmado en las siguientes líneas de este apartado.

Todo proyecto se comienza con mucho optimismo en cuanto a los requisitos que se pretenden conseguir. No siempre se consiguen todos ellos, pero sí se pueden aproximar bastante a los del principio. Para poder establecer el grado de éxito del PFM, se debe de conocer los requisitos conseguidos y el grado de dificultad de los mismos. Los objetivos cubiertos con la conclusión del proyecto han sido:

- Establecer una metodología de evaluación de resultados en reconocimiento facial.
- Evaluación de los diferentes algoritmos implementados en OpenCV.
- Establecer una visión genérica y global del estado de la biometría facial y los algoritmos más utilizados en el mercado.

Como hándicap, no se pudieron mejorar los resultados de partida. Pero tampoco se han empeorado en demasía. Es muy posible que, con la elección de otra base de datos, más genérica, se mejoraran los resultados obtenidos debido a que los métodos están implementados para reducir el ruido en las imágenes (como los ejemplos de imágenes mostrados en el Capítulo 4). En este caso las imágenes eran más genéricas y no tan optimizadas con reducción de ruido, luz y fondo... en estos casos los resultados se hubieran mejorado al no tener las imágenes un tratamiento previo.

Cabe resaltar que una parte para el desarrollo del proyecto ha sido debido a la reutilización de código existente (como la librería OpenCV). Tanto la implementación y los experimentos realizados comparten código OpenCV facilitando el desarrollo.

Pienso que una parte importante para el desarrollo de un proyecto es una buena elección de la base de datos, en este caso ha sido una elección demasiado buena, ya que los resultados ofrecidos eran excelentes. Ofrecían poco margen de mejora, aunque lo aprendido durante la realización del proyecto mejora con creces los resultados.

Desde mi punto de vista, la parte más importante que aprendí con el desarrollo de este proyecto fue inculcarme una metodología de trabajo aplicable a cualquier tipo de proyecto, trabajo, experimento... que, pensándolo bien, es de lo que se trata de conseguir con la realización de un PFM.

Para concluir, decir que me encuentro satisfecho con el trabajo realizado. No en si por los resultados, pero si por el conocimiento adquirido. Que, desde mi punto de vista, es lo más importante que podemos conseguir, conocimiento para poder avanzar sin tropezarnos tanto.



## 6.2 TRABAJOS FUTUROS

Para concluir, presentamos en este apartado las posibles futuras vías de investigación y de mejora:

- Evaluar el algoritmo con una base de datos más genérica, quizás en *frames* de video.
- Establecer un auto ajuste del umbral, para obtener siempre el umbral óptimo para iteración.
- Implementar nuevas técnicas de procesado de imagen para evitar falsos positivos o detecciones faciales erróneas.
- Comprobar su funcionamiento en detección en tiempo real de individuos.



# REFERENCIAS

[1] *12th IEEE International Conference on Automatic Face & Gesture Recognition, FG 2017*, Washington, DC, USA, May 30 - June 3, 2017. IEEE Computer Society 2017, ISBN 978-1-5090-4023-0.

[2] *AVBPA'03 Proceedings of the 4th international conference on Audio and video based biometric person authentication Guildford*, UK, June 09 - 11, 2003.

[A. V. Oppenheim et al, 1998] Alan V. Oppenheim and Alan S. Willsky, “*Signals and Systems, Second Edition*”. Prentice Hall, 1998.

[A. J. Mansfield et al, 2002] A. J. Mansfield and J. L. Wayman, *Best Practices in Testing and Reporting Performance of Biometric Device*, August 2002.

[Anil K. Jain et al. 2004a] Anil K. Jain, Fellow, Arun Ross and Salil Prabhakar, “*An Introduction to Biometric Recognition*”.

[Anil K. Jain et al. 2004b] Anil K. Jain, Sharath Pankanti, Salil Prabhakar, Lin Hong, Arun Ross and James L. Wayman, “*Biometrics: A Grand Challenge*”.

[Anil K. Jain et al, 2000] Anil K. Jain, S. Prabhakar, Lin Hong and S. Pankanti, “*Filterbank-Based Fingerprint Matching*”.

[Anil K. Jain et al. 2000] Anil K. Jain, Robert P.W. Duin and Jianchang Mao, “*Statiscal Pattern Recognition: A Review*”. IEEE transaction sof pattern analysis and machine intelligence, vol. 22, no. 1, January 2000.

[Anil K. Jain et al, 2005] A.K. Jain, K. Nandakumar and A. Ross: “*Score normalization in multimodal biometric systems*”. Pattern Recognition, vol. 38, no. 12, pp. 2270-2285. 2005.

[B. Bustos et al, 2003] B. Bustos, G. Navarro and E. Chávez, “*Pivot Selection Techniques for Proximity Searching in Metric Spaces*”. Pattern Recognition. Lett. Vol. 24, no. 14, pp. 2357-2366, 2006.

[Baudat G. et al, 2001] Baudat G. and Anouar F, “*Kernel-based Methods and Function Approximation*”. MEI, Mars Electronics International. USA 2001.

[C. Conde et al, 2008] Cristina Conde and Enrique Cabello, “*Biometric Facial Verification using 2D and 3D Information*”.

[C. L. Tisse et al, 2002] Christel-loïc TISSE, Lionel MARTIN, Lionel TORRES, Michel ROBERT, “*Person identification technique using human iris recognition* “. Advanced System Technology STMicronics – Université de Montpellier.

[C. L. Tisse, 2003] C.L. Tisse : “*Contribution à la vérification biométrique de personnes par reconnaissance de l’iris*”. Ph. D, Université de Montpellier. 2003.

[C. E. Vivaracho et al, 2006] C. E. Vivaracho, V. Cardenoso y C. Gonzalez, “*Mejoras en un Sistema de Reconociendo de Locutor Basado en RNA Mediante Entrenamiento con Normalización de Canal*”.

[D. Carrero et al., 2009] D. Carrero and B. Ruiz, “*Front-End for a Biometric Extended Experiment Platform (BEEP)*”. Proyecto Fin de Carrera, Julio 2009.

[D. A. Reynold et al, 1995] Douglas A. Reynold y Richard C. Rose, “*Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models*”.

- [D. A. Reynold, 2000] Douglas A. Reynold, Thomas F. Quatieri and Robert B. Dunn, *Speaker Verification Using Adapted Gaussian Mixture Models*.
- [D. Dessimoz et al, 2005] D. Dessimoz, J. Richiardi, C. Champod and A. Drygajlo: “*Multimodal biometrics for identity documents*”. University of Lausanne and EPFL (European Biometrics Portal). Technical Report PFS 341-08.05. 2005.
- [F. Monrose et al, 2000] F. Monrose and A.D. Rubin: “*Keystroke dynamics as a biometric for authentication*”. *Future Generation Computer Systems*, vol. 16, pp. 351-359. Elsevier. 2000.
- [F. M. Rodríguez et al, 2004] F. M. Rodríguez and F. J. Suárez López, “*Identificación Dactilar Basada en Filtros Gabor*”. Universidad de Vigo.
- [Guangming Lu et al, 2002] Guangming Lu, D. Zhang and K. Wang, “*Palmprint Recognition using Eigenpalms Features*”. *Pattern Recognition Letters* 24 (2003) 1463–1467.
- [G. Yang and T. S. Huang, 1994] G. Yang and T.S:Huang. Human face detection in complex background. *Pattern Recognition*, 1994.
- [Hasan Demirel et al, 2009] Hasan Demirel and Gholamreza Anbarjafari, “*Iris Recognition System Using Combined Histogram Statistics*”. Department of Electrical and Electronic Engineering, Eastern Mediterranean University, Gazimağusa, KKTC, Mersin 10, TURKEY.
- [H. D. Chang et al, 1993] H.D. Chang, J.F. Wang, and H.-M. Suen, “*Dynamic Handwritten Chinese Signature Verification*,” in *Proceedings second IEEE International Conference on Document Analysis and Recognition*, 1993, pp. 258–261.
- [Huang et al, 1997] Kai Huang and Hong Yan: “*Off-line signature verification based on geometric feature extraction and neural network classification*”. *Pattern Recognition*, vol. 30, no. 1, pp. 9-17. 1997.
- [Hao Feng et al, 2003] Hao Feng and Chan Choong Wah, “*Online signature verification using a new extreme point warping technique*”.
- [Hong et al, 1999] L. Hong, A.K. Jain and S. Pankanti: “*Can multibiometrics improve performance?*” *Proceedings of AutoID '99*, pp. 59-64. 1999.
- [J. Mercer, 1909] J. Mercer. “*Functions of Positive and Negative Type and Their Connection with the Theory of Integral Equations*”. *Philosophical Transactions of the Royal Society, London*, A 209:415–446, 1909.
- [J. Daugman, 1992] J. Daugman, “*High confidence personal identification by rapid video analysis of iris texture*”, *Proc. Of the IEEE, International Carnahan conf. on security technology*, 1992.
- [J. L. Wayman, 2001] J. L. Wayman, *Fundamentals of Biometric Authentication Technologies*.
- [J. Richiardi et al, 2003] J. Richiardi and A. Drygajlo, “*Gaussian mixture models for on-line signature verification*,” in *International Multimedia Conference, Proceedings 2003 ACM SIGMM workshop on Biometrics methods and applications*, Berkeley, USA, Nov. 2003, pp. 115–122.
- [Jeff P. Foster et al, 2003] Jeff P. Foster, Mark S. Nixon and Adam Prügel-Bennett, “*Automatic Gait Recognition using Area-Based metrics*”. Department of Electronics and Computer Science, ISIS, University of Southampton, Southampton, SO17 1BJ, UK.
- [J. González et al, 2004] J. González, J. Fires, D. Ramos and J. Ortega, “*Bayesian Analysis of Fingerprint, Face and Signatures Evidence*”.

- [J. Coetzer et al, 2004] J. Coetzer, B. M. Herbst and J. A. du Preez, “Offline Signature Verification Using the Discrete Radon Transform and a Hidden Markov Model”.
- [J. Fierrez et al, 2007] Julián Fierrez, Javier Ortega-García, Daniel Ramos, Joaquín González-Rodríguez, “HMM-based on-line signature verification: Feature extraction and signature modeling”.
- [José García-Hernández et al, 2005] José García-Hernández y Roberto Paredes, “Biometric Identification Using Palmprint Local Features”.
- [J. Galbally et al, 2009] J. Galbally, J. Fierrez, J. Ortega-García, C. McCool and S. Marcel, “Hill-Climbing Attack to an Eigenface-Based Face Verification System”.
- [Koichi Ito et al, 2005] Koichi Ito, Ayumi Morita, T. Aoiki, T. Higuchi, H. Nakajima and K. Kobayashi, “A Fingerprint Algorithm Using Phase-Based Image Matching for Low-Quality Fingerprints”.
- [K. Fukushima, 1988] K. Fukushima, “Neocognitron: A Hierarchical Neural Network Capable of Visual Pattern Recognition”. Neural Networks, Vol.1, pp. 119-130, 1988.
- [K. J. Kirchberg et al, 2002] K.J. Kirchberg, O. Jesorsky and R. W. Frischholz, Genetic Model Optimization for Hausdorff Distance-Based Face Localization. Springer, Lecture Notes in Computer Science, LNCS-2359, pp. 103–111, Copenhagen, Denmark, June 2002.
- [K. C. Chan et al, 2003] K. C. Chan, Y. S. Moon and P. S. Cheng, “Fast Fingerprint Verification using Sub-region of Fingerprint Images”.
- [Karthik Nandakumar et al. 2009] Karthik Nandakumar, Anil K. Jain and Arun Ross, “Fusion in Multibiometric Identification Systems: What about the missing data?”.
- [L. Sirovich et al, 1987] L. Sirovich and M. Kirby, “Low-dimensional procedure for the characterization of human faces” Journal of the Optical Society of America, vol. 4, no. 3, pp. 519–524, 1987.
- [Liang Wang et al, 2003] Liang Wang, Tieniu Tan, Weiming Hu and Huazhong Ning, “Automatic Gait Recognition Based on Statistical Shape Analysis”. IEE Transaction of Image Processing, vol. 12, NO. 9, September 2003.
- [L. Puente et al, 2008] L. Puente Rodríguez, A. García Crespo, M. J. Poza Lara and B. Ruiz Mezcuá: “Study of different fusion techniques for multimodal biometric authentication”. Proceedings of the 2008 IEEE International Conference on Wireless & Mobile Computing, Networking & Communication. pp. 666-671. 2008.
- [M. Turk and A. Pentland, 1991] M. Turk y A. Pentland. Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, 3(1):71–86, 1991.
- [M. Pandit et al, 1998] M. Pandit and J. Kittler, “Feature selection for a DTW-based speaker verification system,” in Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP’98), vol. 2, 1998, pp. 769–772.
- [Moon H. and Phillips J, 1998] Moon H. and Phillips J., Analysis of PCA-based Faced Recognition Algorithms. In Boyer K. and Phillips J., editors, *Empirical Evaluation Techniques In Computer Vision*, IEEE Computer Society Press, Los Alamitos, CA, 1998.
- [M. H. Yang et al, 2002] M.H. Yang, D.J. Kriegman, y N. Ahuja. Detecting faces in images: A survey. *Pattern Analysis and Machine Intelligence*, 24:34–58, 2002.
- [M. Padilla et al, 2003] M. Padilla and Z. Fan, “Digital Image Processing Project-Automatic Face Detection Using Color Based Segmentation and Template/Energy Thresholding”, Department of Electrical Engineering, Stanford University.

[M. Welling et al, 2009] M.Welling, "Fisher Linear Discriminant Analysis" Department of Computer Science, University of Toronto, 2009.

[N. Yager et al, 2004] N. Yager and A. Amin: "Fingerprint verification based on minutiae features: a review". Pattern Analysis and Application, vol. 17, pp. 94-113. 2004.

[O. Costell et al, 2006] O. Costell, J. M. Badia and G. Torres, "Biometric Identification System based in Keyboard System".

[Pai-Hsuen Chen et al, 2003] Pai-Hsuen Chen, Chih-Jen Lin, and Bernhard Schölkopf, "A Tutorial on Support Vector Machines".

[P. Belhumeur et al, 1997] Belhumeur, P. N., Hespanha, J. P., & Kriegman, D. J. (1997). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *Pattern Analysis and Machine Intell., IEEE Trans. on*, 17(9), 711–720.

[Papageorgiou et al, 1998] Papageorgiou, C., Oren, M., and Poggio, T. 1998. A general framework for object detection. In *International Conference on Computer Vision*.

[P. J. Phillips et al, 2000a] P.J Phillips, A. Martin, C.L. Wilson and M. Przybocky: "An introduction to evaluating biometric systems". Computer, vol. 33, no. 2. 2000.

[P. J. Phillips et al, 2000b] P. J. Phillips, H. Moon, S. A. Rizvi and P. J. Rauss, "The FERET Evaluation Methodology for Face Recognition Algorithms". IEEE Transactions on Patterns Analysis and Machine Intelligence, Vol. 22, NO. 10, October 2000.

[P. Viola and M.J. Jones, 2001] P. Viola and M.J. Jones, 2001. *Robust Real-Time Face Detection*. International Journal of Computer Vision 57(2), 137–154, 2004\_c 2004 Kluwer Academic Publishers. Manufactured in The Netherlands.

[R. and Poggio et al, 1993] R. and Poggio, T Brunelli, Face recognition: features versus templates. IEEE Trans. PAM1, 1993, vol. 15.

[R. P. Wildes, 1994] R. P. Wildes, J. C. Asmuth, G.L. Green and S.C. Hsu, "A system for automated iris recognition ", IEEE paper, 1994.

[R. Fisher, 1936] R. Fisher. *The use of multiple measures in taxonomic problems*. Ann. Eugenics, 7:179-188, 1936.

[Raphael et al, 1974] D. E. Raphael and J.R. Young: "Automated Personal Identification". SRI International. 1974.

[R. Lienhart and J. Maydt, 2002] R. Lienhart y J. Maydt. An extended set of haar-like features for rapid object detection. En *Proceedings IEEE International Conference on Image Processing, volumen 1*, paginas 900–903, 2002.

[R. Capelli et al, 2006] R. Cappelli, D. Maio, D. Maltoni, J.L. Wayman and A.K. Jain.: "Performance evaluation of fingerprint verification systems". IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, no.1, pp. 3-18. 2006.

[R. S. Reíllo et al, 2007] R. S. Reíllo, C. S. Ávila y M<sup>a</sup>B. F. Saavedra, "La Evaluacion de la Seguridad en Sistemas Biométricos".

[S. Young et al, 2009] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev and P. Woodland, "The HTK Book". 1995-1999 Microsoft Corporation, 2001-2009 Cambridge University Engineering Department. March, 2009.

- [**Shrikanth Mohan, 2004**] Shrikanth Mohan, “*Iris Recognition for Personal Identification*”, Intelligent Systems, Electrical Engineering, Clemson University
- [**T. Ahonen et al, 2004**] Ahonen, T., Hadid, A., and Pietikainen, M. *Face Recognition with Local Binary Patterns*. Computer Vision - ECCV 2004 (2004), 469–481.
- [**T. Rhee et al, 2001**] T. Rhee, S. Cho, and J. Kim, “*On-line signature verification using modelguided segmentation and discriminative feature selection for skilled forgeries*,” in Proceedings Sixth International Conference on Document Analysis and Recognition, September 2001, pp. 645–649.
- [**T. Ganchev et al. 2002**] Todor Ganchev, Anastasios Tsopanoglou, Nikos Fakotakis and George Kokkinakis, *Probabilistic Neural Networks Combined with GMMs for Speaker Recognition Over Telephone Channels*.
- [**T. Marwala et al, 2005**] T. Marwala, S. Chakraverty, U. Mahola, “*Neural Networks and Support Vector Machines For Fault Identification In Cylinders*”, Proceedings of International Symposium on Neural Networks and Soft Computing in Structural. Engineering, Cracow, Poland, 2005.
- [**V. Vapnik, 1995a**] V. Vapnik, “*The Nature of Statistical Learning Theory*”. Springer-Verlag, New York, 1995.
- [**V. S. Nalwa, 1997**] Vishvjit S. Nalwa, “*Automatic On-line Signature Verification*”.
- [**Watanabe, 1985**] S. Watanabe, *Pattern Recognition: Human and Mechanical*. New York: Wiley, 1985.
- [**W. Zhao et al, 2003**] W. Zhao, R. Chellappa, J. Phillips, and A. Rosenfeld: “*Face recognition: A literature survey*”. ACM Computing Surveys, vol. 35, no. 4, pp. 399-458. 2003.
- [**W. Zhao et al, 2003**] R. Chellappa, P.J. Phillips, A. Rosenfeld W. Zhao, Face Recognition: A literature survey. ACM Computing Surveys, 2003, vol. 35.
- [**Xiaoguang Lu, 2005**] Xiaoguang Lu, Image Analysis for Face Recognition. Dept. of Computer Science & Engineering Michigan State University, East Lansing, MI, 48824.
- [**Y. Freund and R. E. Schapire, 1995**] Y. Freund y R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *European Conference on Computational Learning Theory*, paginas 23–37, 1995.