

UNIVERSIDAD MIGUEL HERNÁNDEZ DE ELCHE
ESCUELA POLITÉCNICA SUPERIOR DE ELCHE
INGENIERÍA INDUSTRIAL



**“ANÁLISIS DEL PREPROCESADO DE
IMÁGENES EN EL RECONOCIMIENTO
DE CARAS BASADO EN PCA”**

PROYECTO FIN DE CARRERA

Diciembre – 2010

AUTOR: Sergio Rodríguez Gozábez

DIRECTOR/ES: María Asunción Vicente Ripoll

“Es ist nicht genug zu wissen - man muss auch anwenden.

Es ist nicht genug zu wollen - man muss auch tun”

Johann Wolfgang von Goethe

AGRADECIMIENTOS

En primer lugar, quiero dar las gracias a mi directora del proyecto M^a Asunción Vicente Ripoll y a César Fernández Peris por su atención, disposición y ayuda durante la realización del presente proyecto, sin los cuales éste no habría sido posible.

Agradecer también a todos mis amigos y en especial a mi familia su dedicación y apoyo, pero sobre todo a mi madre, a quien debo hoy por hoy todo lo que soy.

ÍNDICE GENERAL

AGRADECIMIENTOS.....	xi
ÍNDICE GENERAL	xiii
ÍNDICE DE FIGURAS	xv
ÍNDICE DE TABLAS	xix
1. INTRODUCCIÓN	1
1.1. Motivación	3
1.2. Enfoque del proyecto	4
1.3. Objetivos	4
1.4. Organización del trabajo	5
2. INTRODUCCIÓN AL RECONOCIMIENTO DE CARAS.....	7
2.1. Descripción de los sistemas de reconocimiento de patrones.....	9
2.1.1. Módulo de adquisición de datos	10
2.1.2. Módulo de selección y extracción de características.....	11
2.1.3. Módulo de clasificación: aprendizaje y entrenamiento	12
2.1.3.1. Aprendizaje supervisado	13
2.1.3.2. Aprendizaje no supervisado	14
2.2. Análisis de Componentes Principales, PCA.....	15
2.2.1. PCA por maximización de la varianza	16
2.2.2. PCA y Eigenfaces	19
2.2.2.1. Cálculo de las “Eigenfaces”	21
2.2.2.2. Clasificación de caras mediante “eigenfaces”	24
2.2.3. PCA por minimización del error cuadrático medio	27
2.2.4. PCA, SVD, whitening y ICA.....	27
2.2.5. Algunas consideraciones sobre PCA	32
3. PREPROCESADO DE IMÁGENES	33
3.1. Realce o manipulación del contraste	35

3.1.1. Histograma, brillo y contraste	35
3.1.2. Ecualización del histograma.....	38
3.1.3. Realzado basado en el dominio frecuencial	40
3.1.4. Corrección gamma.....	40
3.2. Técnicas de suavizado.....	42
3.2.1. Filtros gaussianos.....	43
3.3. Detección de caras: Viola-Jones	45
3.3.1. Características	45
3.3.2. Algoritmo de aprendizaje	47
3.3.3. Cascada de clasificadores	47
3.4. Máscara elipsoidal	49
4. HERRAMIENTAS	51
4.1. Entorno de desarrollo	53
4.2. Bases de datos	55
4.2.1. Base de datos Olivetti (ORL Database).....	55
4.2.2. Base de datos UMH.....	57
4.2.3. Base de datos Yale B Extended	59
5. EXPERIMENTOS.....	61
5.1. Programa principal.....	63
5.2. Experimentos con la base de datos ORL	68
5.2.1. Experimentos iniciales	68
5.2.2. Resultados.....	70
5.2.3. Observaciones	80
5.3. Experimentos con la base de datos UMH	81
5.4. Experimentos con la base de datos Yale B Extended	87
6. CONCLUSIONES	93
6.1. Conclusiones	95
BIBLIOGRAFÍA	97

ÍNDICE DE FIGURAS

Figura 2.1:	Esquema modular de un sistema de reconocimiento de patrones.....	10
Figura 2.2:	Esquema modular de un sistema de reconocimiento de objetos mediante imágenes bidimensionales.	11
Figura 2.3:	Diversos enfoques en el reconocimiento estadístico de patrones.	14
Figura 2.4:	Ejemplo ilustrativo de la transformación PCA.	16
Figura 2.5:	Ejemplo de un eigenspacio donde se muestran las diferentes clases (correspondientes a las caras de cada uno de los individuos) y la imagen de entrada a identificar (punto en negro de la gráfica).	21
Figura 2.6:	Ejemplo de PCA con imágenes. Arriba, la primera fila recoge las imágenes en color de 9 conocidos pilotos de motociclismo, y son las imágenes de entrada utilizadas para generar el eigenspacio. En la segunda fila, se muestran los ocho vectores propios obtenidos y la imagen media. En el resto de filas se muestran las reconstrucciones junto con su error de reconstrucción.	26
Figura 2.7:	Dos ejemplos artificiales: un conjunto de datos subgaussiano (1ª fila) y un conjunto supergaussiano (2ª fila) (a), ambos transformados por PCA (b), whitening (c) y utilizando el modelo ICA a través del algoritmo FastICA (d). La última figura de cada fila (e) muestra el conjunto de datos original con las direcciones de PCA e ICA.....	31
Figura 2.8:	Ejemplos de conjuntos de datos diferentes cuyas componentes principales son iguales.....	32
Figura 3.1:	Ejemplo de ecualización del histograma.	39
Figura 3.2:	Imagen original e imágenes con diversos factores gamma.....	41
Figura 3.3:	Fuentes de error a)original; b)gaussiano; c)impulsional; d)multiplicativo... ..	42
Figura 3.4:	(De izda. a dcha. y de arriba abajo) imagen con filtrado gaussiano con valores de $\sigma=1,2,3,4$	44
Figura 3.5:	Ejemplos de las características tipo Haar empleadas por Viola y Jones. .	45
Figura 3.6:	El valor de la imagen integral en un punto (x, y) es la suma de los píxeles de la región superior izquierda.....	46
Figura 3.7:	Suma del área rectangular D	46

Figura 3.8:	Arquitectura en cascada.....	48
Figura 3.9:	Imagen original sin y con máscara elipsoidal.....	49
Figura 4.1:	Imágenes de la ORL Database.....	56
Figura 4.2:	Imágenes de los 17 individuos que forman la librería de caras UMH.	58
Figura 4.3:	Imágenes de un individuo de la librería de caras UMH.	58
Figura 4.4:	Imágenes de los 10 sujetos de la "Yale B Database" original.	59
Figura 5.1:	Diagrama de bloques con los preprocesados a realizar.	65
Figura 5.2:	Ejemplo de los distintos estadios de la cadena de preprocesado. Fila superior: una imagen de entrada es preprocesada mediante la corrección Gamma, el filtrado DoG y la ecualización del contraste. Fila inferior: procedimiento idéntico a la primera fila aplicando una máscara elipsoidal tras cada preprocesado.....	65
Figura 5.3:	Aciertos de imágenes originales e imágenes preprocesadas mediante ecualización del histograma; en rojo se muestran los aciertos correspondientes a las imágenes originales y en verde los de las imágenes tras la ecualización del histograma.	68
Figura 5.4:	Aciertos de imágenes originales e imágenes preprocesadas mediante ecualización del histograma para valores de $n_train=[1,5]$	69
Figura 5.5:	Clasificación correspondiente a la imagen test nº4.	71
Figura 5.6:	Clasificación correspondiente a la imagen test nº5.	71
Figura 5.7:	Clasificación correspondiente a la imagen test nº6.	72
Figura 5.8:	Clasificación correspondiente a la imagen test nº7.	72
Figura 5.9:	Clasificación correspondiente a la imagen test nº8.	73
Figura 5.10:	Clasificación correspondiente a la imagen test nº9.....	73
Figura 5.11:	Clasificación correspondiente a la imagen test nº10.....	74
Figura 5.12:	Porcentaje de aciertos en imágenes con diferentes preprocesados sin la detección Viola-Jones.	75
Figura 5.13:	Porcentaje de aciertos en imágenes con diferentes preprocesados con la detección Viola-Jones.	75
Figura 5.14:	Comparativa de aciertos para los casos de clasificación más eficaces.	76
Figura 5.15:	Aciertos de las imágenes originales recortadas.	80
Figura 5.16:	Porcentaje de aciertos en imágenes con diferentes preprocesados sin la detección Viola-Jones. Conjunto de imágenes de test [2,10].	81

Figura 5.17:	Porcentaje de aciertos en imágenes con diferentes preprocesados sin la detección Viola-Jones. Conjunto de imágenes de test [2,30].....	82
Figura 5.18:	Porcentaje de aciertos en imágenes con diferentes preprocesados sin la detección Viola-Jones. Conjunto de imágenes de test [11,30].....	82
Figura 5.19:	Porcentaje de aciertos en imágenes con diferentes preprocesados con la detección Viola-Jones. Conjunto de imágenes de test [2,10].....	83
Figura 5.20:	Porcentaje de aciertos en imágenes con diferentes preprocesados con la detección Viola-Jones. Conjunto de imágenes de test [2,30].....	83
Figura 5.21:	Porcentaje de aciertos en imágenes con diferentes preprocesados con la detección Viola-Jones. Conjunto de imágenes de test [11,30].....	84
Figura 5.22:	Comparativa de aciertos para el conjunto de imágenes de test [2,10]. .	84
Figura 5.23:	Comparativa de aciertos para el conjunto de imágenes de test [2,30]. .	85
Figura 5.24:	Comparativa de aciertos para el conjunto de imágenes de test [11,30].	85
Figura 5.25:	Porcentaje de aciertos en imágenes con diferentes preprocesados.	87
Figura 5.26:	Comparativa de aciertos entre los mejores preprocesados.....	88

ÍNDICE DE TABLAS

Tabla 5.1:	Notación empleada para la designación de los preprocesados.	64
Tabla 5.2:	Técnicas de preprocesado empleadas.	64
Tabla 5.3:	Matriz de confusión de las imágenes originales.	70
Tabla 5.4:	Matriz de confusión caso 100.	77
Tabla 5.5:	Matriz de confusión caso 101.	78
Tabla 5.6:	Matriz de confusión caso 111.	79
Tabla 5.7:	Matriz de confusión caso 000.	89
Tabla 5.8:	Matriz de confusión caso 020.	90
Tabla 5.9:	Matriz de confusión caso 021.	91

Capítulo 1

Introducción

1.1. Motivación

El reconocimiento de rostros es un problema que fue considerado desde las primeras etapas de la visión por computador. Éste es un tema que ha tomado mucho auge en los últimos años, gracias a los avances del poder computacional que han permitido implementar algoritmos más complejos utilizando diversas técnicas.

La identificación de características faciales ha recibido un fuerte impulso gracias al avance en la tecnología de video multimedia, propiciándose así un aumento de cámaras en los lugares de trabajo y en el hogar con un reducido coste. El reconocimiento facial puede ser aplicado en el control de acceso a edificios públicos y privados, cajeros automáticos, laboratorios de investigación y también como clave secreta de acceso para el uso de ordenadores personales o terminales móviles de última generación.

Podrían utilizarse estos sistemas para crear bases de datos con las personas que tienen acceso a lugares emblemáticos como los anteriormente descritos.

El proceso de identificación facial se divide en dos tareas: detección y reconocimiento. La primera de ellas, la detección, comprende la localización de una o varias caras dentro de una imagen, ya sea una imagen fija o una secuencia de video. La segunda tarea, el reconocimiento, consiste en la comparación de la cara detectada en la primera fase con otras almacenadas previamente en una base de datos. Estos procesos, detección y reconocimiento, no deberían ser totalmente independientes, puesto que según la forma en la que se detecte una cara, puede resultar imposible su reconocimiento con caras de una base de datos detectadas de forma 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 captura de las imágenes.

1.2. Enfoque del proyecto

El presente trabajo está enfocado a proporcionar una solución al problema del reconocimiento facial mediante la técnica de extracción de características PCA, estudiando el efecto de diferentes preprocesados aplicados sobre las imágenes. Por tanto los esfuerzos de investigación se centran, por una parte, en la obtención de un algoritmo robusto para la identificación de identidades, y por otra, en el desarrollo de diversas técnicas de preprocesado de imágenes que mejoren el rendimiento del algoritmo desarrollado.

1.3. Objetivos

El objetivo de este proyecto es la creación de un software de reconocimiento facial que permita identificar individuos pertenecientes a diferentes bases de datos, bien desarrolladas por laboratorios de investigación, o bien creadas dentro del marco universitario.

Para alcanzar dicho objetivo principal, el presente proyecto ha de llegar a otros que se describen a continuación:

- Documentación sobre el campo de visión artificial, concretamente en el reconocimiento de patrones y las técnicas de preprocesado de imágenes.
- Breve descripción del entorno de desarrollo empleado, así como otras herramientas.
- Desarrollo del algoritmo de reconocimiento facial mediante PCA.
- Implementación al algoritmo principal de los diversos preprocesados de imágenes a emplear.
- Experimentos y resultados. Comprobación del funcionamiento del algoritmo.
- Conclusiones.

1.4. Organización del trabajo

El contenido de la presente memoria está dividido en los siguientes capítulos:

- *Capítulo 1: Introducción.*

Introducción general, presentación de los objetivos y descripción de la estructura de la memoria del proyecto.

- *Capítulo 2: Introducción al reconocimiento de caras.*

Presentación del contenido teórico de la investigación como base para los siguientes capítulos. Se describe el esquema básico de reconocimiento de patrones, haciendo hincapié en el área de reconocimiento de rostros, y se analiza el método de Análisis de Componentes Principales (PCA), objeto principal de este proyecto.

- *Capítulo 3: Preprocesado de imágenes.*

Descripción de los distintos tipos de preprocesados empleados en el tratamiento de imágenes, profundizando en aquellos que serán objeto de los experimentos.

- *Capítulo 4: Herramientas.*

Información básica del entorno computacional en el que se desarrolla el algoritmo, así como de las bases de datos empleadas en los experimentos.

- *Capítulo 5: Experimentos.*

Explicación de los experimentos y los resultados obtenidos para cada una de las bases de datos.

- *Capítulo 6: Conclusiones.*

Revisión de los objetivos propuestos y las conclusiones derivadas de los resultados del capítulo anterior. Posibles trabajos futuros como continuación a este Proyecto Final de Carrera.

Capítulo 2

Introducción al reconocimiento de caras

2.1. Descripción de los sistemas de reconocimiento de patrones

El esquema general de un sistema de reconocimiento de patrones se muestra en la figura 2.1 [1]. A partir de un patrón natural (*en nuestro caso, un objeto tridimensional*), un bloque receptor adquiere información del patrón (*la cámara capta la imagen del objeto*). El mecanismo extractor de características debe extraer la información útil de los datos captados por el sensor en la fase anterior (*color del objeto en la imagen, silueta,...*), eliminando la información redundante e irrelevante. Finalmente un módulo clasificador se encarga de asignar a la categoría apropiada los patrones de clase desconocida a priori (*identificar el objeto captado por la cámara: es una cara, un objeto,...*).

El esquema de la figura 2.1 corresponde a una configuración estadística de reconocimiento de patrones. Esta configuración es la más simple y consiste en representar cada patrón mediante un vector de números, resultantes del muestreo y cuantificación de las señales externas y cada clase por uno o varios patrones prototipo. Un patrón no es más que un punto del espacio de representación de los patrones, que es un espacio de dimensionalidad determinada por el número de variables consideradas. El estudio del conjunto apropiado de variables, la variabilidad de los patrones de una clase, las medidas de semejanza entre patrones y entre patrones y clases constituye el reconocimiento estadístico de patrones. Sin embargo, el reconocimiento de patrones estadístico no considera el contexto del patrón a etiquetar, es decir, la relación entre diferentes patrones. En ocasiones, los patrones complejos pueden descomponerse recursivamente en patrones más simples hasta llegar a componentes básicos. El sistema de reconocimiento que hace uso de esta información de contexto se suele denominar sintáctico. Un ejemplo de configuración de reconocimiento sintáctica es el reconocimiento de voz.

El reconocimiento de objetos mediante visión por computador se enfoca normalmente desde el punto de vista estadístico, aunque existen excepciones a este enfoque que no se tratarán en el siguiente estudio por no ser objeto del mismo.

A continuación se definirá de manera más detallada cada uno de los elementos que forma parte de un sistema de reconocimiento de patrones estadístico:

2.1.1. Módulo de adquisición de datos

La entrada a un sistema de reconocimiento de patrones estadístico es un vector numérico que contiene los valores muestreados y cuantificados de una serie de señales naturales.

Suponiendo patrones M -dimensionales, un patrón \mathbf{x} es una variable aleatoria M -dimensional compuesta por M componentes:

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_M \end{pmatrix} \quad (2.1)$$

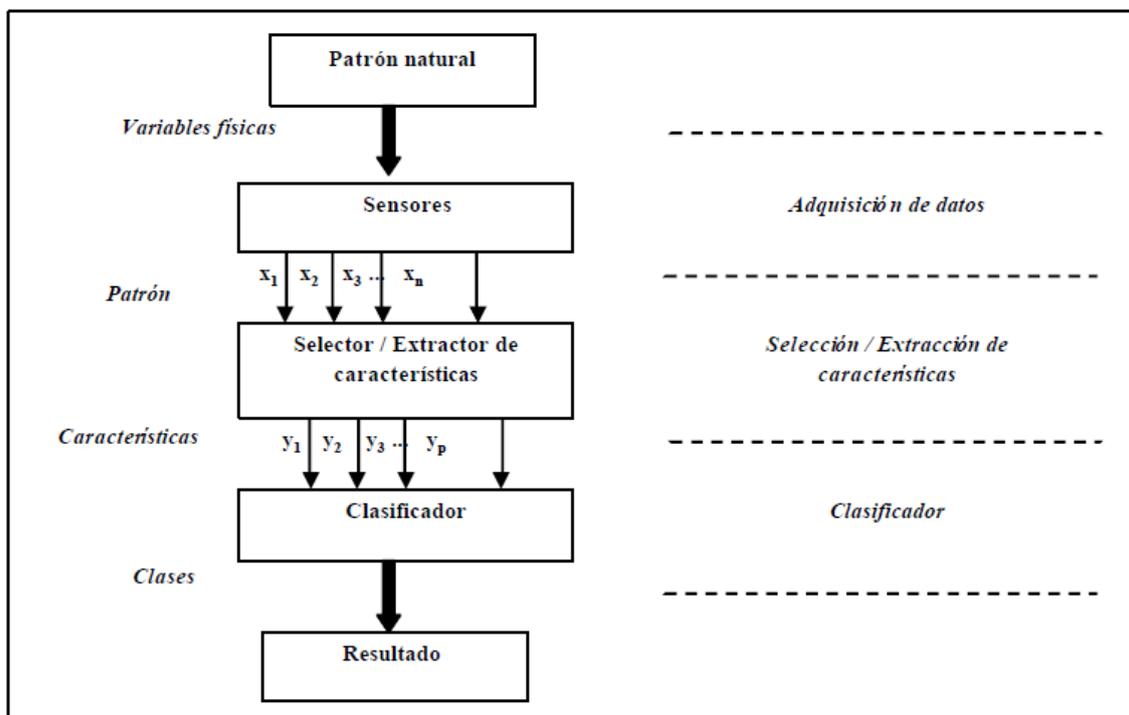


Figura 2.1: Esquema modular de un sistema de reconocimiento de patrones.

2.1.2. Módulo de selección y extracción de características

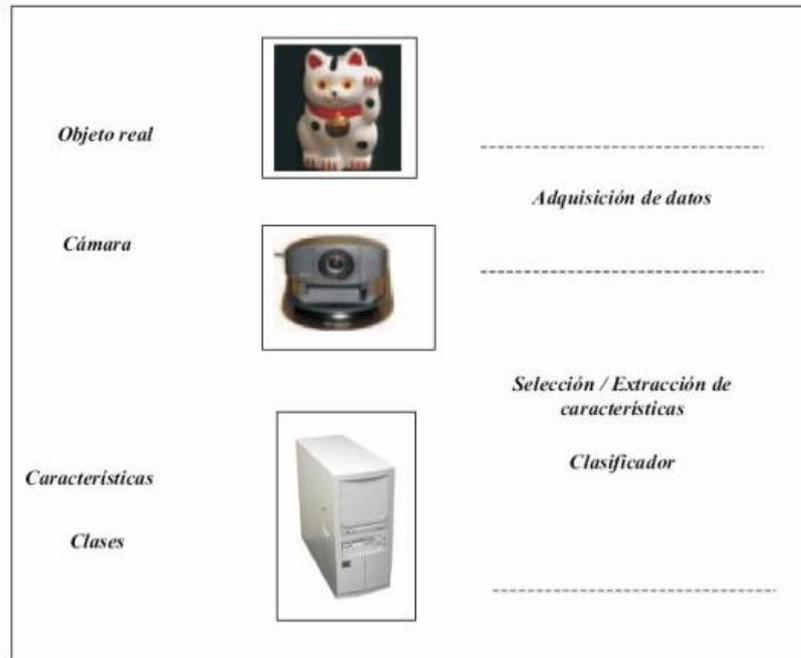


Figura 2.2: Esquema modular de un sistema de reconocimiento de objetos mediante imágenes bidimensionales.

Este módulo trata de resolver el problema de extraer la información más relevante para la clasificación entre la suministrada por los sensores (datos en bruto). Dado un conjunto de patrones M -dimensionales:

$$\mathbf{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_M \end{pmatrix} \quad (2.2)$$

se trata de obtener un nuevo conjunto (características) P -dimensionales:

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_P \end{pmatrix} \quad (2.3)$$

donde $P \leq M$.

Matricialmente, dado un conjunto \mathbf{X} de N patrones M -dimensionales $\mathbf{X} \in \mathfrak{R}^{M \times N}$, se trata de obtener un nuevo conjunto $\mathbf{Y} \in \mathfrak{R}^{P \times N}$, donde $P \leq M$.

2.1.3. Módulo de clasificación: aprendizaje y entrenamiento

El objetivo final de un sistema de reconocimiento de patrones es el etiquetar de forma automática patrones de los cuales desconocemos su clase. El encargado de realizar esta tarea es el clasificador. Suponemos que el sistema dispone de un módulo de adquisición de datos y que se han seleccionado previamente las variables más significativas, cabe ahora definir cuál es el conjunto de etiquetas, es decir, las salidas que debe proporcionar el sistema. El conjunto de todas las posibles etiquetas se denomina como el conjunto de las clases.

Si suponemos que todos los patrones a reconocer son los elementos potenciales de J clases distintas denotadas $\omega_j, j = 1, 2, \dots, J$, llamaremos:

$$\Omega = \{\omega_1, \omega_2, \dots, \omega_J\} \quad (2.4)$$

al conjunto de clases.

Resulta conveniente ampliar el conjunto incorporando una nueva clase, llamada la clase de rechazo (ω_0) que se asigna a todos los patrones para los que no se tiene una certeza aceptable de ser clasificados correctamente en alguna de las clases de Ω .

$$\Omega^* = \{\omega_1, \omega_2, \dots, \omega_J, \omega_0\} \quad (2.5)$$

Una vez establecido el conjunto de clases se procede al diseño del clasificador. Como puede intuirse, el diseño y construcción del clasificador no es una tarea trivial ni directa, ya que el sistema de reconocimiento depende de su buen funcionamiento. Esta es, por tanto, la etapa más importante del sistema de reconocimiento.

El clasificador o regla de clasificación es una función $f : \mathbf{X} \rightarrow \Omega^*$ definida sobre los patrones \mathbf{x} tal que para todo patrón \mathbf{x} , $f(\mathbf{x}) \in \Omega^*$. Los términos aprendizaje y

entrenamiento se utilizan para referirse al proceso de construcción del clasificador. El aprendizaje puede realizarse de dos maneras muy diferentes: aprendizaje supervisado y no supervisado.

2.1.3.1. Aprendizaje supervisado

Un aprendizaje supervisado requiere disponer de un conjunto de patrones de los cuales se conoce su clase cierta. A este conjunto se le denomina conjunto de entrenamiento. Los clasificadores obtenidos a partir de este tipo de aprendizaje se denominan clasificadores supervisados. Disponer de un conjunto de entrenamiento supone que alguien se ha preocupado de etiquetar los patrones de ese conjunto.

En el aprendizaje supervisado, si consideramos que en un caso ideal cada agrupamiento representa a una clase y cada clase tiene asociado un agrupamiento bien diferenciado de los demás, un problema de clasificación supervisada puede plantearse como la búsqueda de las superficies que separan los diferentes agrupamientos. Estas superficies se denominan superficies de decisión. Éstas determinan regiones de decisión de forma que cada clase tiene asociada una región en el espacio de características y la decisión sobre la clase a asignar a un nuevo patrón se hará en base a la región en la que éste se encuentra en el espacio.

La búsqueda de estas superficies (análogamente, regiones) de decisión se puede abordar de dos maneras, dependiendo de si se conoce o supone un determinado modelo estadístico para las clases. Si se supone un completo conocimiento a priori de la estructura estadística de las clases, el aprendizaje se reduce a la estimación de los parámetros determinan las funciones de densidad de probabilidad de las clases. Los clasificadores construidos bajo esta suposición se conocen como clasificadores paramétricos.

Si no se supone un determinado modelo estadístico, bien por desconocimiento o por la imposibilidad de asumir un modelo paramétrico adecuado, el problema resulta más complejo y se puede abordar desde diferentes perspectivas. Las fronteras de decisión están definidas por los prototipos. Los clasificadores construidos sin esta suposición se conocen como clasificadores no paramétricos.

2.1.3.2. Aprendizaje no supervisado

El aprendizaje no supervisado se realiza a partir de un conjunto de patrones del que no se desconoce su clase cierta. En ocasiones, ni siquiera se conoce el número de clases. Básicamente se traduce en encontrar agrupamientos. El objetivo suele ser el de verificar la validez del conjunto de clases para una clasificación supervisada. Las técnicas utilizadas suelen denominarse métodos de agrupamiento o clustering.

Resulta evidente que el reconocimiento de objetos reales tridimensionales a partir de imágenes bidimensionales cabe situarlo en un marco de aprendizaje supervisado, ya que sí se dispone de un conjunto de patrones de los que se conoce su clase cierta. Las técnicas no supervisadas están destinadas a aplicaciones de reconocimiento más complejas donde se desconocen las clases de los patrones e incluso a veces no se sabe a priori cuántas clases pueden existir.

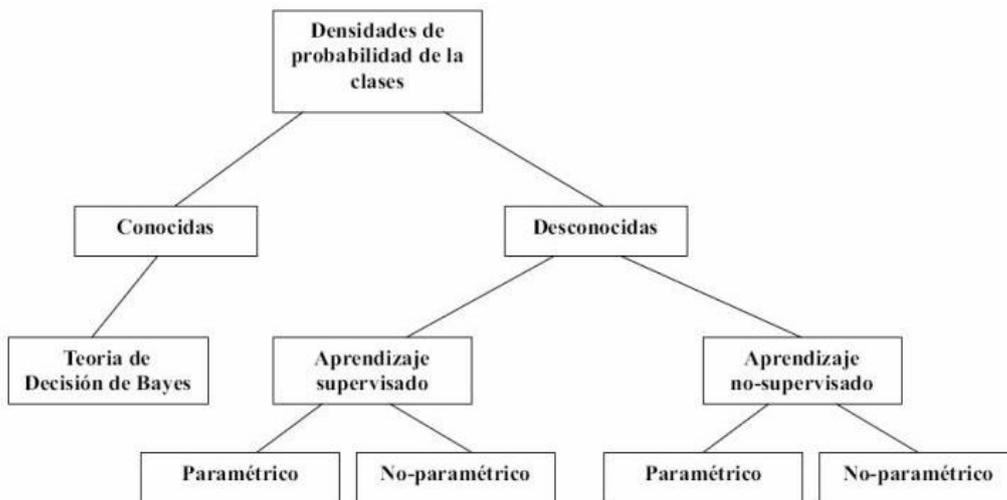


Figura 2.3: Diversos enfoques en el reconocimiento estadístico de patrones.

2.2. Análisis de Componentes Principales, PCA

El Análisis de Componentes Principales (PCA) junto con la relacionada Transformada de Karhunen-Loève, o de Hotelling, o la factorización SVD, son técnicas clásicas muy utilizadas en el análisis estadístico de datos, la extracción de características y la compresión de datos. En el campo de reconocimiento de patrones cabe destacar el éxito del PCA en el reconocimiento e identificación de caras humanas a partir de los trabajos de Turk y Pentland en [2] y Kirby y Sirovich en [3].

El PCA construye una transformación lineal que escoge un nuevo sistema de coordenadas para el conjunto original de datos en el cual la varianza de mayor tamaño del conjunto de datos es capturada en el primer eje (llamado el Primer Componente Principal), la segunda varianza más grande es el segundo eje, y así sucesivamente. En la figura 2.4 se ilustra un ejemplo gráfico de transformación PCA. Para construir esta transformación lineal debe construirse primero la matriz de covarianza o matriz de coeficientes de correlación. Debido a la simetría de esta matriz existe una base completa de vectores propios de la misma. La transformación que lleva de las antiguas coordenadas a las coordenadas de la nueva base es precisamente la transformación lineal necesaria para reducir la dimensionalidad de datos. Además las coordenadas en la nueva base dan la composición en factores subyacentes de los datos iniciales.

Una de las ventajas de PCA para reducir la dimensionalidad de un grupo de datos, es que retiene aquellas características del conjunto de datos que contribuyen más a su varianza, manteniendo un orden de bajo nivel de los componentes principales e ignorando los de alto nivel. El objetivo es que esos componentes de bajo orden a veces contienen el "más importante" aspecto de esa información.

Este apartado contiene una descripción detallada del análisis de componentes principales utilizando dos planteamientos diferentes del problema: por maximización de la varianza y por minimización del error cuadrático medio.

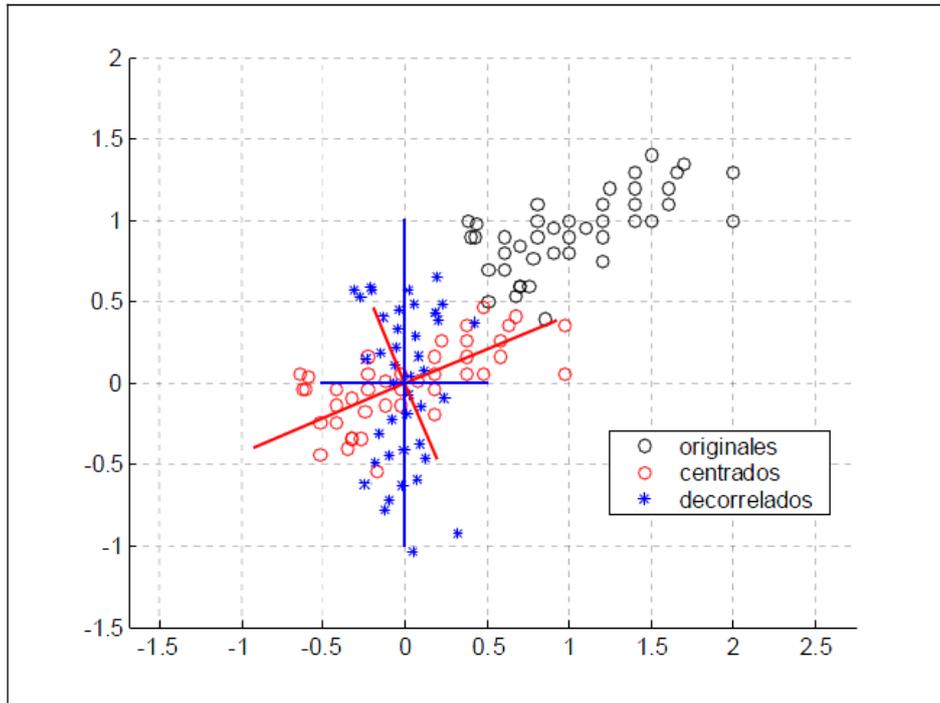


Figura 2.4: Ejemplo ilustrativo de la transformación PCA.

2.2.1. PCA por maximización de la varianza

En este caso se deriva PCA por maximización de la varianza en la dirección de los vectores principales.

Supongamos que se dispone de N vectores \mathbf{x}_j de dimensión M alineados en una matriz de datos $\mathbf{X} \in \mathcal{R}^{M \times N}$. Sea \mathbf{u} dirección de \mathcal{R}^M . La proyección del vector \mathbf{x}_j sobre el vector \mathbf{u} puede calcularse de la siguiente forma:

$$p_j = \langle \mathbf{x}_j, \mathbf{u} \rangle = \mathbf{u}^T \mathbf{x}_j = \sum_{i=1}^M u_i x_{ij} \quad (2.6)$$

El objetivo es encontrar la dirección \mathbf{u} que maximiza la varianza de las proyecciones de todos los vectores de entrada $\mathbf{x}_{ij}, j = 1, \dots, N$. La función a maximizar es:

$$J^{PCA}(\mathbf{u}) = \sigma^2(p_j) = E\{p_j^2\} \quad (2.7)$$

Si la media de las proyecciones se calcula según el siguiente estimador:

$$\bar{p} = \frac{1}{N} \sum_{j=1}^N p_j = \frac{1}{N} \sum_{j=1}^N \sum_{i=1}^M u_i x_{ij} = \sum_{i=1}^M u_i \mu_i \quad (2.8)$$

la varianza a maximizar es igual a¹:

$$J^{PCA}(\mathbf{u}) = \sigma^2(p_j) = E\{p_j^2\} = \frac{1}{N} \sum_{j=1}^N (p_j - \bar{p})^2 = \mathbf{u}^T \mathbf{C}_x \mathbf{u} \quad (2.9)$$

En la ecuación 2.8, μ_i es la media de la fila i de la matriz de datos \mathbf{X} . Así el vector μ está formado por las medias de todas las filas:

$$\mu = [\mu_1, \dots, \mu_M]^T = \frac{1}{N} \sum_{j=1}^N x_j \quad (2.10)$$

por lo tanto²:

$$\hat{\mathbf{X}} = \mathbf{X} - \mu \mathbf{1}_{1 \times N} \quad (2.11)$$

y \mathbf{C}_x es la matriz de covarianzas de \mathbf{X} :

$$\mathbf{C}_x = \frac{1}{N} \hat{\mathbf{X}} \hat{\mathbf{X}}^T \quad (2.12)$$

Como ya se ha dicho, el objetivo es maximizar la función 2.9 y además bajo la restricción que $\|\mathbf{u}\| = 1$. Si se utiliza la técnica de los multiplicadores de Lagrange, la función a maximizar es la siguiente:

$$F(\mathbf{u}, \lambda) = \mathbf{u}^T \mathbf{C}_x \mathbf{u} - \lambda (\mathbf{u}^T \mathbf{u} - 1) = \sum_{i=1}^M \sum_{j=1}^M u_i c_{ij} u_j - \lambda \left(\sum_{i=1}^M u_i^2 - 1 \right) \quad (2.13)$$

¹ x_i es el vector columna i de la matriz \mathbf{X} , mientras que x_i^T denota el vector fila i de la matriz \mathbf{X} .

² $\mathbf{1}_{M \times N}$ denota una matriz de dimensión $M \times N$ donde todos los elementos tienen valor unidad.

Una posible solución a este problema de maximización puede ser la siguiente:

$$\frac{\partial F}{\partial u_l} = 0; \quad l = 1, \dots, M$$

$$\mathbf{C}_x \mathbf{u} = \lambda \mathbf{u} \quad (2.14)$$

Por lo tanto, para encontrar los valores de \mathbf{u} y λ que maximizan 2.13 se deben calcular los vectores propios y los valores propios de la matriz de covarianzas \mathbf{C}_x . El valor propio mayor es igual a la máxima varianza, mientras que su correspondiente vector propio determina la dirección de máxima varianza.

Para diagonalizar la matriz de covarianzas \mathbf{C}_x se realiza la descomposición en valores singulares (SVD) de la matriz \mathbf{C}_x :

$$\mathbf{C}_x = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^T \quad (2.15)$$

en la matriz anterior la matriz ortonormal $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_R] \in \mathcal{R}^{M \times R}$ contiene los vectores propios $\mathbf{u}_1, \dots, \mathbf{u}_R$ en sus columnas y la matriz $\mathbf{\Lambda} \in \mathcal{R}^{R \times R}$ contiene los valores propios $\lambda_1, \dots, \lambda_R$ en su diagonal. Los valores propios y sus correspondientes vectores propios se ordenan de mayor a menor según el valor de los valores propios, es decir, $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_R$. De esta forma y con este orden, la mayor variabilidad de los datos está reflejada en los primeros vectores propios. Por lo tanto, los vectores propios se suelen denominar *vectores principales* (o también *ejes principales* o *direcciones principales*).

Esta forma de calcular los vectores principales y los valores principales es la más simple y la más utilizada. Sin embargo, debido a que el tamaño del vector de datos, M , es muy grande, la matriz de covarianzas es demasiado grande $\mathbf{C}_x \in \mathcal{R}^{M \times M}$ y su diagonalización no es factible computacionalmente. Se plantea por tanto una solución: si el número de vectores de datos es menor que el tamaño de estos vectores ($N < M$), puede realizarse el análisis PCA de la matriz de datos original utilizando el método propuesto por Murakami y Kumar [4] y conocido en la literatura por Truco de Turk y Pentland [2] y

que consiste en un simple truco algebraico. De esta forma, en lugar de utilizar la matriz de covarianzas $\mathbf{C}_x \in \mathfrak{R}^{M \times M}$ se utiliza la matriz de covarianzas $\mathbf{C}_x' \in \mathfrak{R}^{N \times N}$:

$$\mathbf{C}_x' = \frac{1}{N} \widehat{\mathbf{X}}^T \widehat{\mathbf{X}} \quad (2.16)$$

Los vectores propios y los valores propios de la matriz de covarianzas \mathbf{C}_x pueden calcularse a partir de los vectores propios \mathbf{u}_i' y los valores propios λ_i' de \mathbf{C}_x' de la siguiente manera:

$$\lambda_i = \lambda_i'; \quad i = 1, \dots, N \quad (2.17)$$

$$\mathbf{u}_i = \frac{\widehat{\mathbf{X}} \mathbf{u}_i'}{\sqrt{N \lambda_i'}}; \quad i = 1, \dots, N \quad (2.18)$$

La matriz \mathbf{C}_x' es mucho más pequeña que la matriz original \mathbf{C}_x cuando $N \ll M$. Por lo tanto la descomposición en vectores y valores propios de la matriz original $\mathbf{C}_x \in \mathfrak{R}^{M \times M}$ se reduce a la descomposición más factible de $\mathbf{C}_x' \in \mathfrak{R}^{N \times N}$.

2.2.2. PCA y Eigenfaces

El sistema desarrollado por Matthew Turk y Alex Pentland [2] permite localizar y realizar un seguimiento de la cabeza de un individuo y reconocer al mismo comparando las características de la cara con las de los otros individuos ya conocidos.

El sistema descompone imágenes de caras en un conjunto de imágenes características llamadas “*eigenfaces*”, las cuales se toman como componentes principales del conjunto inicial de imágenes de entrenamiento. Estas “*eigenfaces*” representan las características principales de cada individuo y son los vectores propios del conjunto de imágenes; no se corresponden necesariamente con rasgos tales como los ojos, la nariz o las orejas. El reconocimiento se lleva a cabo proyectando una nueva imagen sobre el espacio formado

por las “*eigenfaces*”, denominado “*face space*”³, y clasificándola comparando su posición en ese espacio con las posiciones de los individuos ya conocidos.

En términos matemáticos, el objetivo consiste en encontrar los vectores propios de la matriz de covarianzas del conjunto de imágenes, tratando a la imagen como un punto (o vector) en un espacio de dimensión enorme, donde cada coordenada corresponde al valor de un píxel de la imagen original.

La imagen de cada individuo puede ser representada mediante una combinación lineal de las “*eigenfaces*”. Cada cara puede aproximarse utilizando únicamente las “mejores *eigenfaces*”, es decir, aquellas que tienen los mayores valores propios y por tanto, los de máxima varianza. Estas M “mejores *eigenfaces*” crean un subespacio M -dimensional de todas las imágenes posibles.

Esta aproximación al reconocimiento de caras implica las siguientes operaciones iniciales:

Fase de entrenamiento.

1. Adquisición de la colección de imágenes iniciales (“conjunto de entrenamiento”).
2. Calcular las “*eigenfaces*” del set de entrenamiento, teniendo en cuenta las M imágenes que corresponden a los mayores valores propios. Estas M imágenes definen el “*face space*”.
3. Calcular la correspondiente distribución en el espacio M -dimensional para cada individuo, proyectando las imágenes en el “*face space*”.

Una vez realizada la fase de entrenamiento se procede al reconocimiento de nuevas caras.

³ En este trabajo se utiliza el término “*face space*” para referirse al eigenespacio.

Fase de test.

1. Proyectar la nueva imagen de entrada sobre el “face space” y determinar si ésta es una cara, comprobando si es próxima a este espacio.
2. Si es una cara, utilizar un clasificador para determinar la clase más próxima a la que pertenece la nueva imagen.

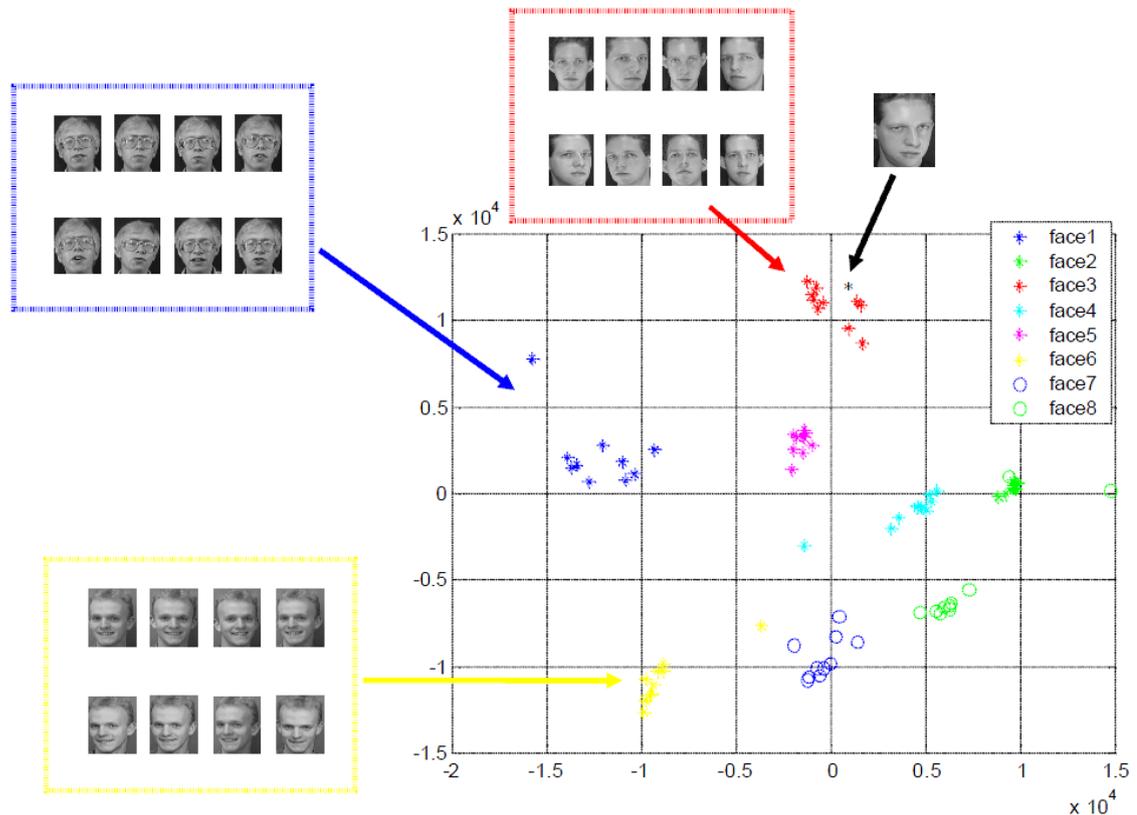


Figura 2.5: Ejemplo de un eigenspace donde se muestran las diferentes clases (correspondientes a las caras de cada uno de los individuos) y la imagen de entrada a identificar (punto en negro de la gráfica).

2.2.2.1. Cálculo de las “Eigenfaces”

Sea una imagen $I(x, y)$ un array bidimensional de $N \times N$ valores de intensidad (8 bits). Una imagen puede ser considerada como un vector de dimensión N^2 , de modo que la típica imagen de 256×256 se convierte en un vector de dimensión 65536, o lo que es lo

mismo, un punto en el espacio 65536-dimensional. Por tanto, un conjunto de imágenes será un conjunto de puntos en un espacio de dimensión enorme.

La idea principal del análisis consiste en encontrar los vectores que formarán el “face space”. Cada vector tiene un tamaño de N^2 , y describe una imagen de $N \times N$. Estos vectores denominados “eigenfaces” son una combinación lineal de las imágenes de las caras originales, ya que son los vectores propios de la matriz de covarianzas.

Sea el conjunto de imágenes de entrenamiento $\Gamma_1, \Gamma_2, \dots, \Gamma_M$. La imagen media del conjunto viene definida por la siguiente expresión:

$$\Psi = \frac{1}{M} \sum_{n=1}^M \Gamma_n \quad (2.19)$$

Cada imagen difiere del resto por el vector:

$$\Phi_i = \Gamma_i - \Psi \quad (2.20)$$

Este gran conjunto de vectores es el objeto de estudio del análisis, el cual busca un conjunto de M vectores ortonormales \mathbf{u}_n que mejor definen la distribución de los datos.

El k -ésimo vector \mathbf{u}_k es elegido de modo que:

$$\lambda_k = \frac{1}{M} \sum_{n=1}^M (\mathbf{u}_k^T \Phi_n)^2 \quad (2.21)$$

es un máximo, sujeto a:

$$\mathbf{u}_l^T \mathbf{u}_k = \delta_{lk} = \begin{cases} 1, & \text{si } l = k \\ 0, & \text{en otro caso} \end{cases} \quad (2.22)$$

Los vectores \mathbf{u}_k y los escalares λ_k son los vectores propios y los valores propios, respectivamente, de la matriz de covarianzas:

$$C = \frac{1}{M} \sum_{n=1}^M \Phi_n \Phi_n^T = AA^T \quad (2.23)$$

donde la matriz $A = [\Phi_1, \Phi_2, \dots, \Phi_M]$. La matriz C de covarianzas es de dimensión $N^2 \times N^2$, por lo que determinar los N^2 vectores propios y valores propios resulta una tarea muy compleja a la hora de tratar con tamaños de imágenes normales.

Sin embargo, si el número de vectores es menor que la dimensión de éstos ($M < N^2$), habrá $M - 1$, en lugar de N^2 , vectores significativos. (Los restantes vectores propios tendrán asociados valores propios igual a cero). De esta forma, se permite reducir el tamaño de la matriz a una dimensión $M \times M$ y luego tomar combinaciones lineales de las imágenes de las caras Φ_i . Considerando los vectores propios \mathbf{v}_i de $A^T A$:

$$A^T A \mathbf{v}_i = \mu_i \mathbf{v}_i \quad (2.24)$$

Multiplicando ambos lados de la igualdad por A , se tiene:

$$AA^T A \mathbf{v}_i = \mu_i A \mathbf{v}_i \quad (2.25)$$

donde se observa que $A \mathbf{v}_i$ son los vectores propios de $C = AA^T$.

Siguiendo este análisis, se construye la matriz $L_{M \times M} = A^T A$, donde $L_{mn} = \Phi_m^T \Phi_n$ y se buscan los vectores propios \mathbf{v}_l de L . Estos vectores determinan combinaciones lineales de las M imágenes del conjunto de entrenamiento que forman las “eigenfaces” \mathbf{u}_l .

$$\mathbf{u}_l = \sum_{k=1}^M \mathbf{v}_{lk} \Phi_k ; \quad l = 1, \dots, M \quad (2.26)$$

De esta forma se reduce el tamaño de las imágenes de N^2 píxeles al tamaño de M , correspondiente al número de imágenes del set de entrenamiento.

2.2.2.2. Clasificación de caras mediante “eigenfaces”

Llegados a este punto, una vez realizado el proceso anterior de cálculo de las “eigenfaces”, una nueva imagen de entrada Γ se transforma en su “eigenface” (proyectada en el “face space”) mediante una simple operación:

$$\omega_k = \mathbf{u}_k^T (\Gamma - \Psi); \quad k = 1, \dots, M' \quad (2.27)$$

siendo M' el número de vectores propios más significativos de la matriz L . A esta operación se la denomina *reconstrucción*.

Estos componentes forman un vector $\Omega^T = \{\omega_1, \omega_2, \dots, \omega_{M'}\}$ que describe la contribución de cada “eigenface” a la representación de la imagen de entrada.

El método más simple utilizado para determinar qué clase de las existentes proporciona una mejor descripción de la imagen de entrada consiste en buscar la clase k que minimiza la distancia Euclídea⁴:

$$\epsilon_k^2 = \|\Omega - \Omega_k\|^2 \quad (2.28)$$

donde Ω_k es el vector que describe la clase k -ésima. Las clases Ω_i se calculan promediando los resultados de la representación de la “eigenface” sobre un pequeño número de imágenes de cada individuo. Una cara se clasifica como perteneciente a una clase k cuando el mínimo ϵ_k está bajo un umbral θ_ϵ , escogido previamente. En otro caso se clasifica como desconocida y puede utilizarse para crear una nueva clase.

La distancia ϵ al cuadrado entre una imagen de entrada y el “face space” es la diferencia al cuadrado de la imagen de entrada ajustada $\Phi = \Gamma - \Psi$ y su proyección en el “face space” $\Phi_f = \sum_{i=1}^{M'} \omega_i \mathbf{u}_i$:

$$\epsilon^2 = \|\Phi - \Phi_f\|^2 \quad (2.29)$$

⁴ Clasificador basado en el vecino más próximo

Del cálculo de ϵ pueden obtenerse cuatro resultados respecto a la imagen de entrada:

- 1) Que sea próxima al “face space” y próxima a una clase. (Individuo reconocido).
- 2) Que sea próxima al “face space” pero no a una clase conocida. (Individuo desconocido pero presente).
- 3) Que sea lejana al “face space” y próxima a una clase.
- 4) Que sea lejana al “face space” y lejana a una clase. (En los casos 3 y 4 la imagen no pertenece a una cara).

En la figura 2.6 se ilustra el proceso de PCA descrito anteriormente. La primera fila recoge las imágenes de 9 pilotos conocidos de motociclismo⁵, correspondientes a las imágenes de entrada utilizadas para la generación del subespacio PCA. Éstas forman la matriz de datos $\Gamma \in \mathbb{R}^{N \times M}$ donde $M = 9$, es el número de pilotos y $N = 273 \times 254 \times 3 = 2080226$, es la dimensión de cada una de ellas. En la segunda fila, se muestran los ocho vectores propios obtenidos a partir del conjunto de imágenes anterior \mathbf{u} , y la media de éste Ψ (última foto de la fila). En el resto de filas se muestran las reconstrucciones de cada una de las imágenes originales a partir de subconjuntos del eigenspacio, es decir, se muestra la reconstrucción de cada imagen utilizando subconjuntos de vectores propios compuestos por 1,2,...,8 vectores; la imagen situada a la izquierda es la reconstruida a partir de un sólo coeficiente y un sólo vector propio, y la situada a la derecha es la reconstruida a partir de todo el subespacio y por tanto es igual a la original. Al final de cada conjunto de imágenes reconstruidas se muestra el error de reconstrucción ϵ de cada una de ellas. Como bien puede apreciarse, el error es diferente para cada imagen, así para el primer piloto (Rossi), el error de reconstrucción es muy elevado en todas las reconstrucciones y sólo se elimina al utilizar el espacio de vectores formado por todo el subespacio, mientras que en para otros pilotos como el quinto (Capirossi), se obtiene una buena reconstrucción con solo 4 vectores propios.

Finalmente se llega a la conclusión de que maximizando la varianza de los primeros vectores propios, el error de reconstrucción está siendo minimizado. Esta consideración da lugar a la posibilidad de plantear el problema de PCA desde el punto de vista de la minimización del error cuadrático medio, objeto del siguiente apartado.

⁵ Los pilotos de las imágenes de izda. a dcha. son: Valentino Rossi, Sete Gibernau, Dani Pedrosa, Jorge Lorenzo, Loris Capirossi, Max Biaggi, Toni Elías, Roberto Rolfo y Manuel Poggiali.

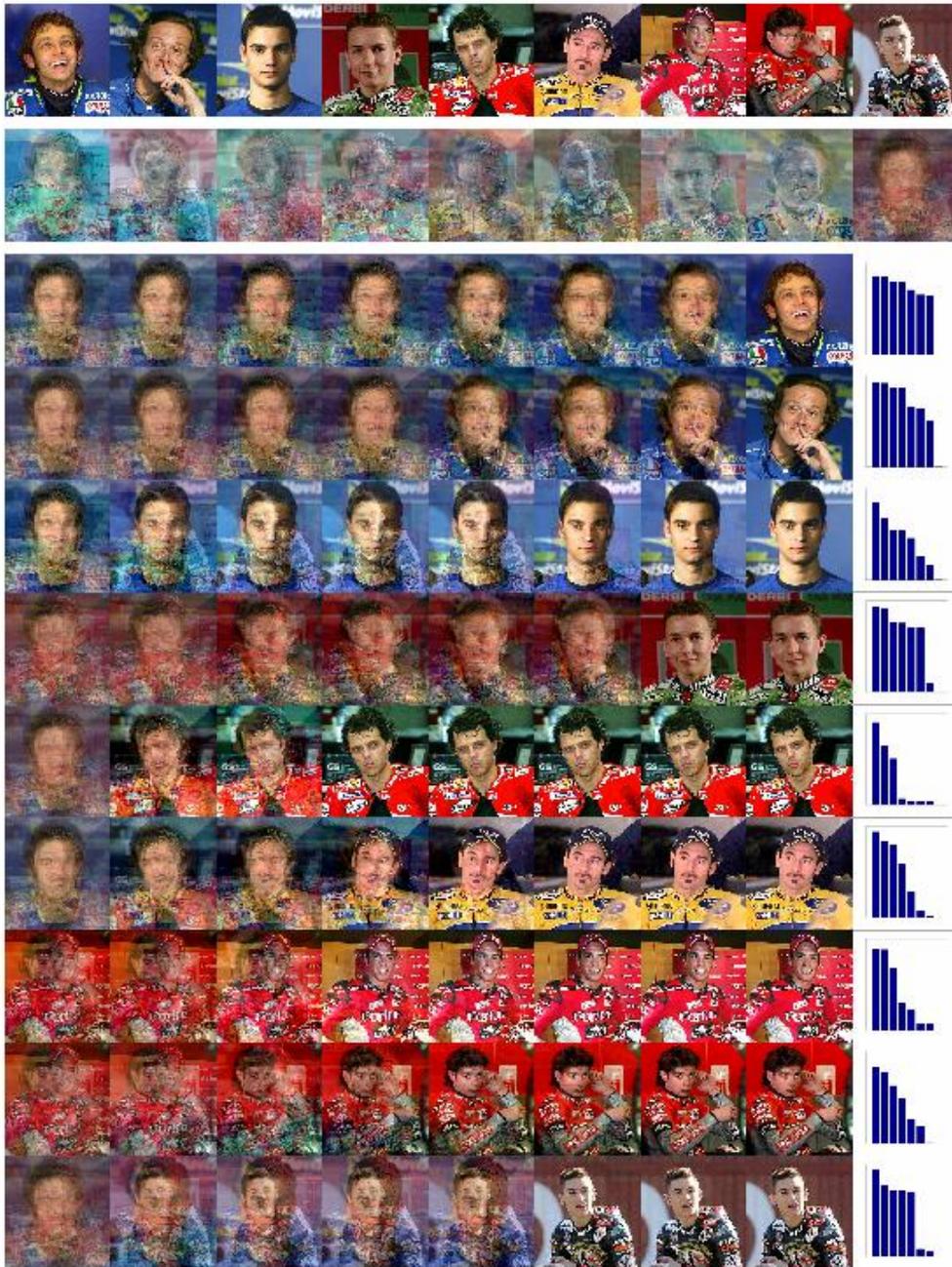


Figura 2.6: Ejemplo de PCA con imágenes. Arriba, la primera fila recoge las imágenes en color de 9 conocidos pilotos de motociclismo, y son las imágenes de entrada utilizadas para generar el eigenespacio. En la segunda fila, se muestran los ocho vectores propios obtenidos y la imagen media. En el resto de filas se muestran las reconstrucciones junto con su error de reconstrucción.

2.2.3. PCA por minimización del error cuadrático medio

Como alternativa a la maximización de la varianza, los vectores principales pueden estimarse a partir de la minimización del error cuadrático medio:

$$J^{PCA}(\mathbf{u}) = E\{\|e\|^2\} = E\{\|\hat{\mathbf{x}} - \hat{\mathbf{y}}\|^2\} = E\{\|\hat{\mathbf{x}} - \sum_{j=1}^N (\mathbf{u}_j^T \hat{\mathbf{x}}) \mathbf{u}_j\|^2\} \quad (2.30)$$

debido a la ortonormalidad de los vectores \mathbf{u}_j la función anterior puede escribirse como:

$$\begin{aligned} J^{PCA}(\mathbf{u}) &= E\{\|\hat{\mathbf{x}}\|^2\} - E\left\{\sum_{j=1}^N (\mathbf{u}_j^T \hat{\mathbf{x}})^2\right\} = \\ &= \text{trace}(\mathbf{C}_x) - \sum_{j=1}^N \mathbf{u}_j^T \mathbf{C}_x \mathbf{u}_j \end{aligned} \quad (2.31)$$

En [5] se demuestra que un mínimo de 2.31, bajo la condición de ortonormalidad de los vectores \mathbf{u}_j , viene dado por la base ortonormal compuesta por los N primeros vectores propios de 2.15. Sin embargo, el criterio de minimización de 2.31 no especifica que la anterior solución sea la única y de hecho cualquier base ortonormal del subespacio de los vectores propios de 2.15 proporciona la misma propiedad de máxima compresión en relación al error cuadrático medio. Aunque esta ambigüedad puede verse como una desventaja, existen otros tipos de criterios en los que un subespacio de PCA es preferible sobre otro, como por ejemplo, el criterio de independencia estadística impuesto por ICA [6].

2.2.4. PCA, SVD, whitening y ICA

Para estudiar la capacidad de separación entre clases de cada uno de los métodos es necesario resaltar las diferencias que existen entre las técnicas de PCA, SVD, ICA y blanqueo de datos o *whitening*.

En la arquitectura de subespacio universal, la matriz de datos original \mathbf{X} , está compuesta por las imágenes de entrenamiento de los diferentes objetos del sistema, de manera que cada columna contiene los valores de los píxeles del vector imagen \mathbf{x} .

Suponiendo que la matriz \mathbf{X} está centrada (se le ha sustraído la imagen media a cada una de sus columnas), la transformada PCA de la matriz \mathbf{X} de tamaño $m \times n$ viene definida por:

$$\mathbf{Y} = \mathbf{U}^T \mathbf{X} \quad (2.32)$$

donde \mathbf{U} es una matriz ortonormal de tamaño $m \times m$. Como se explica en la sección 2.2.1, las componentes principales (las columnas de \mathbf{U}) son las direcciones de máxima varianza de los datos bajo la restricción de ortogonalidad mutua. Los vectores de componentes principales son los vectores propios de la matriz de covarianzas de los datos $\mathbf{C}_x = \frac{1}{N} \widehat{\mathbf{X}} \widehat{\mathbf{X}}^T$ en orden decreciente a sus correspondientes valores propios. El valor propio de mayor valor es igual al valor de la máxima varianza, mientras que su correspondiente vector propio determina la dirección dentro del espacio imagen con esa máxima varianza en los datos. Además, la transformación definida en la ecuación 2.32 proporciona componentes no correladas. Por lo tanto, los vectores de entrada de dimensión elevada y correlados pueden ser eficientemente representados en un espacio de dimensiones inferior donde los vectores no están correlados.

En la ecuación 2.32, \mathbf{Y} es la matriz original de datos proyectada sobre el subespacio PCA definido por los vectores propios. A partir de esa relación, es posible reducir la dimensión de los datos simplemente seleccionando un subconjunto de k vectores propios del conjunto total ($\mathfrak{R}^m \rightarrow \mathfrak{R}^k$),

$$\tilde{\mathbf{Y}} = \tilde{\mathbf{U}}^T \mathbf{X} \quad (2.33)$$

de esta forma $\tilde{\mathbf{Y}}$ es una matriz de tamaño $k \times n$, donde k se puede elegir para que se cumpla que $k \ll m$ y la reducción de dimensión sea considerable.

Es posible ver la transformación desde un punto de vista ligeramente diferente y considerar que \mathbf{U} es una de las matrices de la descomposición SVD de la matriz de datos \mathbf{X} ,

$$\mathbf{X} = \mathbf{U}\mathbf{\Lambda}\mathbf{V}^T \quad (2.34)$$

donde \mathbf{U} y \mathbf{V} son matrices cuyas columnas son ortonormales y su tamaño es $m \times r$ y $n \times r$ respectivamente. $\mathbf{\Lambda}$ es una matriz diagonal de tamaño $r \times r$ cuyos valores singulares son no negativos $\sigma_j, j = 1, 2, \dots, r$, y están ordenados de manera decreciente a lo largo de la diagonal, siendo r el rango de la matriz \mathbf{X} .

De la ecuación 2.34 se deriva que $\mathbf{X}\mathbf{X}^T\mathbf{U} = \mathbf{U}\mathbf{\Lambda}\mathbf{\Lambda}^T$ y $\mathbf{X}^T\mathbf{X}\mathbf{V} = \mathbf{V}\mathbf{\Lambda}^T\mathbf{\Lambda}$, demostrando que las columnas de \mathbf{U} son los vectores propios de $\mathbf{X}\mathbf{X}^T$ y las columnas de \mathbf{V} son los vectores propios de $\mathbf{X}^T\mathbf{X}$.

También es posible la representación de la matriz \mathbf{X} mediante la suma de r matrices de rango unidad,

$$\mathbf{X} = \sum_{j=1}^r \sigma_j \mathbf{u}_j \mathbf{v}_j^T \quad (2.35)$$

Esto implica que los valores singulares nulos pueden ser ignorados, ya que no proporcionan ningún tipo de información. La ecuación 2.36 también muestra que es posible aproximar \mathbf{X} mediante los primeros k vectores propios de \mathbf{U} y \mathbf{V} ,

$$\mathbf{X} \approx \sum_{j=1}^r \sigma_j \mathbf{u}_j \mathbf{v}_j^T = \tilde{\mathbf{U}}\tilde{\mathbf{\Lambda}}\tilde{\mathbf{V}}^T \quad (2.36)$$

Un vector aleatorio y centrado \mathbf{z} se dice que está blanqueado (*white*) si sus elementos están no correlados y tienen varianzas unidad, es decir, que su matriz de covarianzas es igual a la matriz unidad \mathbf{I} . Puesto que la operación de *whitening* puede realizarse mediante un decorrelación seguida de un escalado, puede utilizarse PCA con ese fin, y la ecuación 2.37 define la transformación de *whitening* sobre la matriz de datos original:

$$\tilde{\mathbf{Z}} = \tilde{\mathbf{\Lambda}}^{-1} \tilde{\mathbf{U}}^T \mathbf{X} = \tilde{\mathbf{V}}^T \quad (2.37)$$

Por otro lado, la técnica de ICA intenta expresar los datos originales utilizando vectores aleatorios estadísticamente independientes. La matriz de datos aleatorios observada \mathbf{X} se modela como:

$$\mathbf{X} \approx \mathbf{A}\mathbf{S} \quad (2.38)$$

donde \mathbf{S} es la matriz que contiene los vectores aleatorios estadísticamente independientes y \mathbf{A} es la matriz mezcla. El modelo anterior de la ecuación 2.38 se puede expresar como:

$$\mathbf{S} = \mathbf{W}\mathbf{X} \quad (2.39)$$

donde \mathbf{W} es igual a la matriz pseudoinversa de \mathbf{A} , es decir, $\mathbf{W} = \mathbf{A}^\dagger$. En el modelo ICA, los vectores \mathbf{w}_i se obtienen de forma que las filas de la matriz \mathbf{S} tienen distribuciones no-gaussianas máximas y están mutuamente decorreladas. Una forma sencilla de conseguir esto es primero, blanquear los datos según la ecuación 2.37, y después, buscar una proyección ortogonal no-normal (\mathbf{R}):

$$\mathbf{S} = \mathbf{R}^T \tilde{\mathbf{\Lambda}}^{-1} \tilde{\mathbf{U}}^T \mathbf{X} = \mathbf{R}^T \tilde{\mathbf{Z}} \quad (2.40)$$

por lo tanto, en 2.40 se muestra que ICA es una operación de blanqueo seguida de una rotación, y así, el modelo ICA también puede expresarse de la siguiente forma:

$$\mathbf{X} \approx \tilde{\mathbf{U}} \tilde{\mathbf{\Lambda}} \tilde{\mathbf{V}}^T = \underbrace{\tilde{\mathbf{U}} \tilde{\mathbf{\Lambda}} \mathbf{R}}_{\mathbf{A}} \underbrace{\mathbf{R}^T \tilde{\mathbf{V}}^T}_{\mathbf{S}} = \mathbf{A}\mathbf{S} \quad (2.41)$$

Cabe remarcar el hecho que en el modelo ICA ninguna de las matrices de la transformación (\mathbf{A} o \mathbf{W}) están restringidas a matrices ortogonales. La restricción (exacta [7] o aproximada [8], según la elección del algoritmo) es que la transformación sea decorrelante, esto es, que las filas de la matriz de datos independiente \mathbf{S} sean (exacta o aproximadamente) una transformación ortogonal de los datos blanqueados.

En la figura 2.7 se ilustra gráficamente las transformaciones que producen los métodos PCA, *whitening* e ICA (en el caso de utilizar el algoritmo *FastICA*⁶ [6], [7]) sobre dos conjuntos de datos artificiales de sólo 2 dimensiones. En la primera fila, los datos originales tienen una distribución uniforme sobre un paralelogramo (datos subgaussianos); mientras que los datos originales en la segunda fila siguen una distribución *sparse* (datos supergaussianos).

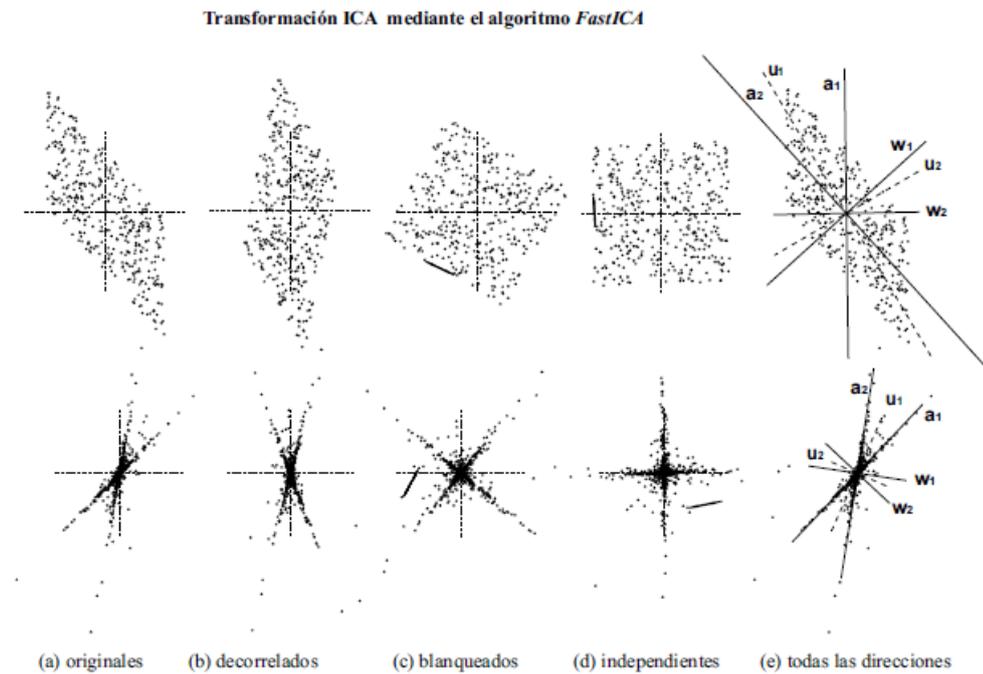


Figura 2.7: Dos ejemplos artificiales: un conjunto de datos subgaussiano (1ª fila) y un conjunto supergaussiano (2ª fila) (a), ambos transformados por PCA (b), whitening (c) y utilizando el modelo ICA a través del algoritmo FastICA (d). La última figura de cada fila (e) muestra el conjunto de datos original con las direcciones de PCA e ICA.

⁶ En el algoritmo *FastICA*, los datos son primero blanqueados y posteriormente rotados, tal y como se describe en la ecuación 1.39

2.2.5. Algunas consideraciones sobre PCA

PCA es una técnica de extracción de características sencilla y poderosa pero que en ocasiones presenta una serie de limitaciones debidas a su naturaleza. La transformación PCA no puede distinguir conjuntos de puntos cuyas estadísticas de segundo orden sean iguales. Esta limitación se ilustra en la figura 2.8 (a), donde se muestran cuatro conjuntos de datos diferentes cuyas componentes principales son iguales.

Además, puesto que se trata de una técnica no supervisada, los ejes principales no indican a priori las direcciones que mejor separan las clases. Por ejemplo, si consideramos el conjunto de datos de la figura 2.8 (d) suponiendo que cada nube de puntos es una clase diferente, y pretendemos clasificar los puntos reduciendo previamente los datos a una sola dimensión, la proyección de los datos sobre la primera componente principal (el eje vertical) produce la confusión de las dos clases, mientras que la proyección sobre la segunda componente principal (el eje horizontal) permite la separación completa entre las dos clases.

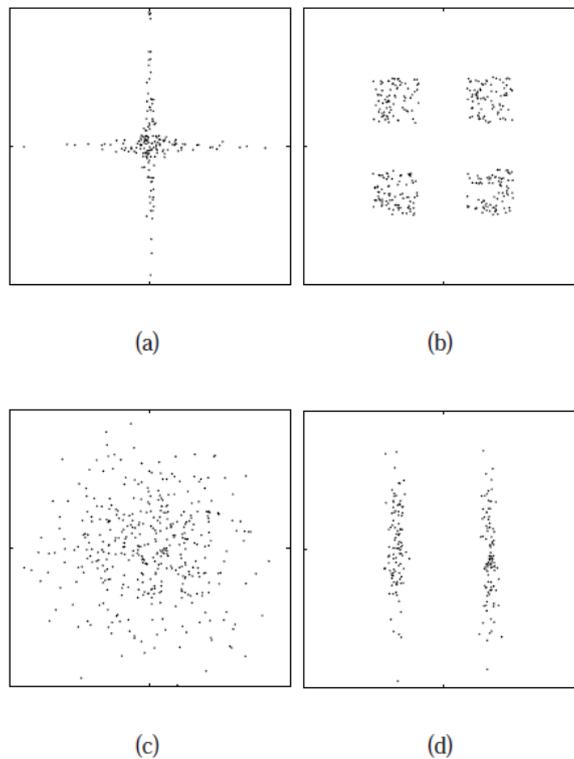


Figura 2.8: Ejemplos de conjuntos de datos diferentes cuyas componentes principales son iguales.

Capítulo 3

Preprocesado de imágenes

3.1. Realce o manipulación del contraste

Las técnicas de realce, entre ellas la ecualización del histograma, pretenden aumentar el contraste de las imágenes, con el fin de mejorar algunas de sus características visuales para las siguientes etapas del análisis automático de las imágenes.

Normalmente, la escena capturada por el sistema de visión se encuentra con unas fuentes de iluminación no colimadas y variables con el tiempo y con objetos que producen reflejos y sombras no deseadas. De otro lado, las ganancias de las cámaras no son lineales. El doble de intensidad radiada en un píxel, no se traduce en el doble de cuantificación en el nivel de gris de ese píxel. Todas estas imperfecciones deben ser corregidas tanto en la formación de las imágenes como con el uso de técnicas preprocesado de realce. Ambas deben ser utilizadas.

Estas técnicas de realce también se aplican cuando se pretende utilizar técnicas de segmentación. El objetivo de este procesado es facilitar las tareas de partición de la imagen. Con este fin se aumenta el contraste entre los objetos de la imagen.

Muchas de las aplicaciones de realzado se fundamentan en operaciones punto a punto, es decir, no tienen en cuenta el entorno de vecindad del píxel. Para su estudio se requiere introducir los conceptos de histograma, brillo y contraste. Los fundamentos de estas técnicas están en la adecuación del rango dinámico de la imagen y en la ecualización del histograma [9].

3.1.1. Histograma, brillo y contraste

El histograma es una función discreta que contabiliza el número de ocurrencias de cada nivel de gris presenta en una imagen. Se representa en un diagrama donde en el eje de abscisas está el nivel de gris y en el de ordenadas la frecuencia de cada nivel de gris en la imagen. Si al histograma se le divide por el número de píxeles de la imagen se obtendrá la función de probabilidad muestral de cada nivel de gris en la imagen:

$$h(i) = \text{número de ocurrencia del nivel de gris } i \text{ en la imagen}$$

$$p(i) = \frac{h(i)}{M \cdot N} \quad (3.1)$$

Siendo M y N el número de filas y columnas respectivamente de la imagen, $h(i)$ es la función histograma y $p(i)$ su probabilidad. Un histograma no dice nada sobre la disposición espacial de las distintas intensidades, por lo que dos imágenes diferentes pueden tener igual histograma. Sin embargo, el histograma habla sobre el tipo de adquisición que ha sido realizado.

El histograma proporciona información estadística de la imagen. Esta información es bastante útil para conocer cómo se ha producido el proceso de formación de la imagen. Así, se define el brillo de la imagen como el valor medio de la imagen que coincide con el valor medio del histograma:

$$\text{Brillo} = \mu = \frac{1}{M \cdot N} \sum_{x=1}^M \sum_{y=1}^N f(x, y) = \sum_{i=0}^{I-1} i \cdot p(i) \quad (3.2)$$

Donde $f(x, y)$ devuelve el nivel de gris del píxel con coordenadas (x, y) , I es el número de grises que se ha empleado en el proceso de cuantificación de la imagen. Una imagen que presente un valor de brillo mayor que otro sobre la misma escena, indica que el nivel de radiación medio que ha llegado a la cámara es mayor en una imagen que en otra, bien porque se disminuyó el número F de diafragma o porque se varió el tiempo de integración. Este efecto se observa en el histograma como un desplazamiento neto hacia la derecha del eje de abscisa. A medida de que la imagen parezca más clara, el histograma se mueve hacia valores más altos de los niveles de grises. La variación del brillo se efectúa con la operación aritmética de suma o resta a cada píxel de una cantidad determinada.

La varianza del histograma también es asociada al contraste de la imagen:

$$\text{Contraste} = \sigma^2 = \frac{1}{M \cdot N} \sum_{x=1}^M \sum_{y=1}^N (f(x, y) - \mu)^2 = \frac{1}{M \cdot N} \sum_{x=1}^M \sum_{y=1}^N f(x, y) =$$

$$= \frac{1}{M \cdot N} \sum_{i=0}^{I-1} (i - \mu)^2 \cdot h(i) = \sum_{i=0}^{I-1} (i - \mu)^2 \cdot p(i) \quad (3.3)$$

El contraste indica la dispersión de los niveles de grises en la imagen. Una imagen con poco contraste indica que hay poca variabilidad de los niveles de grises en la imagen. Su efecto se muestra en un histograma muy concentrado, con un rango dinámico pequeño. El rango dinámico es la variación de niveles de grises en la imagen. Otro estadístico interesante es la función energía:

$$\text{Energía} = \sum_{i=0}^{I-1} p(i)^2 \quad (3.4)$$

Ésta indica el grado de dispersión de grises en la imagen. Si hubiese un único nivel de gris la energía sería máxima y de valor uno. A medida de que aumente el número de niveles de grises existentes en la imagen disminuirá la energía. La función inversa a la energía es la entropía:

$$\text{Entropía} = - \sum_{i=0}^{I-1} p(i) \cdot \log(p(i)) \quad (3.5)$$

A mayor entropía más desorden en la imagen, esto es, más niveles de grises participan en la imagen. La entropía será máxima cuando todos los niveles de grises sean equiprobables.

Se dice que una imagen está saturada cuando su histograma presenta valores muy altos en sus extremos del rango dinámico. Las imágenes con bajo contraste o saturadas tienen pérdidas de información en sus adquisiciones. La solución está en una nueva formación de la escena con valores diferentes de los parámetros de la captura. Las técnicas de preprocesado digital no aumentan el nivel de información, aunque acondicionan la imagen para las siguientes etapas de interpretación de las imágenes.

3.1.2. Ecuación del histograma

En cualquier función de densidad probabilística que es transformada a uniforme se maximiza la explotación de la información. Este fenómeno que rige a los procesos continuos también se puede aplicar sobre información discreta. La conversión de cualquier forma de histograma a uno uniforme, si bien es cierto que no aumentará la información, como ya se ha comentado anteriormente, si mejorará la imagen para las siguientes etapas. Recuerde que el valor máximo de la entropía se dará cuando el histograma sea uniforme.

El fundamento se basa en realizar una conversión en el histograma de forma que la probabilidad de cualquier nivel de gris en la imagen sea idéntica; es decir, obtener un histograma con una distribución uniforme. Desde el punto de vista del histograma, esta transformación producirá que el número de ocurrencias de un nivel de gris sea el mismo valor siempre, $h(i) = \frac{M \cdot N}{I}$.

Para obtener esta técnica de preprocesado se partirá de la función de distribución de la imagen. Ésta se define, para variables aleatorias discretas, como el sumatorio de probabilidades desde el inicio de la variable hasta un valor prefijado. Por tanto, el sumatorio de la función en todo el espectro de la variable aleatoria independiente es la unidad:

$$F(r) = \sum_{i=0}^r p(i) \quad (3.6)$$

$$F(I - 1) = \sum_{i=0}^{I-1} p(i) = 1 \quad (3.7)$$

En este caso, la probabilidad de la variable aleatoria corresponderá con la probabilidad de que aparezca el nivel gris i en la imagen. La función de distribución de una imagen estará determinada por el histograma acumulado relativizado por el tamaño de la imagen.

Sea $F(r)$ la función de distribución de una imagen cualquiera. Un histograma uniforme se caracterizará por tener:

$$F(r') = \sum_{i=0}^{r'} p(i) = (r' + 1) \frac{1}{I} \quad (3.8)$$

Siendo $F(r')$ la función de distribución uniforme. La conversión del histograma se establece igualando las funciones de distribución que posee la imagen con la función de distribución deseada. Con esta relación se establece la función de transferencia del histograma, de r a r' :

$$F(r') = F(r) \Rightarrow (r' + 1) \frac{1}{I} = F(r) \Rightarrow r' = F(r) \cdot I - 1 \quad (3.9)$$

El carácter discreto de una variable supondrá una aproximación a una densidad uniforme.

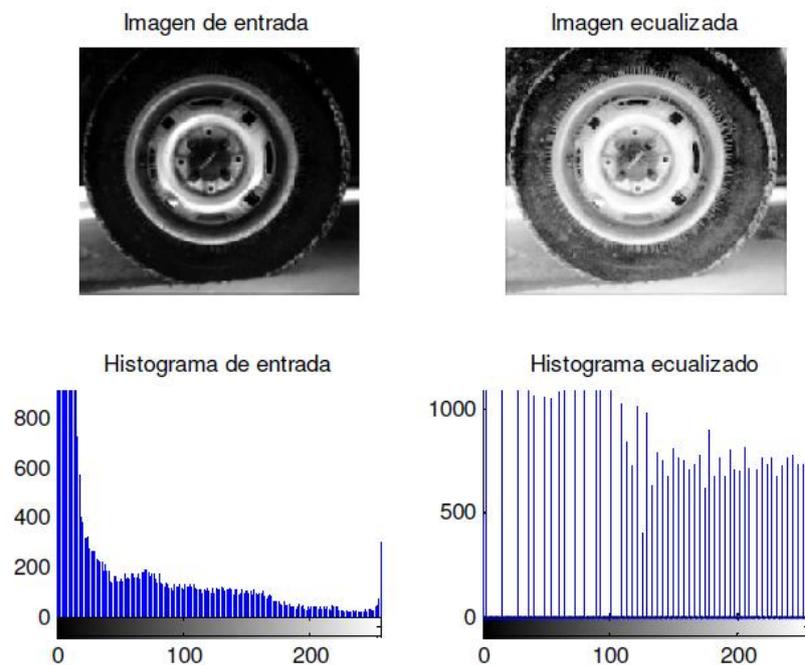


Figura 3.1: Ejemplo de ecualización del histograma.

3.1.3. Realzado basado en el dominio frecuencial

Las componentes de alta frecuencia están relacionadas con los bordes de los objetos y con los pequeños detalles de la imagen. El aumento del contraste puede estar basado en una combinación lineal entre la imagen original y el resultado de un filtro paso alto. La imagen salida será:

$$\begin{aligned} \text{Imagen Salida} &= A * (\text{Imagen}) - (\text{Paso bajo de la imagen}) \\ &= (A - 1) * (\text{Imagen}) + (\text{Imagen}) - (\text{Paso bajo de la imagen}) \\ &= (A - 1) * (\text{Imagen}) + (\text{Paso alto de la imagen}) \end{aligned}$$

A es un escalar; cuando es la unidad se tendrá el resultado del filtro paso alto, en cambio para A mayor a la unidad se estará realizando un énfasis sobre las componentes de alta frecuencia, produciendo el proceso de realzado y para A menor a la unidad aparece como el negativo de la imagen. A este algoritmo se le denomina *unsharp masking*. Desde el punto de vista de la implementación se suele utilizar la imagen menos el resultado de ésta con una máscara laplaciana.

3.1.4. Corrección gamma

Se trata de una transformación no lineal de niveles de gris que sustituye el nivel de gris i por i^γ (para $\gamma > 0$) o por $\log(i)$ (para $\gamma = 0$), donde el parámetro $\gamma \in [0,1]$.

$$T(i) = i^\gamma \quad (3.10)$$

Esta transformación realza el rango dinámico de la imagen en zonas oscuras o sombras, mientras que lo comprime en regiones brillantes y destacadas. El principio básico radica en el producto de la iluminación de entrada y la superficie de reflexión. Se pretende extraer la información a nivel de objeto independiente de la iluminación; el hecho de tomar logaritmos facilita esta tarea, transformando el producto en una simple suma.

Como puede observarse en la figura 3.2, si $\gamma < 1$, la imagen de salida será más clara y con $\gamma > 1$ se volverá más oscura. Este tipo de transformación se realiza con valores

normalizados de intensidad en el intervalo $[0,1]$. Por ejemplo, suponga que γ fuese 2 y no estuviera el rango de intensidad normalizado, por ejemplo en el intervalo de $[0\ 255]$. Se observa que el rango de salida sería mayor a 256 niveles de gris.

No obstante, cabe destacar que esta y otras transformaciones no introducen mayor información en las imágenes, pues los resultados son sólo expansiones de los rangos dinámicos. En cambio, estas operaciones facilitan las siguientes etapas de extracción de información de mayor relevancia.

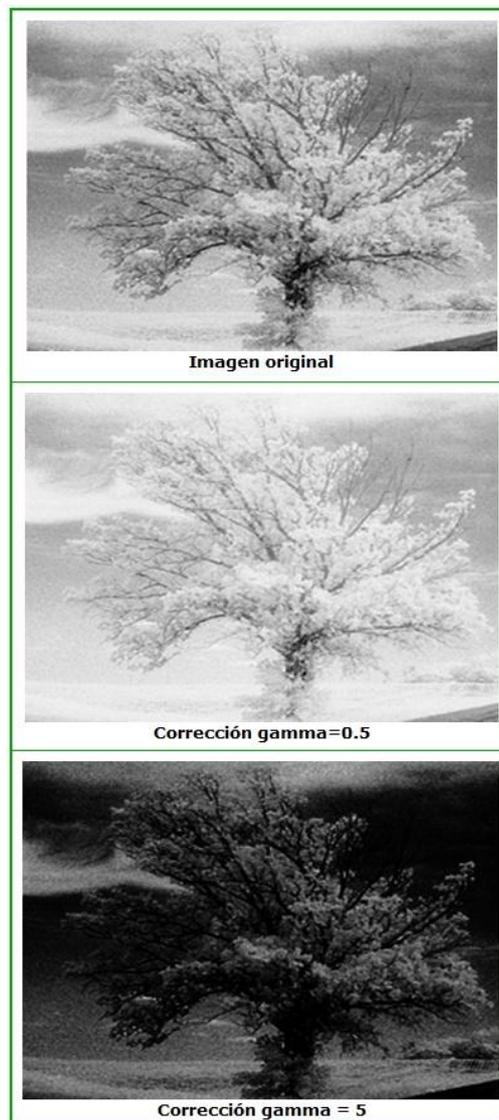


Figura 3.2: Imagen original e imágenes con diversos factores gamma.

3.2. Técnicas de suavizado

Las técnicas de suavizado de las imágenes intentan eliminar el nivel de ruido presente en la imagen. En el proceso de formación de la imagen se ha generado ruido que se ha sumado a la señal. Este ruido se explica como una variable aleatoria que sigue una función de densidad determinada. El origen del ruido es múltiple y existen tres tipos básicos: gaussiano, impulsional y multiplicativo.

El ruido más común se origina en el proceso de captación al convertir los fotones en carga eléctrica, debido al carácter no determinista de esta transformación. Este tipo de ruido se explica mediante el modelo de función de densidad normal o gaussiana. El segundo tipo de perturbación se consigue como consecuencia de la saturación de carga que recibe un píxel ya sea por exceso o por defecto. A este tipo de ruido se le denomina de ‘sal y pimienta’. La saturación se produce debido a la sensibilidad de las cámaras al infrarrojo y al encontrar sobre el escenario objetos que están calientes. Este segundo tipo de comportamiento se modela con una función de densidad de tipo impulsional. Por último, hay ruido generado por la falta de iluminación uniforme sobre la escena capturada. Este error es como si la imagen estuviera formada por la multiplicación de dos imágenes, por un lado la imagen que refleja la intensidad de iluminación sobre los objetos y de otro la función de refracción de los cuerpos vistos en la escena. A esta categoría de ruido se le relaciona con los filtros homomórficos.

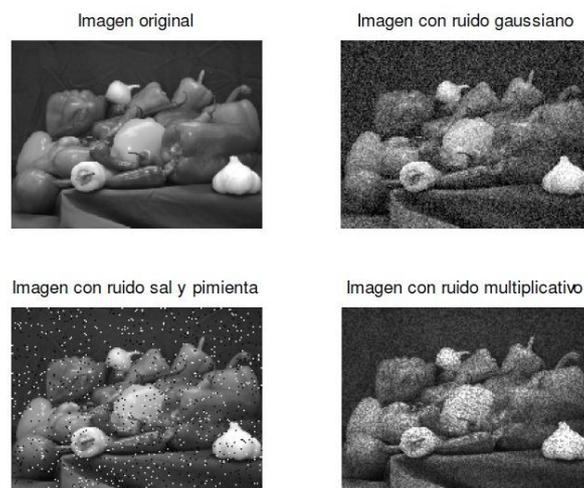


Figura 3.3: Fuentes de error a)original; b)gaussiano; c)impulsional; d)multiplicativo.

3.2.1. Filtros gaussianos

Los filtros gaussianos son máscaras de convolución que emplean la discretización de las funciones de densidad normal de media cero y varianza dada, $N(0, \sigma^2)$.

$$h_{\sigma}(x, y) = e^{-\frac{(x^2+y^2)}{2\sigma^2}} \quad (3.11)$$

Las respuestas en frecuencia de estas máscaras son filtros cuyos anchos de banda dependen de sus varianzas. No presentan rizados o lóbulos secundarios en el módulo y son de fase lineal. Al igual que los filtros binomiales, las máscaras de Gauss bidimensionales se construyen a partir de las unidimensionales. Su ventaja, respecto a los filtros binomiales está en su parametrización dependiendo de la varianza. A medida que la varianza sea más elevada significará que tiene en cuenta a vecinos más alejados; por el contrario, si la varianza es más pequeña indicará que sólo ponderará con los más próximos. La varianza funciona como grado de ponderación en la vecindad. Ésta es la interpretación en el dominio espacial. Desde el punto de vista frecuencial, la varianza determina el ancho de banda del filtro de paso bajo que supone la máscara de Gauss. El lóbulo principal del módulo de la respuesta frecuencia viene determinado por la varianza. A mayor varianza menor ancho de banda.

En el proceso de discretización de la máscara Gaussiana aparece el problema del número de coeficientes finitos a emular a la función continua. Para evitar la formación de lóbulos secundarios en la respuesta frecuencial de la máscara, se toma como regla que el tamaño del operador, w , debe ser al menos de:

$$w \geq 3c \quad (3.12)$$

Siendo c el tamaño del lóbulo central y dependiente de la varianza:

$$c = 2\sqrt{2\sigma^2} \quad (3.13)$$

Esta regla indica que el 98% de los posibles casos de la variación de la variable aleatoria normal están comprendidos. Si la varianza es unitaria, la máscara debería de

ser al menos 9×9 . Nótese que se busca una máscara bidimensional que tenga simetría par. También hay que considerar que esta máscara tiene simetría radial, por lo que sólo es necesario calcular un cuadrante de la máscara.

En la figura 3.4 se visualizan distintas imágenes correspondientes a cuatro valores distintos de la varianza. Como puede observarse, cuanto mayor es la varianza, mayor es el suavizado de la imagen.



Figura 3.4: (De izda. a dcha. y de arriba abajo) imagen con filtrado gaussiano con valores de $\sigma=1,2,3,4$.

3.3. Detección de caras: Viola-Jones

El Método de Viola-Jones es un método de detección de objetos en una imagen, desarrollado por los investigadores Paul Viola y Michael Jones en el año 2001. Éste es el primero de todos los métodos capaz de detectar eficazmente y en tiempo real objetos en una imagen. Actualmente consiste uno de los métodos más conocidos y más utilizados para la detección de caras y personas [10].

El método de Viola-Jones es un método de aproximación basado en la apariencia; éste, como método de aprendizaje supervisado, está dividido en dos etapas: una primera etapa de aprendizaje del clasificador basada en un gran número de ejemplos positivos (es decir, los objetos de interés, como por ejemplo las caras) y de ejemplos negativos, y una fase de detección mediante la aplicación de este clasificador a las imágenes no conocidas.

3.3.1. Características

En lugar de trabajar directamente con los valores de los píxeles, y para una mayor eficiencia y rapidez del método, Viola y Jones proponen utilizar un determinado tipo de características, las *características tipo Haar*, que se calculan como la diferencia de la suma de los píxeles de dos o más zonas rectangulares adyacentes. La figura 3.5 muestra ejemplos de estas características propuestas por Viola y Jones de 2, 3 ó 4 rectángulos, en los que los píxeles de la zona oscura (la negativa) se restan a la suma de los píxeles blancos [11].

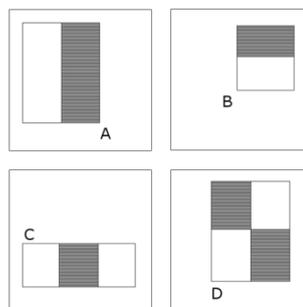


Figura 3.5: Ejemplos de las características tipo Haar empleadas por Viola y Jones.

Para el cálculo de estas características de forma rápida y eficiente, los autores proponen un nuevo método, la *imagen integral*: una representación en forma de imagen del mismo tamaño que la imagen original, donde cada uno de los puntos se calcula como la suma de los píxeles situados por encima de él y a su izquierda.

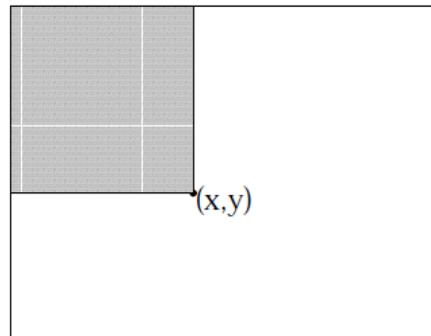


Figura 3.6: El valor de la imagen integral en un punto (x,y) es la suma de los píxeles de la región superior izquierda.

Como se muestra en el ejemplo de la figura 3.7, mediante el uso de la imagen integral puede calcularse cualquier suma rectangular únicamente con cuatro referencias. El valor de la imagen integral en el punto 1 es la suma de los píxeles del rectángulo A; en el punto 2 es $A + B$; en el 3 es $A + C$; y en el punto 4 es $A + B + C + D$. La suma en el rectángulo D se calcula como: $D = 4 + 1 - 2 - 3$.

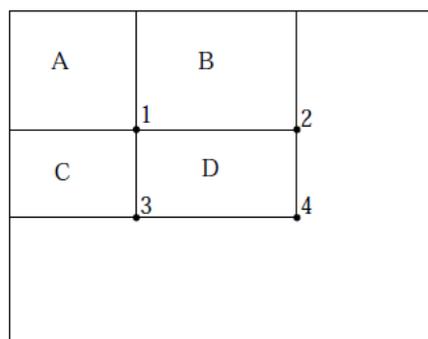


Figura 3.7: Suma del área rectangular D.

3.3.2. Algoritmo de aprendizaje

Las características se calculan para todas las posiciones y a todas las escalas con una pequeña ventana de detección, normalmente de 24x24 píxeles o de 20x15 píxeles, generando para el caso de 24x24 píxeles hasta 45396 funciones.

El sistema emplea una variante del algoritmo de aprendizaje *AdaBoost*, tanto para la selección de las mejores características como para el entrenamiento de los clasificadores que las utilizan.

3.3.3. Cascada de clasificadores

La evaluación de los clasificadores fuertes generados por el proceso de aprendizaje puede ser rápida, pero no lo suficiente como para ser llevada a cabo en tiempo real. Por este motivo Viola y Jones propusieron un esquema basado en una cascada de clasificadores fuertes [12]. Tal y como se visualiza en la figura 3.8, cada etapa corresponde a un clasificador fuerte y está entrenada con todos los ejemplos que la etapa anterior no ha clasificado correctamente más algunos nuevos. Por tanto, en la etapa de entrenamiento, cada etapa se entrena con un conjunto óptimo de características capaces de detectar cada vez ejemplos más complicados; es decir, las primeras etapas se encargan de descartar sub-imágenes que son muy diferentes de una cara, mientras que las últimas etapas pueden rechazar ejemplos mucho más complejos como pueden ser globos, pelotas, dibujos, etc...

Si se utilizase un clasificador que tuviera un único estado, normalmente habría que aceptar los falsos negativos para reducir la tasa de falsos positivos. Sin embargo, para las primeras etapas del clasificador en cascada se acepta una alta tasa de falsos positivos esperando que las etapas posteriores puedan encargarse de reducir esta tasa mediante clasificadores más especializados. Con esto se pretende también reducir la tasa de falsos negativos en el clasificador final, ya que una sub-ventana será clasificada como cara sólo en el caso de que haya pasado por todas las etapas del clasificador.

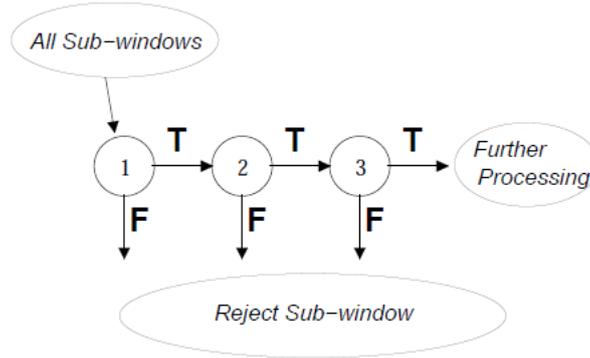


Figura 3.8: Arquitectura en cascada.

Dada una cascada de clasificadores, la tasa de falsos positivos se calcula:

$$F = \prod_{i=1}^K f_i \quad (3.14)$$

donde F es la tasa de falsos positivos del detector, K es el número de clasificadores y f_i es la tasa de falsos positivos calculada para el clasificador i . Por otro lado la tasa de detección de la cascada se calcula:

$$D = \prod_{i=1}^K d_i \quad (3.15)$$

Donde D es la tasa de detección del detector, K es el número de clasificadores y d_i es la tasa de detección calculada para el clasificador i . Una vez establecido esto, puede observarse como por ejemplo un detector consistente en 10 etapas con una tasa de detección del 99% por etapa y una tasa de falsos positivos del 30% por clasificador supondría una tasa de detección global del $0.99^{10} \approx 0.9$ y una tasa de falsos positivos global del $0.3^{10} \approx 6 \cdot 10^{-6}$.

3.4. Máscara elipsoidal

La máscara elipsoidal es un preprocesado el cual a partir de un tamaño dado de imagen, recorta una elipse alrededor de la cara, devolviendo los píxeles internos a la elipse como un vector y despreciando los hallados fuera de ésta.

Este tipo de máscara es muy útil, ya que únicamente procesa la información que se encuentra en el interior de la elipse (la relativa al rostro), y por lo tanto elimina la información que a priori puede resultar innecesaria.

En la figura 3.9 se muestra una imagen original y su correspondiente tras la aplicación de la máscara elipsoidal.



Figura 3.9: Imagen original sin y con máscara elipsoidal.

Capítulo 4

Herramientas

4.1. Entorno de desarrollo

El entorno utilizado para desarrollar el algoritmo es el software MATLAB, en particular se emplea la versión *r2009b*. El nombre de MATLAB proviene de la contracción de los términos MATrix LABoratory [13]. Es un entorno de computación y desarrollo de aplicaciones que integra análisis numérico, cálculo matricial, proceso de señal y visualización gráfica en un entorno completo. En la actualidad goza de un alto nivel de implantación en centros de educación, así como en departamentos de investigación y desarrollo de muchas compañías industriales nacionales e internacionales.

MATLAB fue originalmente desarrollado en lenguaje FORTRAN y al pasar de los años fue complementado y reimplementado en lenguaje C. Actualmente la licencia de MATLAB es propiedad de MathWorks Inc. Está disponible para un amplio número de plataformas y opera bajo sistemas operativos como UNIX, Macintosh y Windows. Actualmente, MATLAB dispone de su propio lenguaje de programación el cual, es una magnífica herramienta de alto nivel para desarrollar aplicaciones técnicas, fácil de utilizar y que aumenta significativamente la productividad de los programadores respecto a otros entornos de desarrollo.

Adicionalmente, cuenta con un código básico y un conjunto de librerías especializadas (*toolboxes*). Estas librerías cubren casi todas las áreas principales en el mundo de la ingeniería y la simulación, destacando entre ellas: procesamiento de imágenes, procesamiento de señales, control robusto, estadística, análisis financiero, matemática simbólica, redes neuronales, lógica difusa, identificación de sistemas, simulación de sistemas dinámicos, etc.

El Toolbox que se ha utilizado para desarrollar el algoritmo del presente proyecto es el denominado *Image Processing Toolbox*. Este paquete de herramientas proporciona a MATLAB un conjunto de funciones que amplían las capacidades del producto para desarrollar aplicaciones y nuevos algoritmos en el campo del procesamiento y análisis de imágenes. Algunas de las funciones más importantes son:

- Análisis de imágenes y estadística.

- Diseño de filtros y recuperación de imágenes.
- Mejora de imágenes.
- Operaciones morfológicas.
- Definición de mapas de color y modificación gráfica.
- Operaciones geométricas.
- Transformación de imágenes.

MATLAB almacena la mayoría de las imágenes en niveles de gris como arreglos bidimensionales (matrices) en los cuales cada elemento de la matriz corresponde a la intensidad de un píxel de la imagen. Algunas imágenes, como las imágenes a color (RGB), requieren de un arreglo tridimensional (hipermatrices), donde en el primer plano en el espacio tridimensional representa la intensidad de rojo de los píxeles, el segundo plano representa la intensidad de verde de los píxeles y el tercer plano representa la intensidad de azul de los píxeles. Para reducir el espacio en memoria requerido para almacenar imágenes, MATLAB almacena los datos en arreglos de 8 o 16 bits sin signo, que corresponden a las clases *uint8* y *uint16*, respectivamente.

4.2. Bases de datos

En este apartado se muestran y comentan las propiedades de las bases de datos utilizadas durante la realización de los experimentos del proyecto. Cabe destacar, por una parte, que cada una de estas bases de datos está formada por un número distinto de sujetos e imágenes de los mismos, y por otra parte, que cada una de estas bases de datos está tomada bajo condiciones diferentes.

4.2.1. Base de datos Olivetti (ORL Database)

Se trata de una base de datos formada por los rostros de 40 individuos, a razón de 10 imágenes por individuo tomadas entre abril de 1992 y abril de 1994 en el *Olivetti Research Laboratory* en Cambridge [14].

Todas las imágenes de la base de datos ORL fueron tomadas en diversos momentos bajo unas condiciones de iluminación y de fondo de imagen controladas, existiendo variaciones en las expresiones faciales: ojos abiertos o cerrados, personas sonriendo o serias, con o sin lentes, etc. Preferentemente los sujetos se hallan en posición frontal, permitiendo una tolerancia de rotación de aproximadamente 20 grados y con una variación en la escala del 10%. Las imágenes están disponibles en escala de grises con una resolución de 92x112 píxeles en formato PGM.

La base de datos ORL puede obtenerse gratuitamente a través de Internet en la página Web del Olivetti Research Laboratory: <http://www.cam-orl.co.uk/facedatabase.html>.

Todas las imágenes de los rostros disponibles en la base de datos ORL pueden apreciarse en detalle en la figura 4.1.

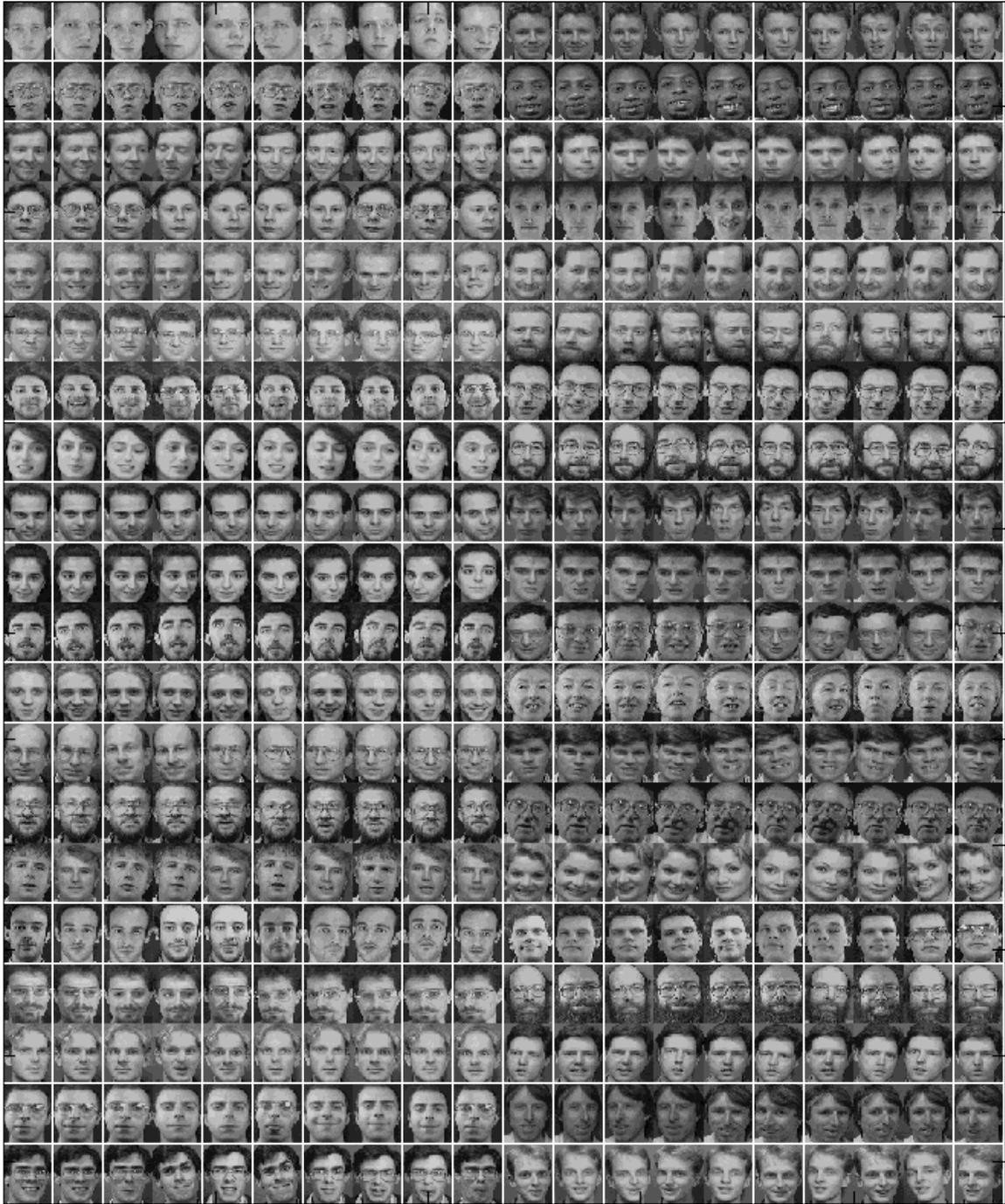


Figura 4.1: Imágenes de la ORL Database.

4.2.2. Base de datos UMH

Se trata de una base de datos formada por 17 individuos, a razón de 30 imágenes por individuo tomadas en el curso 2005/06 en la Universidad Miguel Hernández de Elche.

Las imágenes de esta base de datos fueron tomadas bajo diferentes condiciones de iluminación y en 4 tipos de escenarios, existiendo variaciones tanto en las expresiones faciales (sonriendo, serios, vista hacia arriba o abajo, etc.) como en la posición de los rostros frente al dispositivo de adquisición:

- 10 fotos sobre fondo blanco
- 10 fotos sobre fondo complejo
- 5 fotos sobre fondo blanco con oclusiones (utilizando gafas, sombreros, etc.)
- 5 fotos sobre fondo complejo con oclusiones

Las imágenes están disponibles en color con una resolución de 121x151 píxeles en formato JPEG.

La base de datos UMH fue proporcionada por la profesora M^a Asunción Vicente Ripoll durante el transcurso de redacción del presente proyecto.

Algunas de las imágenes de los rostros disponibles en la base de datos ORL pueden apreciarse en detalle en las figuras 4.2 y 4.3.

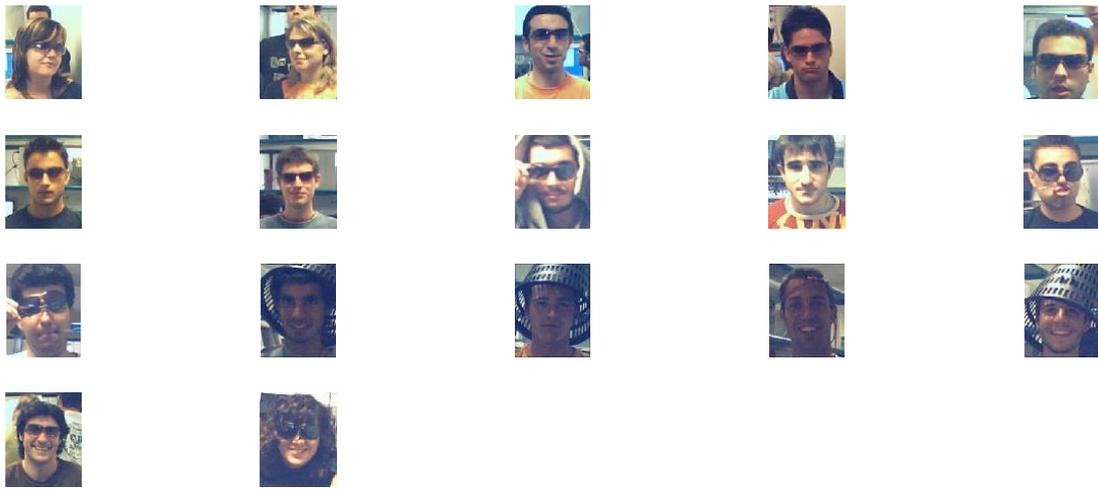


Figura 4.2: Imágenes de los 17 individuos que forman la librería de caras UMH.

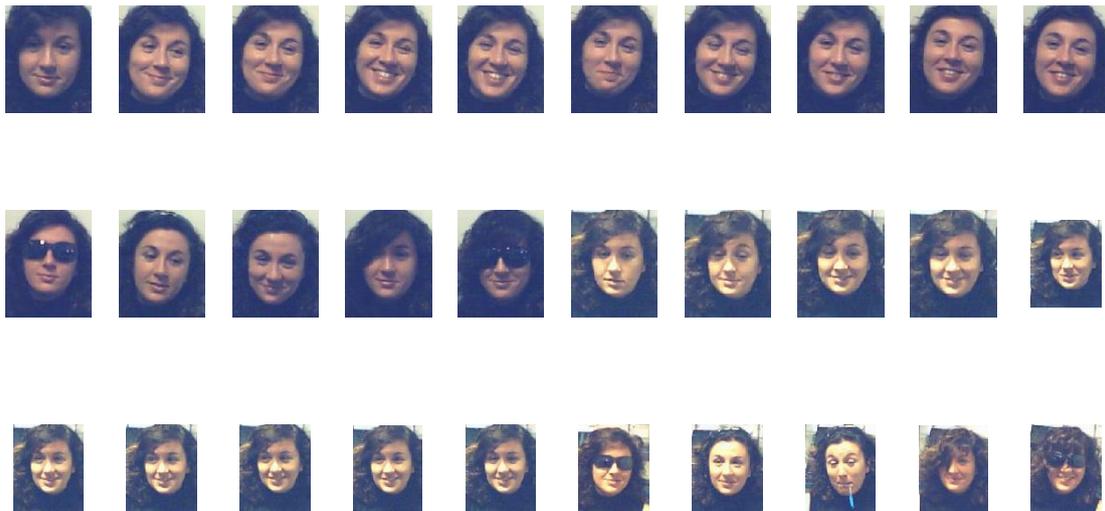


Figura 4.3: Imágenes de un individuo de la librería de caras UMH.

4.2.3. Base de datos Yale B Extended

La base de datos Yale B contiene imágenes de 10 sujetos tomadas bajo 64 condiciones de iluminación diferentes y 9 poses cada una. Recientemente ha sido actualizada con el nombre de “Extended Yale Face Database”; esta actualización se compone de 38 individuos (29 varones y 9 mujeres), a razón de 65 imágenes por individuo. En ambos casos las imágenes se dividen en 5 subconjuntos, en función del ángulo entre el eje óptico de la cámara y la dirección de la fuente de iluminación (12°, 25°, 50°, 77° y 90°).

En los experimentos realizados durante el transcurso del proyecto se han tomado únicamente las 10 primeras imágenes de la “Extended Yale Face Database” (en total 380 imágenes) [15]. Por otro lado cabe mencionar que se trata de la versión “Cropped” de la librería de caras, en la que las imágenes de los sujetos están recortadas y centradas en los rostros, por lo que posteriormente, en la etapa de experimentos, no deberá realizarse el preprocesado de detección de caras mediante Viola-Jones.

Las imágenes fueron capturadas por una cámara Sony XC-75 y están disponibles en escala de grises con una resolución de 192x168 píxeles⁷ en formato PGM.

En la figura 4.4 se muestran imágenes de los 10 sujetos correspondientes a la base de datos original.



Figura 4.4: Imágenes de los 10 sujetos de la “Yale B Database” original.

⁷ Originalmente las imágenes de la base de datos estaban disponibles con una resolución de 640x480 píxeles

Capítulo 5

Experimentos

5.1. Programa principal

El programa principal correspondiente a cada una de las bases de datos empleadas se encarga de leer las imágenes de la librería seleccionada, y para un número determinado de imágenes de entrenamiento realiza, basándose en el Análisis de Componentes Principales, la clasificación de las imágenes de test seleccionadas. Concretamente el programa realiza la transformación de *whitening*, habiendo aplicado previamente el truco de Turk & Pentland para facilitar y acelerar el análisis de PCA. El clasificador empleado en el desarrollo del código es el vecino más cercano: k-NN con $k=1$ y la **norma L2** como medida de la distancia.

Los parámetros a introducir por el usuario son:

- *objects*: define el número de clases, es decir, el número de individuos de la base de datos ORL.
- *np*: es el número de ejemplos por clase, es decir, el número de imágenes seleccionadas de cada individuo.
- *n_train*: determina el número de imágenes de entrenamiento en orden secuencial en la lectura de la base de datos ORL. El resto de imágenes de *np* formará el conjunto de imágenes de test.
- *n_vectors*: es el número de vectores utilizado para reducir el conjunto de datos de entrada.
- *see*: es un parámetro que determina la visualización de las gráficas: NO ==> 0; SÍ ==> 1

Por otro lado, en adición, el programa principal aplica diferentes técnicas de preprocesado a las imágenes, dando lugar a un total de 12 combinaciones diferentes de preprocesados. El objetivo de estos preprocesados es mejorar la tarea clasificación de las imágenes de test, es decir, mejorar el reconocimiento e identificación de los individuos de cada una de las bases de datos.

Para la definición de cada uno de los preprocesados, se ha empleado una notación basada en 3 bits, cuyo significado se muestra en la tabla 5.1:

n° bit	Valor asignado	Definición
2	0	-
	1	Detector Viola-Jones
1	0	-
	1	Ecuilización del histograma
	2	“Ecuilización Tan-Triggs”
0	0	-
	1	Máscara elipsoidal

Tabla 5.1: Notación empleada para la designación de los preprocesados.

A continuación, en la tabla 5.2 se muestran todos los preprocesados empleados en los experimentos del presente proyecto:

Notación	Descripción del preprocesado
000	Imágenes originales
001	Máscara elipsoidal
010	Ecuilización del histograma
011	Ecuilización del histograma + Máscara elipsoidal
020	“Ecuilización Tan-Triggs”
021	“Ecuilización Tan-Triggs” + Máscara elipsoidal
100	Detector Viola-Jones
101	Detector Viola-Jones + Máscara elipsoidal
110	Detector Viola-Jones + Ecuilización del histograma
111	Detector Viola-Jones + Ecuilización del histograma + Máscara elipsoidal
120	Detector Viola-Jones + “Ecuilización Tan-Triggs”
121	Detector Viola-Jones + “Ecuilización Tan-Triggs” + Máscara elipsoidal

Tabla 5.2: Técnicas de preprocesado empleadas.

En la tabla 5.2 se muestran todos los preprocesados descritos en el capítulo 3, con la particularidad de la aparición de un nuevo tipo de preprocesado anteriormente no descrito denominado *Ecualización Tan-Triggs*. Éste recibe el nombre de los investigadores Xiaoyang Tan and Bill Triggs, autores del artículo de investigación *Preprocessing and Feature Sets for Robust Face Recognition* [16], del que se ha extraído la idea.

La *Ecualización Tan-Triggs* está basada en una serie de pasos destinados a estudiar el efecto de variaciones en la iluminación, ensombrecidos locales y brillos. La cadena de preprocesado se muestra en la figura 5.1 y su efecto en la imagen de un rostro en la figura 5.2.

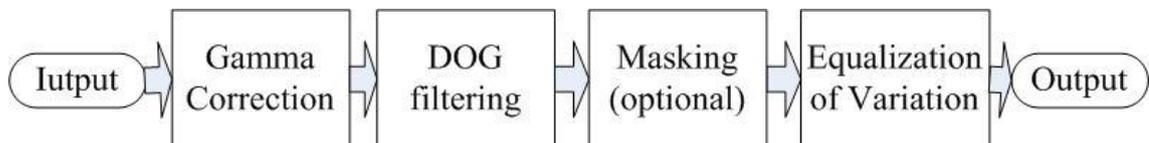


Figura 5.1: Diagrama de bloques con los preprocesados a realizar.

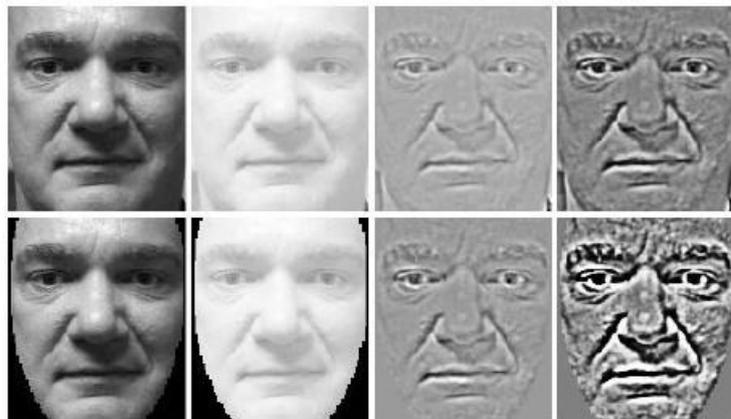


Figura 5.2: Ejemplo de los distintos estadios de la cadena de preprocesado. Fila superior: una imagen de entrada es preprocesada mediante la corrección Gamma, el filtrado DoG y la ecualización del contraste. Fila inferior: procedimiento idéntico a la primera fila aplicando una máscara elipsoidal tras cada preprocesado.

Los parámetros de cada uno de los diferentes preprocesados son los siguientes:

- Corrección gamma: $\gamma = 0.2$
- Filtro gaussiano (DoG): $\sigma_0 = 0,25$; $\sigma_1 = 2$
- Máscara (opcional): elipsoidal
- Ecuación: factor de potencia, $\alpha = 0.1$; umbral, $\tau = 10$

La ecualización que realiza el preprocesado descrito consiste en una aproximación basada en un proceso de dos estadios (ecuaciones 5.1 y 5.2); ésta normaliza los niveles de intensidad de la imagen mediante un estimador que distribuye la señal, eliminando los valores extremos de la misma (correspondientes a brillos, ruidos en los bordes, variaciones de iluminación, sombras, etc.).

$$I(x, y) \leftarrow \frac{I(x, y)}{(\text{Media}(|I(x', y')|^\alpha))^{1/\alpha}} \quad (5.1)$$

$$I(x, y) \leftarrow \frac{I(x, y)}{(\text{Media}(\min(\tau, |I(x', y')|^\alpha))^{1/\alpha}} \quad (5.2)$$

α es un exponente que reduce la influencia de los valores extremos, y τ es un umbral que discrimina estos valores tras la primera fase de normalización.

Llegados a este punto, la imagen resultante aún contiene algunos valores extremos; para reducir su influencia en las posteriores etapas, por último se aplica una función no lineal: $I(x, y) \leftarrow \tau \tanh(I(x, y)/\tau)$ que limita la imagen I al rango $(-\tau, \tau)$.

Para finalizar, y antes de dar paso a los apartados concernientes a los experimentos, cabe mencionar que para la presentación y análisis del resultado de las clasificaciones se ha recurrido a la construcción de la matriz de confusión. En el campo de la inteligencia artificial una matriz de confusión es una herramienta de visualización que se emplea en aprendizaje supervisado. Se trata de una matriz de orden $|X| \times |Y|$ (donde J es el número de clases) donde cada columna representa el número de predicciones de cada clase, mientras que cada fila representa a las instancias en la clase real. En la diagonal de esta

matriz se halla el número de asignaciones correctas, mientras que en los elementos fuera de ésta se recogen las migraciones o fugas (clasificaciones erróneas).

A partir de los valores de la diagonal pueden deducirse algunos valores indicativos de la calidad de la clasificación, como es el caso de la bondad media de la clasificación (también referenciada en el documento como porcentaje de aciertos):

$$Bondad\ media = \frac{1}{J} \sum_{j=1}^J \acute{E}xito(j) \quad (5.3)$$

Para una mejor visualización, todos los gráficos del documento pueden consultarse con más detalle en el CD adjunto.

5.2. Experimentos con la base de datos ORL

5.2.1. Experimentos iniciales

En el primer experimento se ha ejecutado el programa *selección_prep_orl.m* para visualizar el efecto que tendría en el porcentaje de aciertos (imágenes clasificadas correctamente) la aplicación del preprocesado de ecualización del histograma a las imágenes originales.

Los parámetros designados para esta prueba son: *objects=40*; *np=10*; *n_train=3*; *n_vectors=100*; *see=1*. El resultado obtenido se muestra en la figura 5.3.

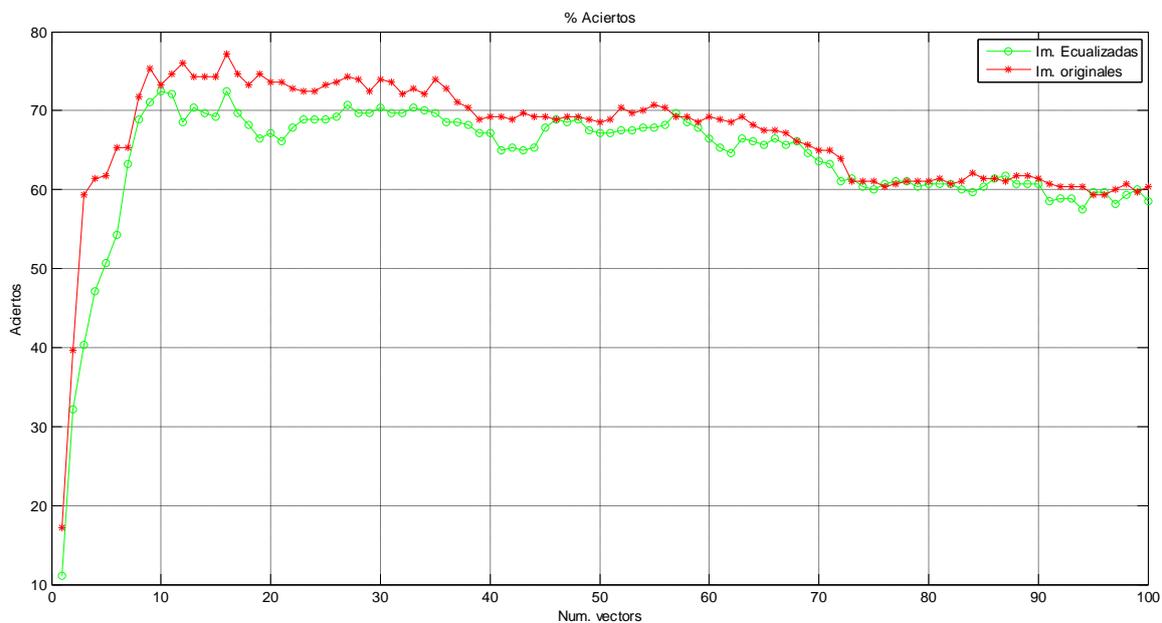


Figura 5.3: Aciertos de imágenes originales e imágenes preprocesadas mediante ecualización del histograma; en rojo se muestran los aciertos correspondientes a las imágenes originales y en verde los de las imágenes tras la ecualización del histograma.

A priori podría pensarse que la ecualización del histograma fuera a proporcionar un porcentaje de aciertos mayor que las imágenes sin ningún tipo de preprocesado, pero como se observa esto no ocurre así. Puesto que las imágenes de la base de datos ORL

son imágenes en niveles de gris con un buen nivel de iluminación y fondo controlados, la ecualización del histograma no mejora, sino que empeora los resultados.

En la figura 5.4 se muestran los porcentajes de aciertos para las imágenes originales y las preprocesadas mediante ecualización del histograma, variando el conjunto de imágenes de entrenamiento: desde una única imagen de entrenamiento y 9 de test, hasta 5 imágenes de entrenamiento y 5 de test. Cabe destacar que para los casos de $n_{train}=[3,5]$ se ha tomado el valor de $n_{vectors}=100$, mientras que para los casos de $n=[1,2]$, $n_{vectors}=40$.

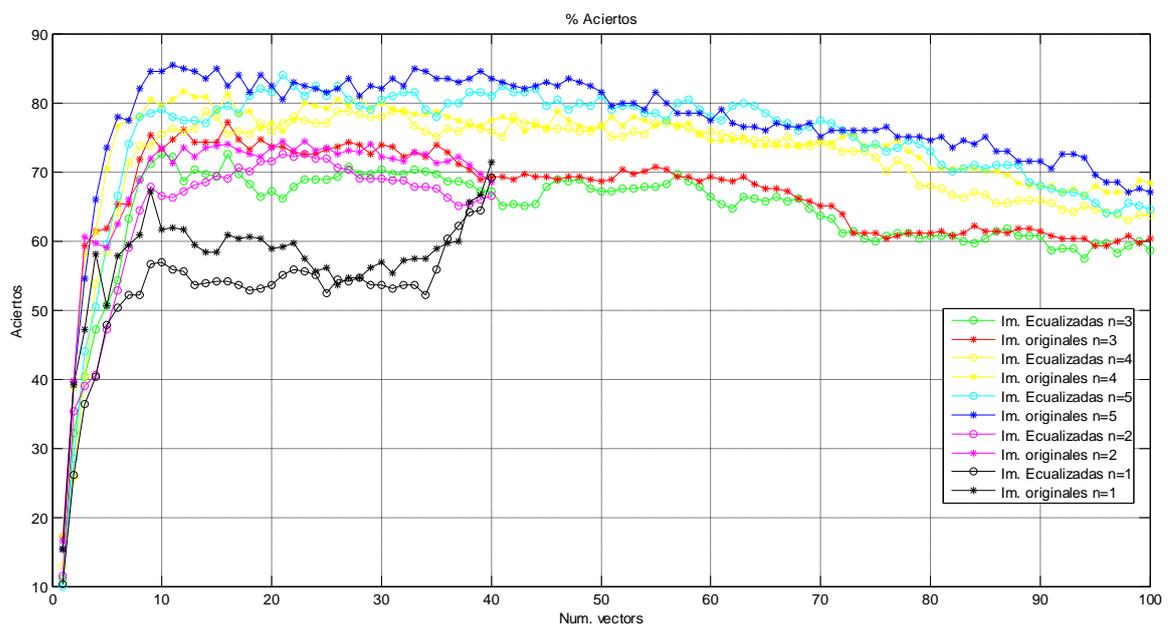


Figura 5.4: Aciertos de imágenes originales e imágenes preprocesadas mediante ecualización del histograma para valores de $n_{train}=[1,5]$.

Tal y como era de esperar, según aumenta el número de imágenes de entrenamiento, el porcentaje de aciertos aumenta, ya que las clases están definidas por un mayor número de elementos (mayor cantidad de información).

Para estudiar la calidad de la clasificación, se han comprobado las clasificaciones correspondientes al individuo número 1 para las imágenes de test desde la imagen número 4 hasta la 10. Los resultados se muestran a continuación en las figuras 5.5 hasta la 5.11.

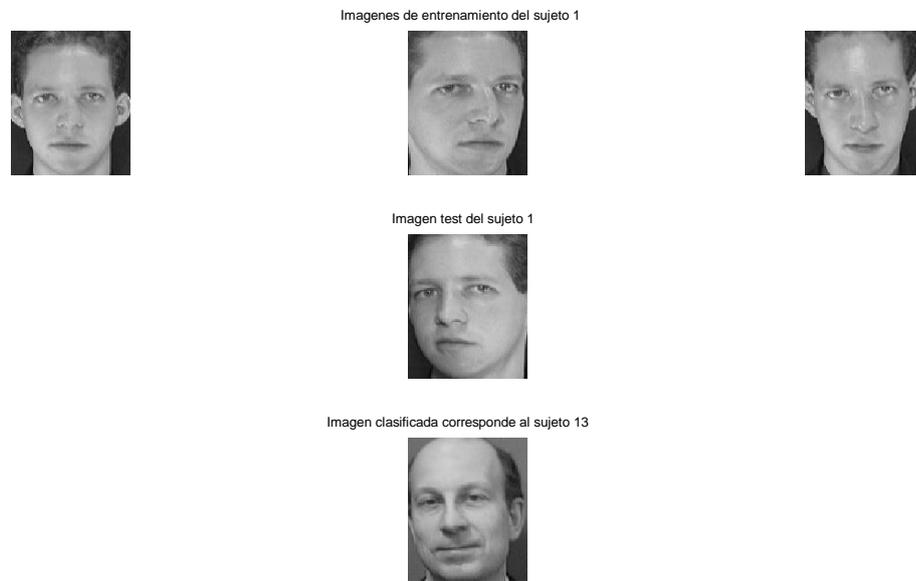


Figura 5.5: Clasificación correspondiente a la imagen test nº4.

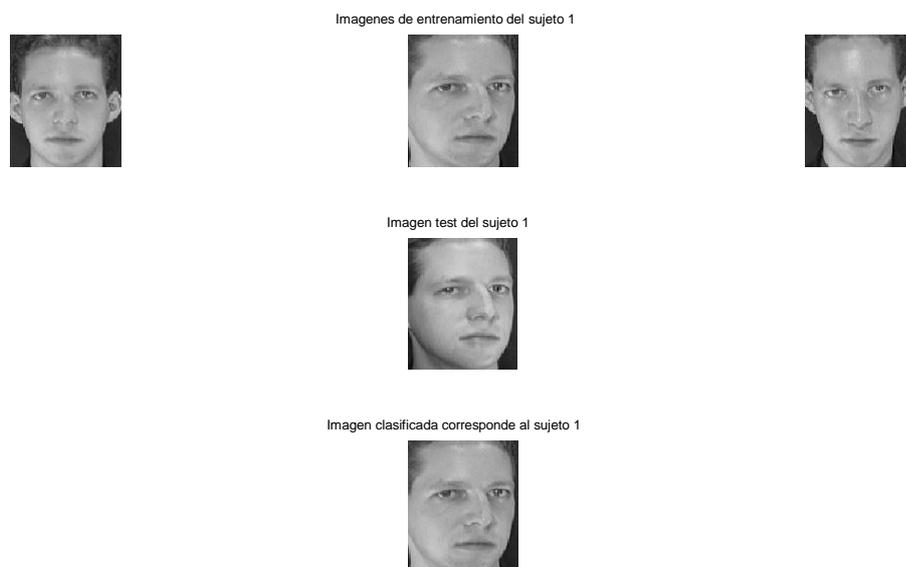


Figura 5.6: Clasificación correspondiente a la imagen test nº5.



Figura 5.7: Clasificación correspondiente a la imagen test n°6.

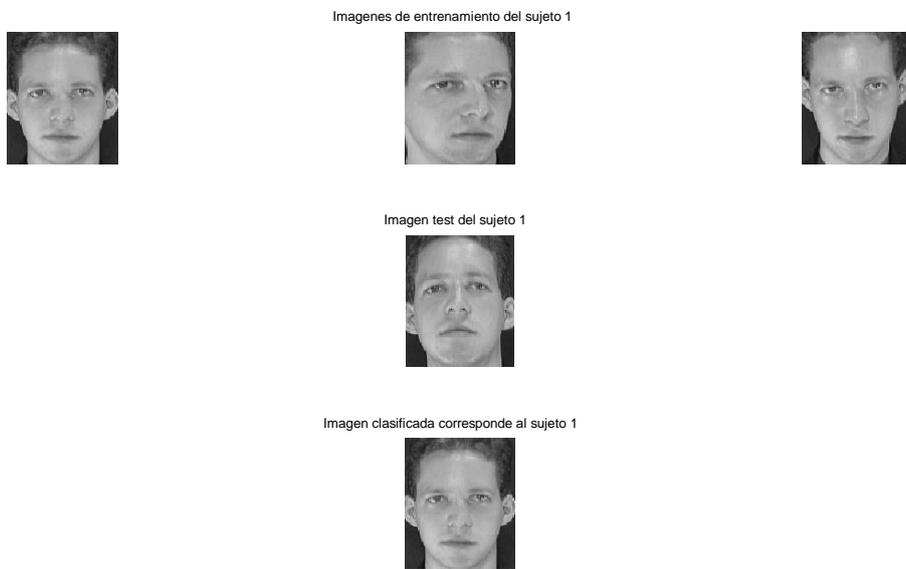


Figura 5.8: Clasificación correspondiente a la imagen test n°7.

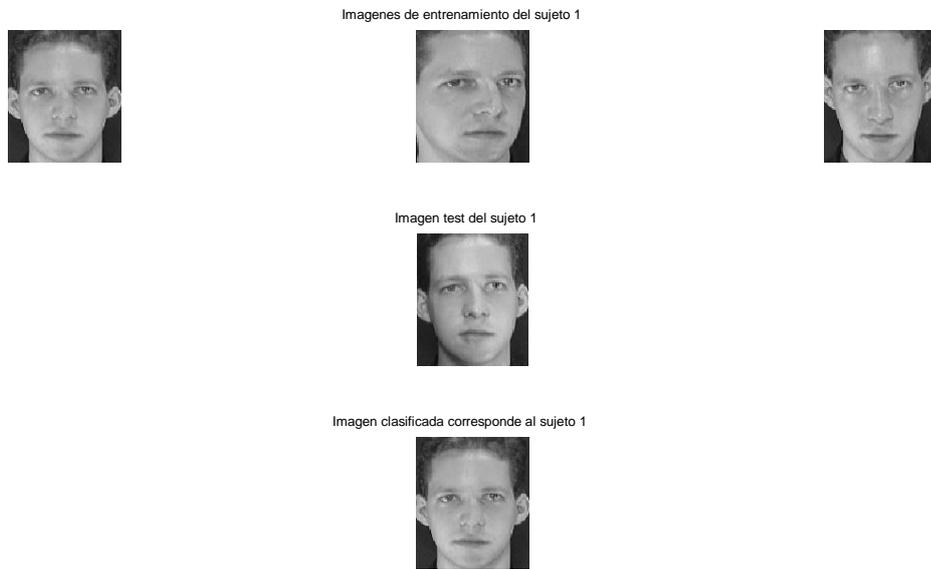


Figura 5.9: Clasificación correspondiente a la imagen test n°8.

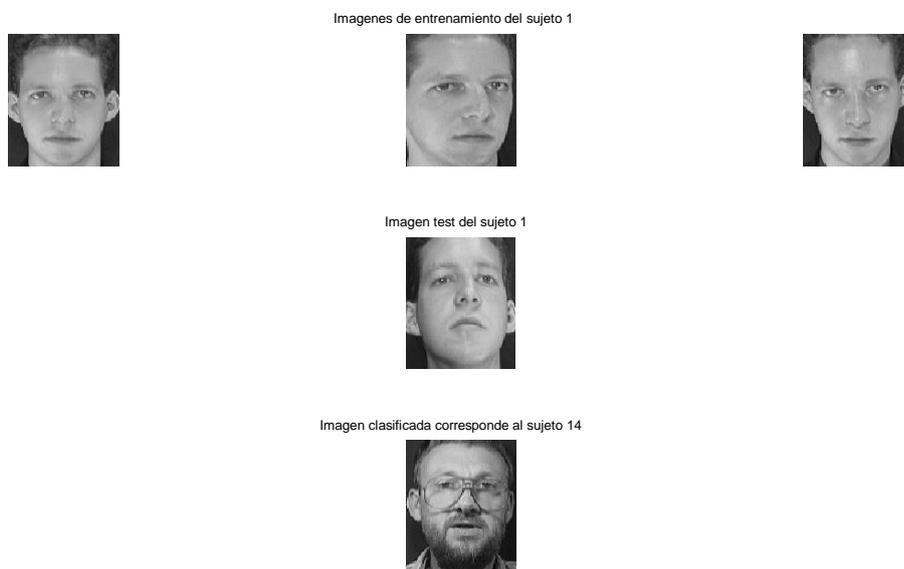


Figura 5.10: Clasificación correspondiente a la imagen test n°9.



Figura 5.11: Clasificación correspondiente a la imagen test n°10.

En la primera fila de la tabla 5.3 puede visualizarse la clasificación correspondiente al sujeto 1:

- 3 aciertos: se corresponden con las imágenes de test n° 5, n° 7 y n° 8.
- 1 imagen clasificada como sujeto 13: imagen test n°4.
- 1 imagen clasificada como sujeto 14: imagen test n°9.
- 1 imagen clasificada como sujeto 27: imagen test n°10.
- 1 imagen clasificada como sujeto 38: imagen test n°6.

Para este caso de las imágenes originales se obtuvo un valor de la bondad media de 60.3571, tal y como se observa en la figura 5.3.

En las figuras 5.12 y 5.13 se muestran los resultados derivados de la aplicación de las diferentes técnicas de preprocesado. De forma más clara se visualiza en la figura 5.14 la comparación entre los resultados obtenidos para las imágenes originales y los resultados obtenidos para los preprocesados más eficaces.

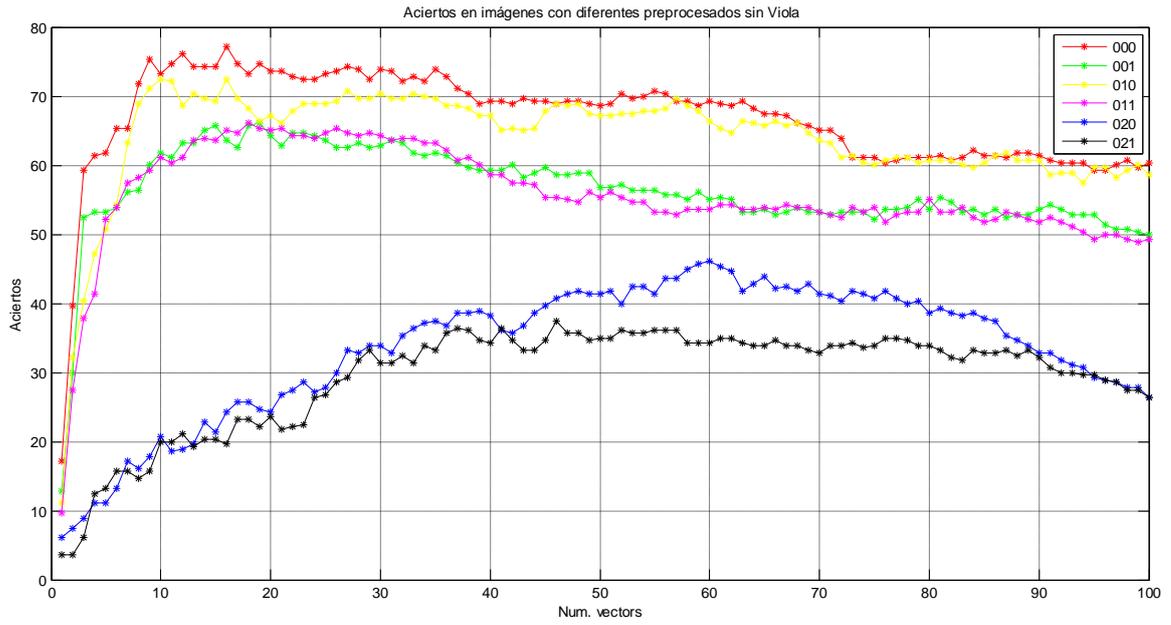


Figura 5.12: Porcentaje de aciertos en imágenes con diferentes preprocesados sin la detección Viola-Jones.

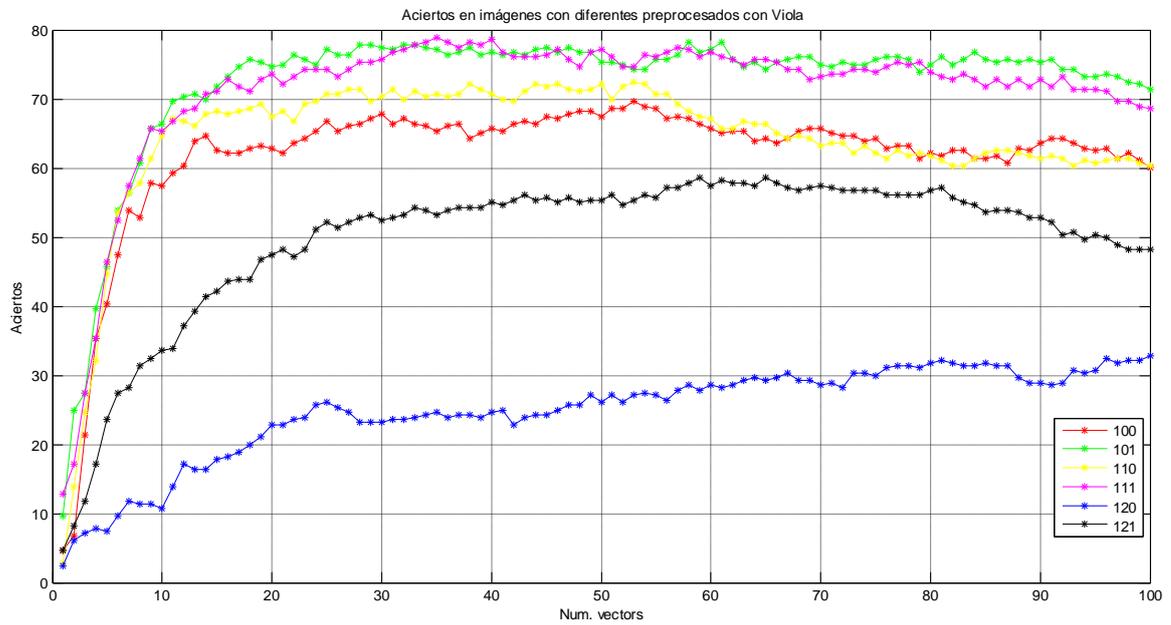


Figura 5.13: Porcentaje de aciertos en imágenes con diferentes preprocesados con la detección Viola-Jones.

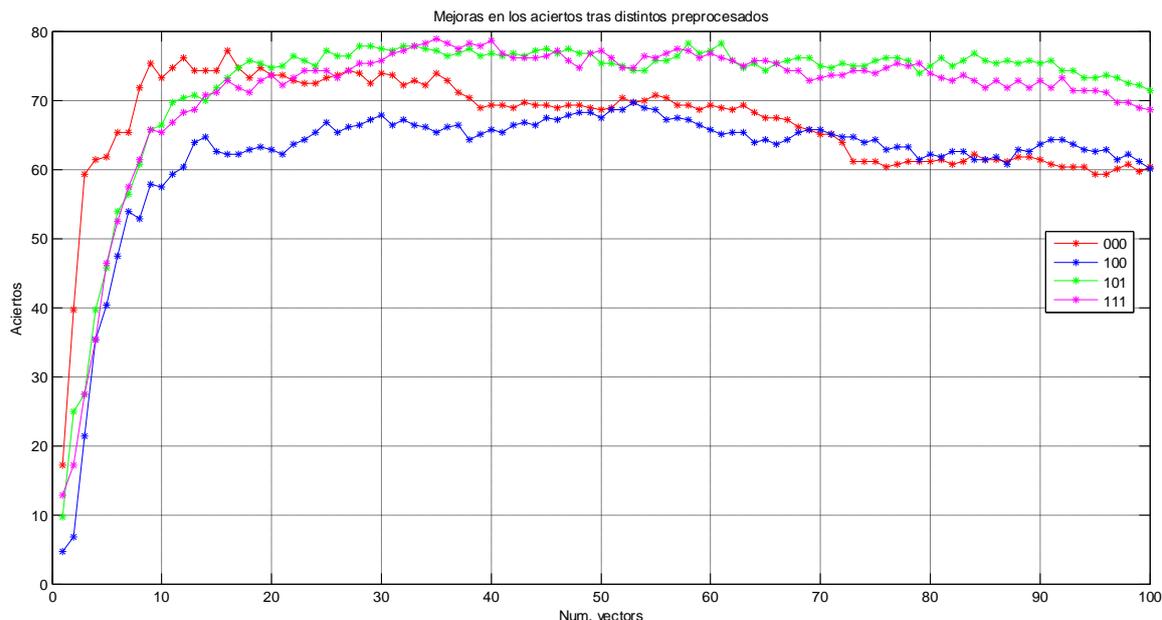


Figura 5.14: Comparativa de aciertos para los casos de clasificación más eficaces.

Observando las anteriores figuras puede concluirse que la aplicación de los preprocesados de los casos 101 y 111 supone una pequeña mejora en la clasificación de los sujetos. Sin embargo, en aquellos casos en los que la Ecuación Tan-Triggs está presente, el porcentaje de aciertos disminuye notablemente.

Como anteriormente se ha mencionado, la base de datos ORL contiene imágenes con un buen nivel de iluminación y fondo controlados, por lo que la ecualización, tanto la normalizada como la Tan-Triggs, no hace sino empeorar los resultados obtenidos para las imágenes originales. En cambio, la combinación del detector de Viola-Jones y la máscara elipsoidal sí induce mejoras en los resultados.

En las tablas 5.4, 5.5 y 5.6 se muestran las matrices de confusión para los casos 100, 101 y 111, donde pueden comprobarse los resultados obtenidos en las gráficas anteriores. La bondad media para cada uno de los casos es la siguiente:

- Caso 100 → 60%
- Caso 101 → 71.4286%
- Caso 111 → 68.5714%

5.2.3. Observaciones

A modo de prueba, se ha recortado cada una de las imágenes de la librería ORL, reduciendo su tamaño a los 20x20 píxeles de la esquina inferior derecha [17], y se ha procedido a su clasificación, obteniendo los resultados mostrados en la figura 5.15.

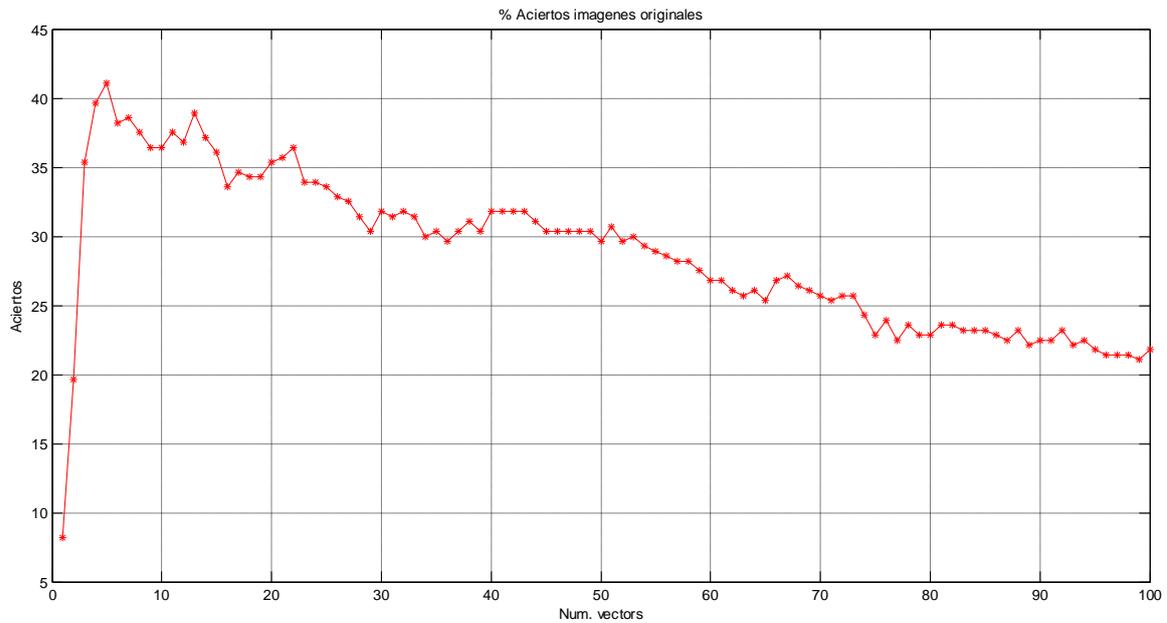


Figura 5.15: Aciertos de las imágenes originales recortadas.

Esta gráfica muestra un porcentaje de aciertos de más del 40% en el caso de tomar $n_vectors=5$. De este resultado puede deducirse que el fondo de la imagen, además del propio rostro, proporciona gran cantidad de información y en algunos casos puede ser un factor decisivo a la hora de discriminar entre las clases. He aquí el porqué de muchas de las clasificaciones erróneas en el caso de las imágenes originales; los sujetos no se confunden por las similitudes entre sus apariencias, sino porque el fondo de las imágenes confundidas resulta bastante parecido. Por ello que la aplicación conjunta del detector Viola-Jones y la máscara elipsoidal mejore los resultados, ya que se elimina el fondo, posible fuente de confusión.

5.3. Experimentos con la base de datos UMH

Los parámetros designados para estas pruebas son: $objects=17$; $np=30$; $n_train=3$; $n_vectors=17$; $see=1$. Para esta base de datos se han tomado diferentes conjuntos de imágenes de test y se ha ejecutado el programa *selección_prep_umh.m*, obteniéndose en los resultados mostrados a continuación:

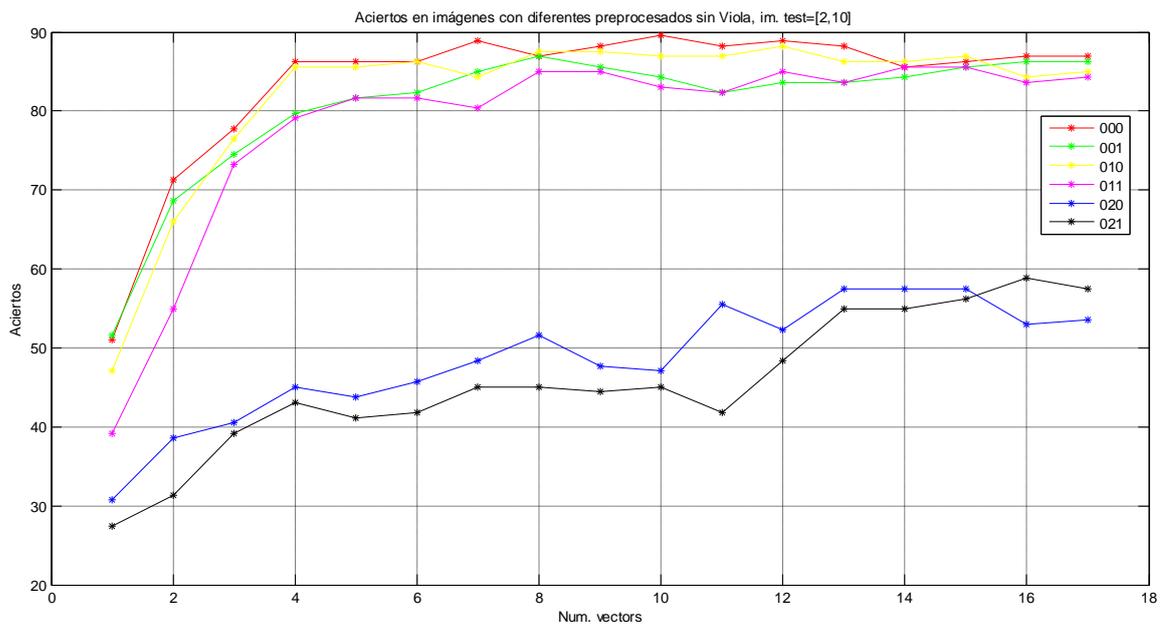


Figura 5.16: Porcentaje de aciertos en imágenes con diferentes preprocesados sin la detección Viola-Jones. Conjunto de imágenes de test [2,10].

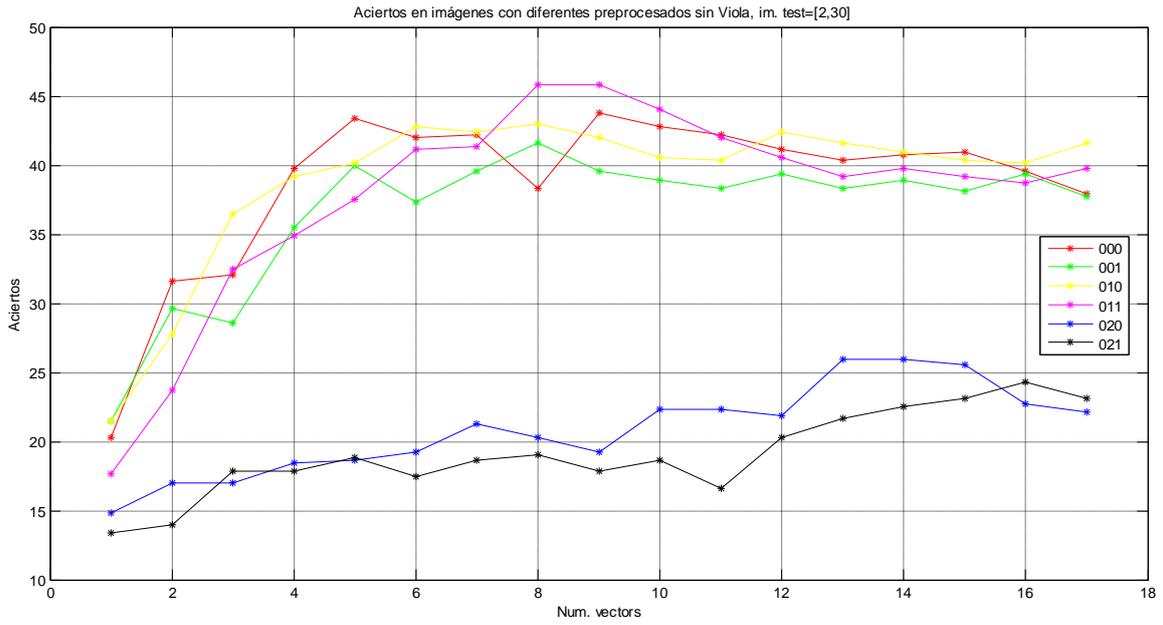


Figura 5.17: Porcentaje de aciertos en imágenes con diferentes preprocesados sin la detección Viola-Jones. Conjunto de imágenes de test [2,30].

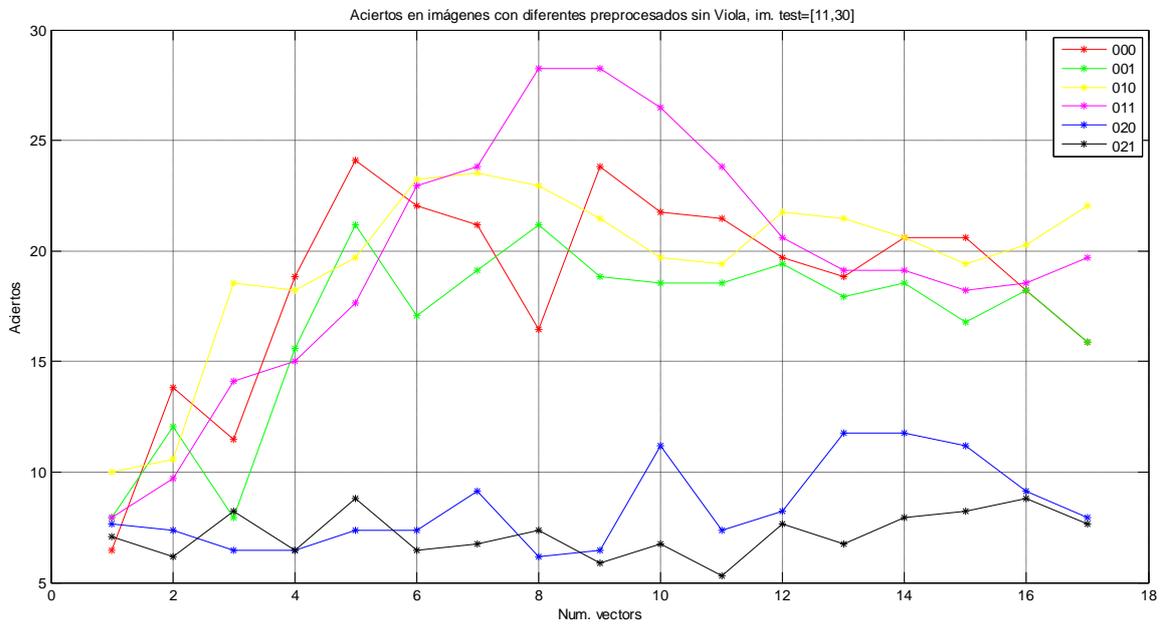


Figura 5.18: Porcentaje de aciertos en imágenes con diferentes preprocesados sin la detección Viola-Jones. Conjunto de imágenes de test [11,30].

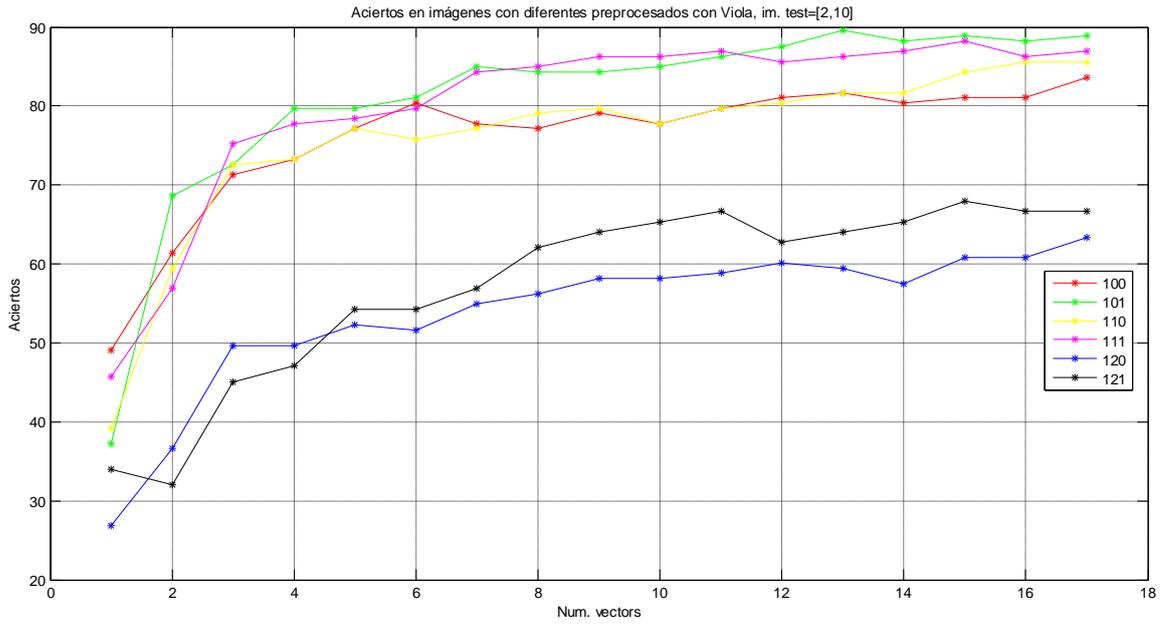


Figura 5.19: Porcentaje de aciertos en imágenes con diferentes preprocesados con la detección Viola-Jones. Conjunto de imágenes de test [2,10].

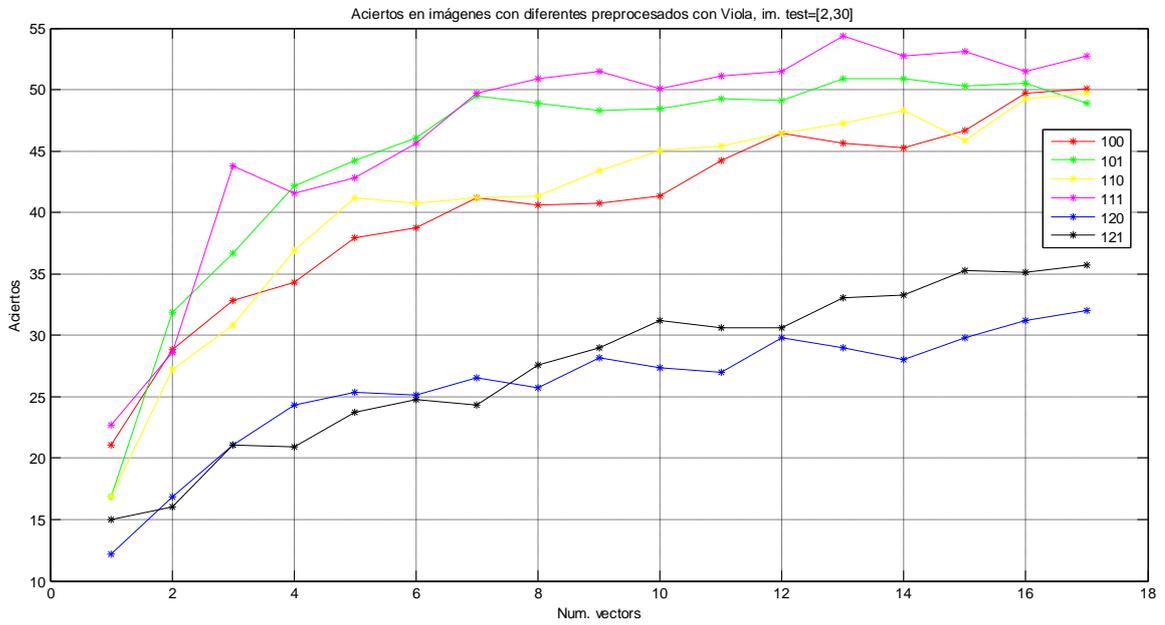


Figura 5.20: Porcentaje de aciertos en imágenes con diferentes preprocesados con la detección Viola-Jones. Conjunto de imágenes de test [2,30].

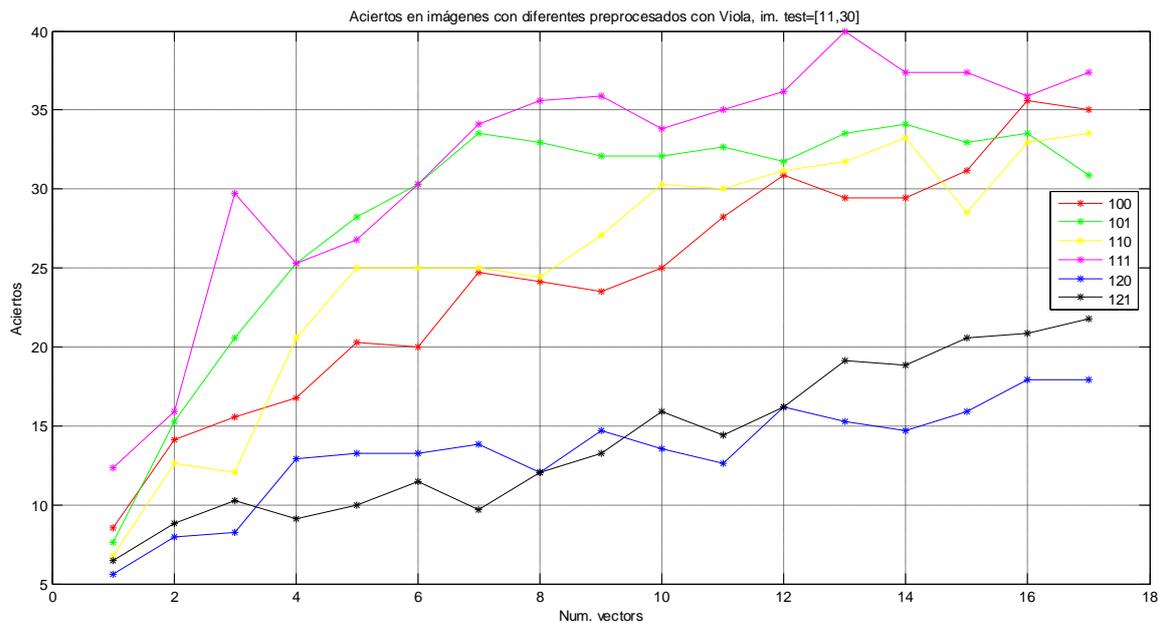


Figura 5.21: Porcentaje de aciertos en imágenes con diferentes preprocesados con la detección Viola-Jones. Conjunto de imágenes de test [11,30].

Para una mejor visualización, las figuras 5.22, 5.23 y 5.24 recogen las comparativas, para cada uno de los conjuntos de entrenamiento, entre las imágenes originales y los resultados que proporcionan mejoras en la clasificación.

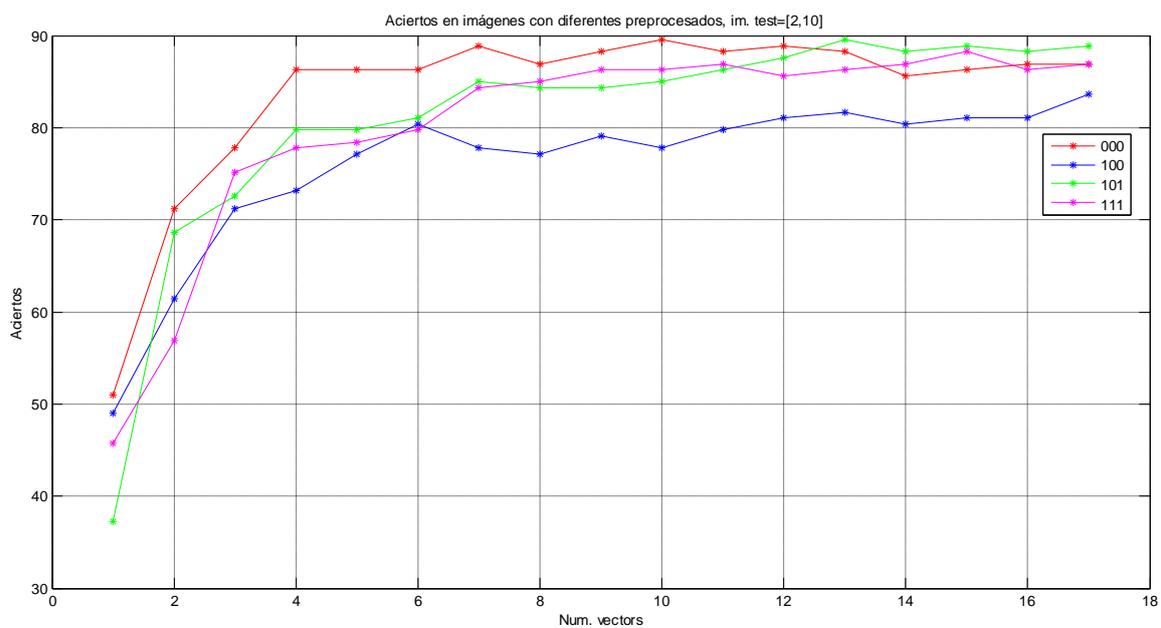


Figura 5.22: Comparativa de aciertos para el conjunto de imágenes de test [2,10].

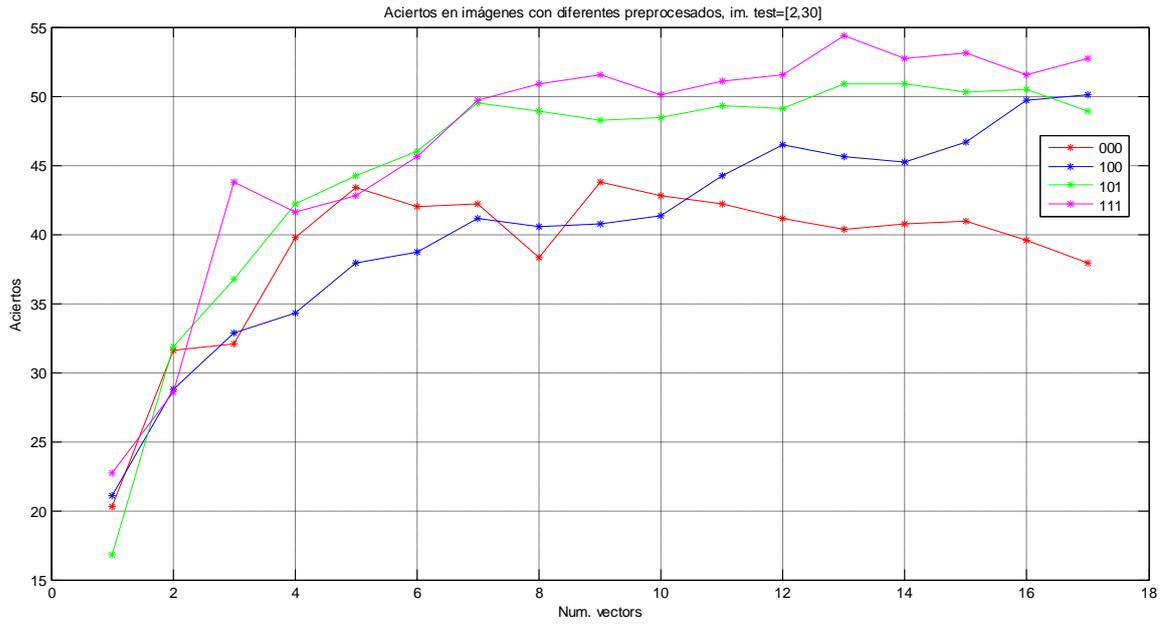


Figura 5.23: Comparativa de aciertos para el conjunto de imágenes de test [2,30].

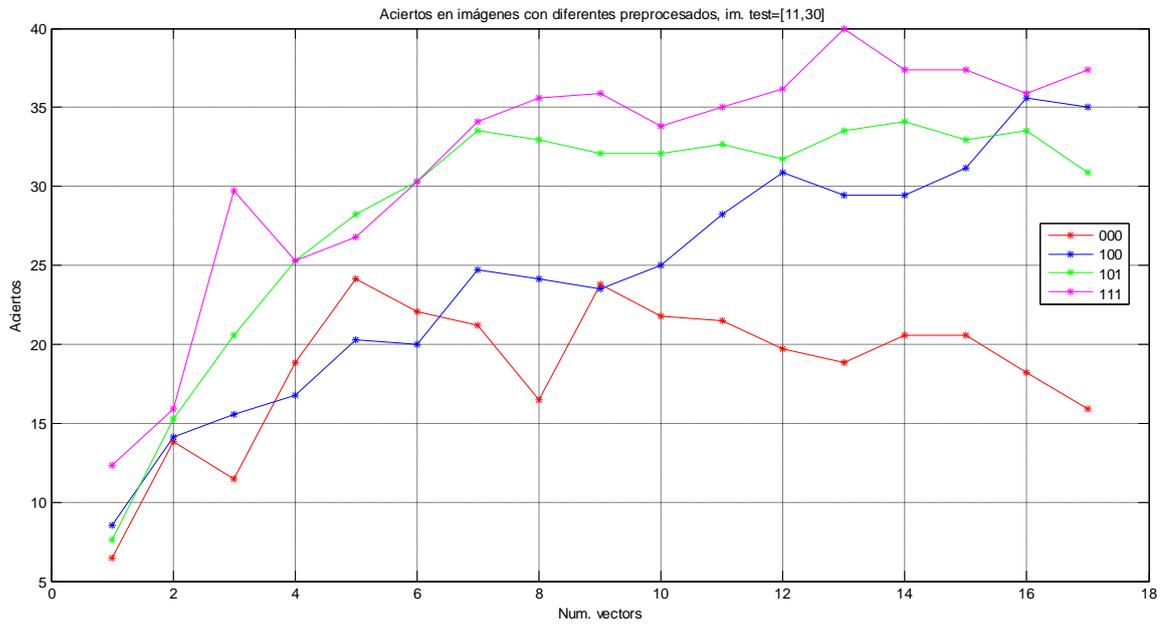


Figura 5.24: Comparativa de aciertos para el conjunto de imágenes de test [11,30].

Una primera conclusión que puede extraerse de la clasificación, es el porcentaje de aciertos según el conjunto de imágenes de test; puesto que las 10 primeras imágenes de la librería UMH son muy parecidas y las variaciones entre ellas son muy pequeñas, al tomar el conjunto de test [2,10], el porcentaje de aciertos es muy alto, mientras que si se toma el conjunto [2,30] o el [11,30] los porcentajes descienden considerablemente, ya que las imágenes correspondientes al último intervalo difieren mucho de las primeras.

Al igual que ocurría con la base de datos ORL, en general los preprocesados de los casos 101 y 111 mejoran la bondad media con respecto a las imágenes originales, pero sin embargo, en este caso, para los conjuntos de imágenes de test [2,30] y [11,30] la detección de Viola-Jones sí mejora los resultados.

De la misma forma, la ecualización Tan-Triggs empeora en todos los casos los resultados, ya que las imágenes de esta librería también fueron tomadas bajo buenas condiciones de iluminación.

5.4. Experimentos con la base de datos Yale B Extended

Los parámetros designados para estos experimentos son: $objects=38$; $np=10$; $n_{train}=3$; $n_{vectors}=114$; $see=1$. Para visualizar los resultados se ha lanzado el programa *selección_prep_yale_tot.m*; los resultados obtenidos se recogen en la figura 5.25.

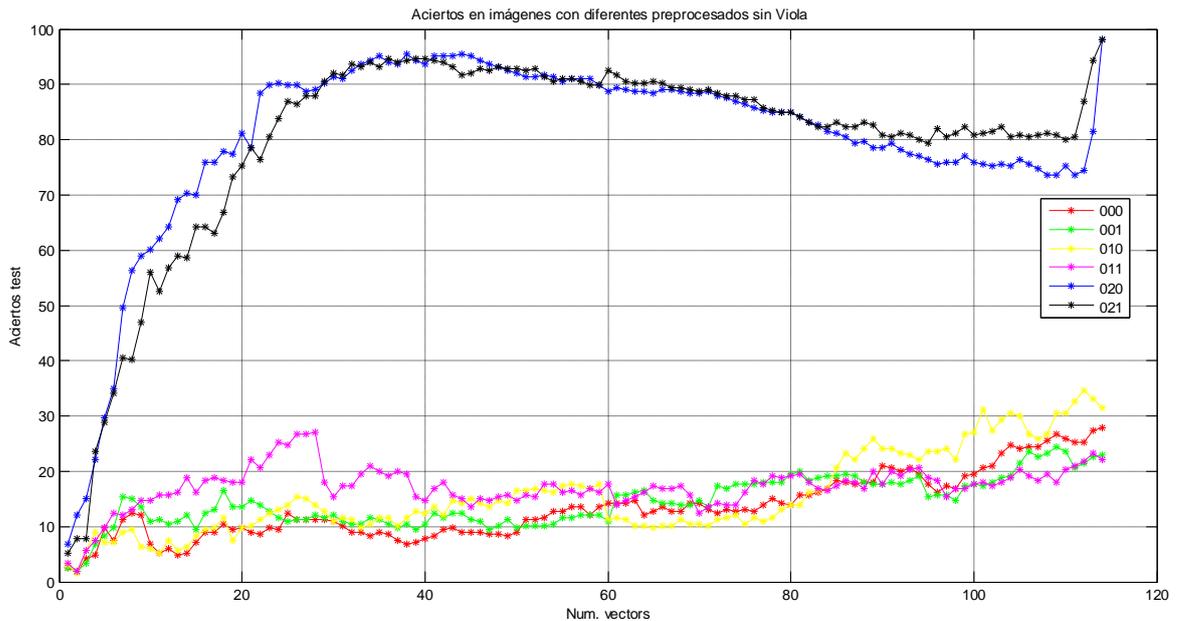


Figura 5.25: Porcentaje de aciertos en imágenes con diferentes preprocesados.

Sorprendentemente, para esta base de datos, la ecualización Tan-Triggs supone una extraordinaria mejora en la clasificación, tal y como puede distinguirse en la figura 5.26; en el caso de las imágenes originales el porcentaje de aciertos era del 27.8195%, mientras que para los casos 020 y 021 es del 98.1203%. Las tablas 5.7, 5.8 y 5.9 muestran las matrices de confusión pertenecientes a estos preprocesados.

Las imágenes de esta base de datos fueron tomadas bajo numerosas condiciones de iluminación, en muchos casos siendo ésta muy escasa, impidiendo prácticamente el reconocimiento visual de los individuos. Por ello la ecualización Tan-Triggs resulta un preprocesado óptimo para imágenes en este tipo de condiciones.

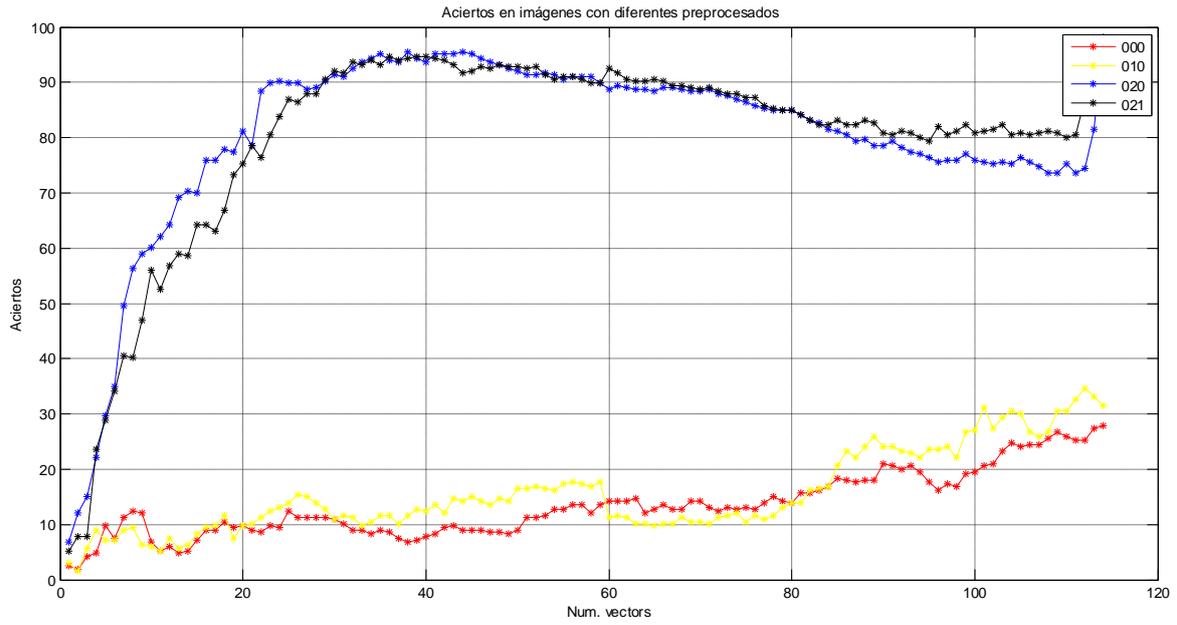


Figura 5.26: Comparativa de aciertos entre los mejores preprocesados.

En la figura 5.26 también puede observarse la pequeña mejora inducida tras la aplicación de la ecualización normalizada, aunque en este caso no es muy significativa.

Capítulo 6

Conclusiones

6.1. Conclusiones

En este proyecto final de carrera se aborda el problema de la identificación de caras, estudiando el efecto de diferentes preprocesados aplicados sobre las imágenes.

El trabajo con imágenes presenta principalmente dos problemas: alta variabilidad inclusive para una misma persona y muy alta dimensionalidad. Para resolver estos problemas se ha escogido la técnica de extracción de características PCA, que permite reducir considerablemente la dimensionalidad de los datos. Trabajando con 3 bases de datos diferentes, se ha aplicado a las imágenes de todas ellas distintas técnicas de preprocesado y se ha estudiado su efecto sobre la identificación de los individuos.

En el primer experimento se trabajó con la base de datos ORL, empleada habitualmente en trabajos de investigación; entre todos los preprocesados objeto del presente estudio, se observó que la detección Viola-Jones junto con la máscara elipsoidal mejoraba modestamente la clasificación de los sujetos de la base de datos. Sin embargo, en aquellos casos en los que la ecualización Tan-Triggs estaba presente, el porcentaje de aciertos disminuía notablemente. De esto puede deducirse que para el caso de imágenes tomadas en óptimas condiciones de iluminación, la ecualización Tan-Triggs no resulta útil. Por otra parte, en el caso de que las imágenes contengan demasiado fondo y éste no sea muy característico (hasta el punto de suponer un factor discriminatorio clave en la identificación), la detección de Viola-Jones y la elipse ayudan a trabajar únicamente con la información relativa al rostro suponiendo una pequeña mejora en la clasificación.

La base de datos UMH, utilizada en el segundo experimento, constituía una base de datos real y sus imágenes fueron tomadas bajo distintas condiciones de iluminación y en diferentes escenarios. Por ello, al igual que con la anterior base de datos, los preprocesados de Viola-Jones y la máscara elipsoidal inducían mejoras, mientras que la ecualización Tan-Triggs empeoraba los resultados. No obstante, puesto que la iluminación en muchas imágenes no era muy buena, la ecualización normalizada sí resultaba útil.

En el tercer y último experimento se empleó la base de datos Yale B Extended; en este caso, muchas de las imágenes estaban muy oscuras, por lo que identificar a los individuos resultaba una tarea muy difícil, lo que se traducía en porcentajes de aciertos muy bajos para las imágenes originales. En este caso la ecualización Tan-Triggs mejoraba la identificación hasta tal punto de alcanzar porcentajes del 98%.

En resumen, en primer lugar puede concluirse que los preprocesados de Viola-Jones y la máscara elipsoidal son muy buenos para cuando en las imágenes aparece no únicamente el rostro de los individuos a clasificar. En segundo lugar, la ecualización normalizada mejora la tarea de identificación siempre y cuando las condiciones de iluminación no sean muy precarias; y por último, y en tal caso de precariedad es cuando puede recurrirse a la ecualización Tan-Triggs.

Bibliografía

BIBLIOGRAFÍA

1. **Vicente Ripoll, M^a Asunción.** *Reconocimiento y búsqueda selectiva de objetos en entornos no estructurados mediante apariencia multicanal y multiescala. Tesis doctoral.* Elche, Alicante. 2006.
2. **M.A. Turk and A. Pentland.** *Eigenfaces for recognition. Journal of Cognitive Neuroscience*, 3(1):71-86. 1991.
3. **M. Kirby and L. Sirovich.** *Application of karhunen-loeve procedure for characterization of human faces. IEEE Transaction on Pattern Analysis and Machine Intelligence*, (12):103-108. 1990.
4. **H. Murakami and V. Kumar.** *Efficient calculation of primary images from a set of images. IEEE Trans. on Pattern Analysis and Machine Intelligence*, (4(5)): 511-515. 1982.
5. **K.I. Diamantaras and S.Y. Kung.** *Principal Component Neural Networks: Theory and Applications.* Wiley. 1996.
6. **A. Hyvärinen, J. Karhunen, and E. Oja.** *Independent Component Analysis.* Wiley-Interscience. New York. 2002.
7. **A. Hyvärinen.** *Fast and robust fixed-point algorithms for independent component analysis. IEEE Transactions on Neural Networks*, 10(3):626–634. 1999.
8. **A.J Bell and T.J. Sejnowski.** *An information-maximization approach to blind separation and blind deconvolution. Neural Computation*, 7(1129-1159). 1995.
9. **Carlos Platero Dueñas.** Capítulo 4: Técnicas de preprocesado. [En línea] <http://www.elai.upm.es/spain/Asignaturas/Robotica/ApuntesVA/cap4Procesadov1.pdf>. 2009.
10. **Wikipedia.** [En línea]. http://en.wikipedia.org/wiki/Viola-Jones_object_detection_framework.
11. **P. Viola and M. Jones.** *Robust Real-time Object Detection.* Vancouver, Canada. 2001.
12. **M.L. Guevara, J.D. Echeverry and W.A. Ureña.** *Faces Detection in Digital Images Using Cascade Classifiers. Scientia et Technica Año XIV, No 38, Junio de 2008.* Universidad Tecnológica de Pereira. ISSN 0122-1701.
13. **Santillán, Iván Danilo García.** *Visión Artificial y Procesamiento Digital de Imágenes usando Matlab (Cap. 5).* IBARRA – ECUADOR, 2008. ISBN: 978-9942-01-790-1.

14. **ORL face database.** [En línea]
<<http://www.uk.research.att.com/facedatabase.html>>..
15. **Database, Yale B Extended.** [En línea]
<http://cvc.yale.edu/projects/yalefacesB/yalefacesB.html>
16. **Xiaoyang Tan and Bill Triggs.** *Preprocessing and Feature Sets for Robust Face Recognition.* Montbonnot, France.
17. **Lior Shamir.** *Evaluation of Face Datasets as Tools for Assessing the Performance of Face Recognition Methods.* 79: 225-230. 2008.