



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

— **TELECOM** ESCUELA
TÉCNICA **VLC** SUPERIOR
DE **UPV** INGENIEROS
DE TELECOMUNICACIÓN

ALGORITMO DE RECONOCIMIENTO DE PATRONES BASADO EN CODIFICACIÓN FISIOLÓGICA EN CEREBRO DE PRIMATES

Alejandro Castel Baixauli

Tutor: Andreu M. Climent

Cotutor: María S. Guillem

Trabajo Fin de Grado presentado en la Escuela
Técnica Superior de Ingenieros de
Telecomunicación de la Universitat Politècnica de
València, para la obtención del Título de Graduado
en Ingeniería de Tecnologías y Servicios de
Telecomunicación

Curso 2017-18

Valencia, 3 de diciembre de 2017

RESUMEN

Recientemente se han identificado los mecanismos biológicos por los cuales el cerebro primate es capaz de codificar, almacenar e identificar los rostros mediante un número limitado de neuronas. La utilización de algoritmos de reconocimiento basados en la codificación fisiológica tiene el potencial de revolucionar las capacidades de las técnicas de inteligencia artificial.

El objetivo del presente trabajo es desarrollar y evaluar la eficacia de dos algoritmos de reconocimiento de patrones bioinspirados que utilizan la descomposición en características fundamentales para la identificación de rostros humanos.

Se ha implementado y evaluado los algoritmos (1) *Eigenfaces*, que superpone linealmente componentes principales, y (2) *Active Appearance Model (AAM)*, que muestrea datos relativos a la forma y la apariencia del rostro. Además, se han comparado sus resultados, resaltando así sus fortalezas y también sus debilidades. La eficacia de dichos algoritmos se ha validado mediante la evaluación de la capacidad de los mismos para el reconocimiento facial de un rostro humano. Se ha utilizado un banco de 200 imágenes de 320x260 píxeles del Laboratorio de Inteligencia Artificial FEI, en Sao Paulo. Se han extraído las principales características de cada imagen para realizar una abstracción de un '*espacio cara*' a través del cual poder codificar cada identidad en un vector de números concretos.

Los algoritmos desarrollados han sido capaces de almacenar la información de cada uno de los rostros de acuerdo a unos ejes de información fundamental. Concretamente, vectores de 199 ejes ortogonales, específicos e individuales a cada identidad. Lo que supone una reducción de un 99% con respecto a almacenar una imagen completa. A través de esta codificación, hemos podido reconstruir todos los rostros de nuestra base de datos, así como identificarlo correctamente con más de un 90% de éxito para el caso del algoritmo AAM y de más de un 80% para el caso Eigenfaces.

Gracias a Eigenfaces y AAM, somos capaces de obtener un vector del rostro codificado que define su apariencia, lo que facilita su almacenamiento y procesamiento. Mejorando el modelo, somos capaces de aumentar la identificación facial basándonos en los recientes estudios sobre la codificación facial en el cerebro de los primates, aumentando la eficiencia y la velocidad de manera significativa.

Actualment han sigut identificats els mecanismes biològics pels quals el cerebro primate pot codificar, emmagatzemar e identificar els rostres mitjançant un número limitat de neurones. L'utilització d'algoritmes de reconeixement basats en la codificació fisiològica té el potencial de revolucionar les capacitats de les tècniques d'intel·ligència artificial.

L'objectiu del present treball és desenvolupar i avaluar l'eficàcia de dos algoritmes de reconeixement de patrons bioinspirats que utilitzen la descomposició en característiques fonamentals per a la identificació de rostres humans.

S'ha implementat i avaluat els algoritmes (1) Eigenfaces, que superposa linealment components principals, i (2) Active Appearance Model (AAM), que mostreja dades relatives a la forma i l'aparença de la cara. A més, s'han comparat els seus resultats, ressaltant així les seves fortaleeses i també les seves debilitats. L'eficàcia d'aquests algoritmes s'ha validat mitjançant l'avaluació de la capacitat dels mateixos per al reconeixement facial d'un rostre humà. S'ha utilitzat un banc de 200 imatges de 320x260 píxels del Laboratori d'Intel·ligència Artificial FEI, a Sao Paulo. S'han extret les principals característiques de cada imatge per realitzar una abstracció d'un 'espai cara' a través del qual poder codificar cada identitat en un vector de números concrets.

Els algoritmes desenvolupats han estat capaços d'emmagatzemar la informació de cada una de les cares d'acord a uns eixos d'informació fonamental. Concretament, vectors de 199 eixos ortogonals, específics i individuals a cada identitat. El que suposa una reducció d'un 99% respecte a emmagatzemar una imatge completa. A través d'aquesta codificació, hem pogut reconstruir tots els rostres de la nostra base de dades, així com identificar correctament amb més d'un 90% d'èxit per al cas de l'algoritme AAM i de més d'un 80% per al cas Eigenfaces.

Gràcies a Eigenfaces i AAM, som capaços d'obtenir un vector de la cara codificat que defineix la seva aparença, el que facilita el seu emmagatzematge i processament. Millorant el model, som capaços d'augmentar la identificació facial basant-nos en els recents estudis sobre la codificació facial en el cervell dels primats, augmentant l'eficiència i la velocitat de manera significativa.

ABSTRACT

Recently, the biological mechanisms by which the primate brain is able to encode, store and identify faces through a limited number of neurons have been identified. The use of recognition algorithms based on physiological coding has the potential to revolutionize the capabilities of artificial intelligence techniques.

The objective of the present work is to develop and evaluate the effectiveness of two algorithms of recognition of bioinspired patterns that use the decomposition in fundamental characteristics for the identification of human faces.

We have implemented and evaluated the algorithms (1) Eigenfaces, which linearly superimpose principal components, and (2) Active Appearance Model (AAM), which samples data related to the shape and appearance of the face. In addition, their results have been compared, highlighting their strengths and also their weaknesses. The effectiveness of these algorithms has been validated through the evaluation of their capacity for facial recognition of a human face. A bank of 200 images of 320x260 pixels from the FEI Artificial Intelligence Laboratory in Sao Paulo has been used. The main features of each image have been extracted to make an abstraction of an 'face space' through which to encode each identity in a vector of specific numbers.

The developed algorithms have been able to store the information of each of the faces according to fundamental information axes. Specifically, vectors of 199 orthogonal axes, specific and individual to each identity. What is a reduction of 99% with respect to store a complete image. Through this coding, we have been able to reconstruct all the faces of our database, as well as correctly identify it with more than 90% success in the case of the AAM algorithm and more than 80% for the Eigenfaces case.

Thanks to Eigenfaces and AAM, we are able to obtain a vector of the coded face that defines its appearance, which facilitates its storage and processing. By improving the model, we are able to increase facial identification based on recent studies on facial coding in primate brains, increasing efficiency and speed significantly.

0. Índice

1. Introducción	6
2. Objetivos	10
3. Estado del arte	11
3.1 Reconocimiento de imágenes basado en la apariencia	13
3.2 Reducción de dimensionalidad y extracción de características	15
3.3 Técnicas de reconocimiento facial actuales	20
3.4 Modelo de ejes (Axis Model)	22
3.5 Active Appearance Model	23
4. Metodología	27
4.1 Base de datos	27
4.2 Software utilizado	28
4.3 Desarrollo algoritmo Eigenfaces	28
4.4 Desarrollo algoritmo AAM	30
4.5 Medición de tiempos	34
4.6 Reducción de codificación	34
4.7 Identificación bajo condiciones de ruido y luminosidad adversas	36
5. Resultados	37
5.1 Reconstrucción facial en algoritmo AAM	37
5.2 Identificación facial	38
5.3 Resultados eliminando componentes	40
5.4 Identificación en condiciones de ruido e iluminación adversas	41
6. Presupuesto de desarrollo	44
7. Discusión	45
8. Conclusión	47
9. Bibliografía y referencias	48

1. Introducción

Durante los últimos años, el reconocimiento facial se ha convertido en uno de las aplicaciones más estudiadas en campos como la biometría, el procesado de imagen o el reconocimiento de patrones. Una de las razones que ha llevado a este crecimiento son las necesidades cada vez mayores de aplicaciones de seguridad y vigilancia utilizadas en diferentes ámbitos.

El creciente interés en el reconocimiento facial y los sistemas biométricos, lo podemos ver reflejado de 3 maneras distintas:

- En primer lugar, se puede ver en la comunidad científica donde por ejemplo se pasó de la publicación de unos 140 artículos en 2007, relacionados con el reconocimiento y diferentes herramientas para llevarlo a cabo, a través de publicaciones en diferentes organismos como el IEEE, a unos 180 en 2008, incluyendo conferencias como el *“IEEE internacional conference on automatic face and gesture recognition”* (IEEE FG 2008).
- En segundo lugar, se puede observar en el aumento en el número de soluciones comerciales y el crecimiento de este mercado. Ejemplo de ello son los sistemas basados en el reconocimiento para identificación civil/criminal y los sistemas de video vigilancia desarrollados por empresas como Cognitec[1] e Identix[2].
- En tercer y último lugar, se ve reflejado este aumento en la industria del reconocimiento facial desde el punto de vista de los clientes, los cuales cada vez solicitan más productos funcionales basados en esta tecnología. Un claro ejemplo es el sistema de desbloqueo del último dispositivo presentado recientemente por la compañía Apple, el iPhone X [3].

Estos sistemas en su objetivo de determinar la identidad de una persona, pueden tener en cuenta o analizar muchas características. Estas características pueden ser tanto físicas, como sería el caso de las pupilas, la cara, el iris, etc., o se puede tratar de características psicológicas como serían los gestos. Todas estas propiedades, presentan sus propias peculiaridades, las cuales pueden ser analizadas siguiendo entre otros, los siguientes criterios:

- Universalidad: indica como de común es encontrar esta característica en todas las personas u objetos a reconocer.
- Carácter distintivo: indica si dicha propiedad, es suficientemente diferente entre un conjunto de personas u objetos diferentes.
- Permanencia: indica la estabilidad en el tiempo de dicha característica.
- Colectividad: indica si la característica es fácilmente adquirida y medida por el sistema.
- Rendimiento: indica la precisión, velocidad y coste (recursos) necesarios para llevar a cabo el reconocimiento.

- Aceptabilidad: indica en qué medida está la gente preparada para aceptar el uso de esta técnica.
- Elusión: indica la respuesta del sistema cuando alguien está tratando de engañarlo.

Gracias a estas propiedades que se han enumerado, se puede describir bien las características que va a utilizar el sistema y así poder ver si es adecuado o no para este tipo de aplicaciones. Se pueden encontrar otros modos de evaluar estas características, uno de ellos es el análisis de Zephyr[4] gracias al cual se pueden clasificar diferentes indicadores biométricos según su carácter intrusivo, precisión, coste y dificultad. Según otras escalas, considerando 6 parámetros biométricos, las características faciales obtienen la mayor puntuación para sistemas de lectura de documentación de viaje MRTD[5](Figura 1.1).

Según todos estos criterios que se acaban de ver, el reconocimiento facial destaca por ser una técnica con una alta capacidad de respuesta frente a múltiples características biométricas y una buena aceptación lo cual hace de los sistemas basados en estos indicadores útiles para el desarrollo de aplicaciones no intrusivas.

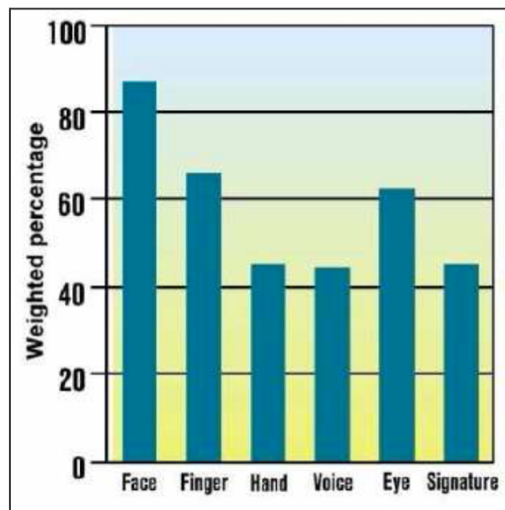


Figura 1.1 - Clasificación de características biométricas según compatibilidad MRTD[34].

Por otro lado, las aplicaciones basadas en el reconocimiento facial actuales, presentan algunas limitaciones o inconvenientes como pueden ser:

- Orientación del rostro.
- Ruido.
- Iluminación (incluyendo interior / exterior).
- Expresión facial.
- Oclusión debido a objetos o accesorios tales como gafas de sol, sombreros...
- Vello facial.
- Envejecimiento.

Aun así, algunas de estas barreras, se puede solucionar de una forma eficaz como sería el caso del envejecimiento. Para ello bastaría con una actualización periódica de la

base de datos de la aplicación. Un ejemplo de imágenes de un mismo individuo a medida que avanza en el tiempo para evaluar el sistema frente a estas limitaciones. Es posible también entrenar los modelos para que estén preparados a la hora de recibir rostros desde un perfil distinto al frontal. Del mismo modo con expresiones faciales distintas.

Debido a que las aplicaciones basadas en el reconocimiento facial [6] no necesitan de la interacción del individuo, son aplicaciones de mucho interés para el entorno de video vigilancia, aeropuertos, comisarías, etc. De igual modo son aplicaciones necesarias y de utilidad en entornos controlados donde por ejemplo se pueda llevar a cabo el testeado de las aplicaciones antes de salir al mercado.

En toda aplicación de reconocimiento facial, se debe tener en cuenta diferentes etapas. Una de ellas es la etapa de detección facial, la cual en multitud de ocasiones se trata de manera conjunta con el reconocimiento. En este proyecto tendremos en cuenta que la detección facial se ha realizado correctamente, por simplicidad, aunque se explicará brevemente. Sin embargo, no es este el objetivo del proyecto, si no centrarse en el nuevo modelo de codificación y reconocimiento propuesto según la fisiología de los primates. De este modo el sistema de reconocimiento recibirá como entrada un conjunto de imágenes 'cara', ya detectadas con diferentes expresiones y condiciones ambientales, las cuales serán procesadas por nuestro sistema para después hacer el reconocimiento (Figura 1.2).

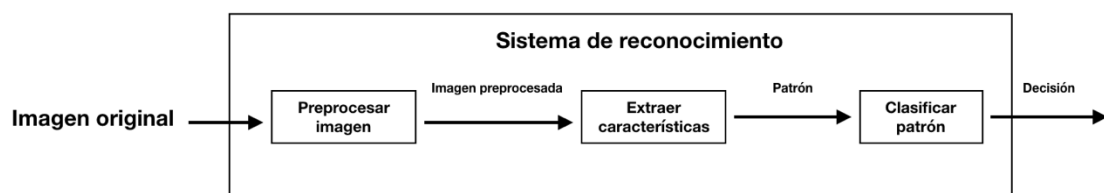


Figura 1.2 - Diagrama de bloques general en un sistema de reconocimiento facial.

Finalmente se puede decir que estamos ante un problema de reconocimiento de patrones, pues el objetivo será el de clasificar y reconocer imágenes cara dependiendo de sus características. Dentro de esta rama se pueden encontrar 2 grandes categorías, sistemas supervisados y sistemas no supervisados. La diferencia entre ambos casos reside en que el caso supervisado conoce con antelación a quien pertenece cada imagen y asigna una etiqueta identificadora a cada imagen, mientras que los no supervisados no lo hacen. En el caso de este proyecto nos enfrentaremos a casos no supervisados.

Comentaremos brevemente diferentes técnicas de extracción de características en una imagen de una cara y nos centraremos en el análisis PCA, que será el utilizado en nuestros algoritmos. Las tareas que desarrollaremos se pueden enumerar cronológicamente del siguiente modo:

- Estudio de las diferentes técnicas de selección de características, prestando especial atención a PCA (Principal Component Analysis).
- Evaluación de los principales problemas como pueden ser los cambios en la iluminación y en la expresión o posición de la cara.
- Implementación del algoritmo en Matlab.

- Testeo de todo el sistema utilizando imágenes frontales y con diferentes condiciones de iluminación y ruido.

2. Objetivos

El objetivo del presente trabajo es desarrollar y evaluar dos algoritmos de reconocimiento de patrones, conocidos como Eigenfaces[7] y Active Appearance Model[8], que nos permitan identificar rostros humanos dentro de una base de datos. Estos algoritmos bioinspirados extraen las principales características de cada imagen y la comparan con las características almacenadas de individuos en una base de datos previamente entrenada. La peculiaridad de estos algoritmos es que son capaces de codificar todas estas características en un único vector que las define. El primer objetivo del trabajo es obtener dicho vector.

Una vez obtenido, se comprobará que se ha realizado el análisis de componentes correctamente reconstruyendo, para el caso del AAM, cada rostro a partir del vector obtenido. Se comprobará que dicha reconstrucción ha sido satisfactoria a partir del vector completo de componentes.

Confirmada la reconstrucción, demostraremos la efectividad de ambos algoritmos a la hora de identificar individuos en condiciones óptimas de captación de imágenes. Comprobaremos qué algoritmo es más fiable a la hora de identificar rostros humanos y por qué.

Demostraremos que el número de componentes necesarias para obtener el rostro original puede reducirse significativamente, puesto que no todas las componentes llevan implícita información de la misma importancia. Observaremos que algunas componentes son más importantes que otras a la hora de reconstruir y codificar una cara y, por lo tanto, a la hora de identificarla, por lo que podremos eliminarlas. Comprobaremos hasta donde es capaz el algoritmo de reconocer un rostro eliminando ciertas componentes para obtener una aproximación del nivel de codificación máximo que podemos alcanzar.

Tomaremos tiempos de procesamiento e identificación para comparar ambos algoritmos y comparar cuál de los dos es más rápido y eficiente y demostrar que el algoritmo Eigenfaces lo es, puesto que no requiere de un procesamiento previo de las imágenes. Esto nos permitirá concluir en qué situaciones podremos preferir implementar un algoritmo u otro.

Por último, someteremos a ambos algoritmos a pruebas de robustez. Añadiremos a las imágenes de entrada condiciones adversas de iluminación y de ruido blanco gaussiano que nos permitirán monitorizar la fiabilidad de dichos algoritmos. En este apartado demostraremos como el nivel de acierto disminuye conforme empeoramos la calidad de las imágenes, como cabría esperar, sin embargo, demostraremos que las condiciones de iluminación afectan más en la detección que las interferencias por ruido gaussiano. Además, probaremos que el AAM es un sistema más robusto en cuando a la tolerancia a interferencias.

Con estos experimentos evaluaremos que algoritmo se comporta mejor y que semejanzas guardan con la codificación facial en el cerebro de los primates.

3. Estado del Arte

Podemos diferenciar dos familias de técnicas de reconocimiento facial: técnicas basadas en la apariencia y técnicas basadas en modelos. En cada una de estas familias, se encuentran varios métodos para caracterizar la imagen. Podemos ver un ejemplo de algunos métodos de clasificación en la Figura 3.1.

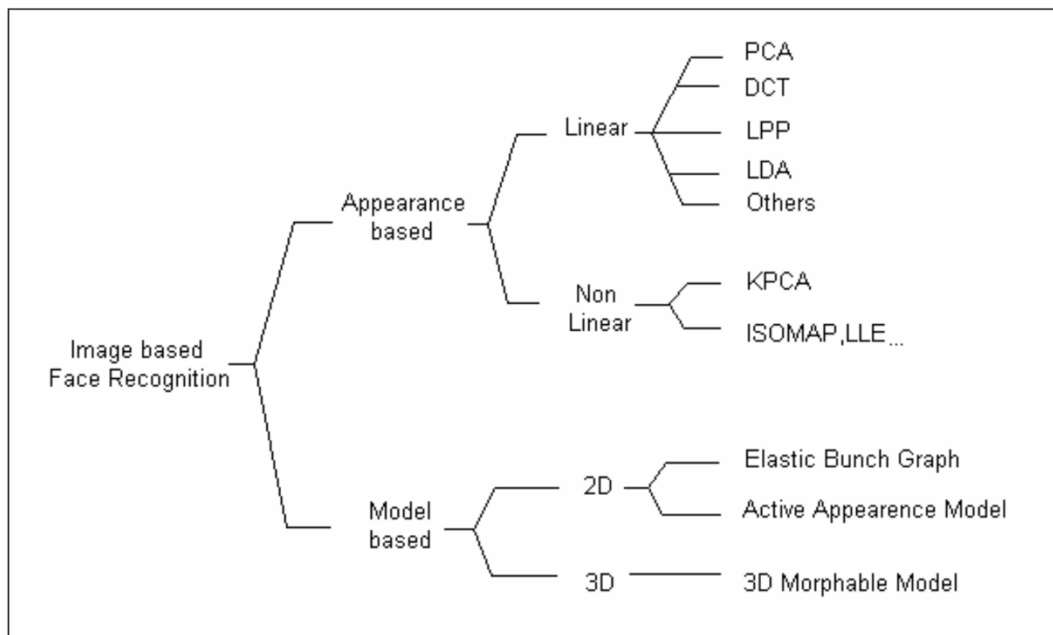


Figura 3.1 - Algunos métodos de clasificación

Los sistemas basados en la apariencia se utilizan directamente sobre las imágenes sin hacer uso de modelos 3D. Estos tipos de sistemas representan un objeto en función de diferentes vistas del mismo. En estos sistemas cada imagen se representa como un punto en un subespacio vectorial, de forma que la comparación entre la imagen de test y las imágenes de referencia se realiza en el subespacio vectorial. El objetivo de estos algoritmos es clasificar las diferentes caras en el nuevo subespacio, pero para ello será necesario entrenar previamente el sistema con imágenes de diferentes caras con diferentes vistas. Estas técnicas serán explicadas con más detalles más adelante donde se expondrán los métodos de clasificación basados en la apariencia que se han utilizado.

Por otro lado están los sistemas basados en modelos (Figura 3.2), los cuales intentan construir un modelo lo más descriptivo posible de la cara humana capaz de detectar con precisión las variaciones faciales.

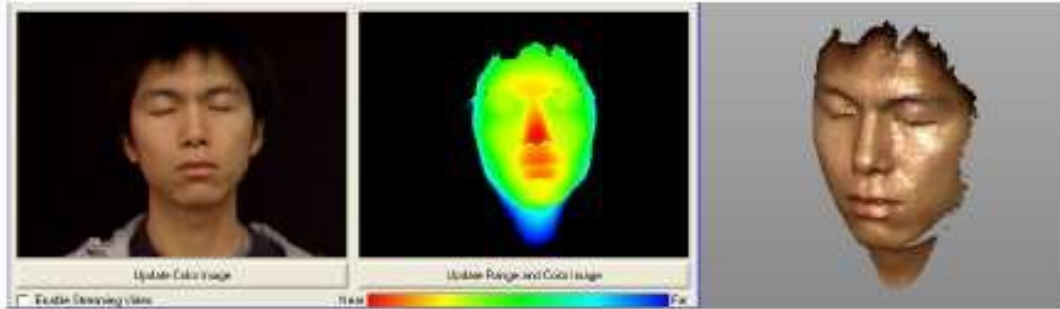


Figura 3.2 - Imagen 2D, mapa de profundidad y representación 3D del modelo.

Estos sistemas tratan de obtener características biométricas de las imágenes para realizar el reconocimiento (distancia entre ojos, grosor de la nariz...)(Figura 3.3). Habitualmente estas técnicas requieren de imágenes de gran resolución lo cual puede ser un problema para ciertas aplicaciones, tales como una cámara de seguridad de un establecimiento. Cuando se utilizan estos sistemas, el algoritmo sabe con antelación el objeto que ha de representar y lo que intenta hacer es que corresponda la cara real con el modelo. El proceso que se suele seguir cuando se usan estas técnicas está formado por tres pasos [9]:

1. Construcción del modelo.
2. Ajustar el modelo a la imagen de test.
3. Utilizar los parámetros del modelo ajustado para calcular la similitud entre la imagen de test y las imágenes de referencia para realizar el reconocimiento.

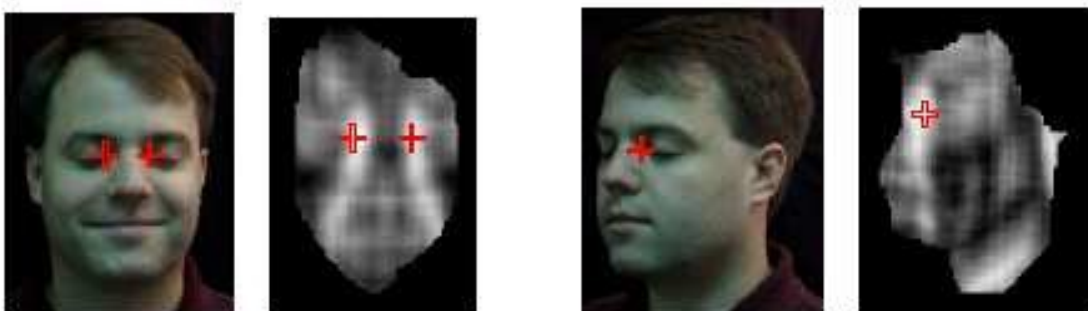


Figura 3.3 - Detección de la posición y distancia de los ojos a través del método de convolución

Algunas de las ventajas e inconvenientes de los sistemas basados en la apariencia frente a los basados en modelos son las siguientes:

- Ventajas:
 - a. Más rápidos.
 - b. Requieren de un menor tamaño de las imágenes.
 - c. Menor complejidad.
 - d. No requieren de un conocimiento previo de las imágenes.
- Inconvenientes:
 - e. Más afectados por cambios en la orientación o expresión de la cara.
 - f. Más dificultades frente a cambios en la iluminación.

A continuación se hará una pequeña descripción de la literatura que existe sobre algunos de los métodos más relevantes de reconocimiento facial y una pequeña explicación de cómo se va a realizar el reconocimiento de las imágenes.

3.1 Reconocimiento de imágenes basado en la apariencia

En este estudio se hará uso de métodos basados en la apariencia, los cuales para el reconocimiento de imágenes requiere de los bloques que podemos encontrar en la Figura 3.4.

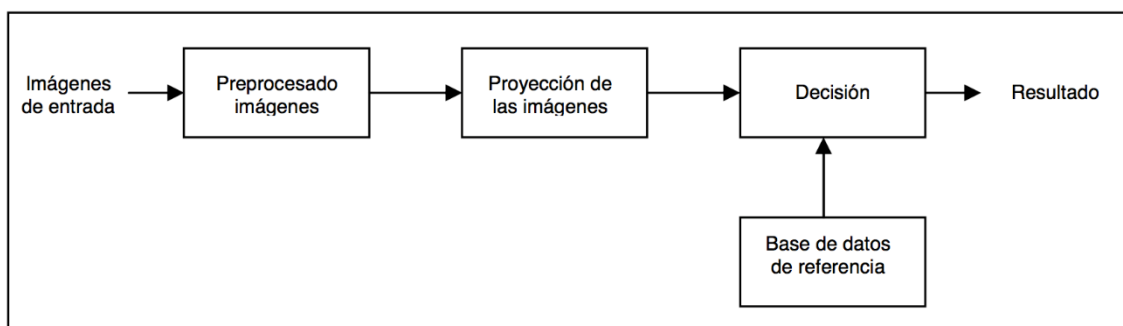


Figura 3.4 - Diagrama de bloques del sistema.

Para realizar los diferentes experimentos que se realizan en este estudio el sistema hace uso de los siguientes conjuntos de imágenes:

1. Imágenes de referencia: son las que están almacenadas y son conocidas por el sistema y son utilizadas para saber si una imagen de entrada pertenece o no a algún sujeto registrado en el sistema.
2. Imágenes de test: son las que recibiremos en el sistema y se tienen que reconocer.
3. Imágenes de entrenamiento: son utilizadas en el método PCA para conseguir las matrices de proyección (el método PCA *Principal Component Analysis*, se

explicará con detalle más adelante).

En nuestro caso, las imágenes almacenadas en la base de datos y las imágenes utilizadas para crear las matrices de proyección serán las mismas.

Los pasos que se llevan a cabo para este reconocimiento, sin tener en cuenta el preprocesado previo de las imágenes, son los que se detallan a continuación [10]:

- 1) El sistema necesita de un set de imágenes de referencia las cuales son procesadas y proyectadas ($y = W \cdot x$) en el nuevo subespacio para después ser utilizadas para conocer si una imagen pertenece o no a un individuo registrado en el sistema (Figura 3.5).

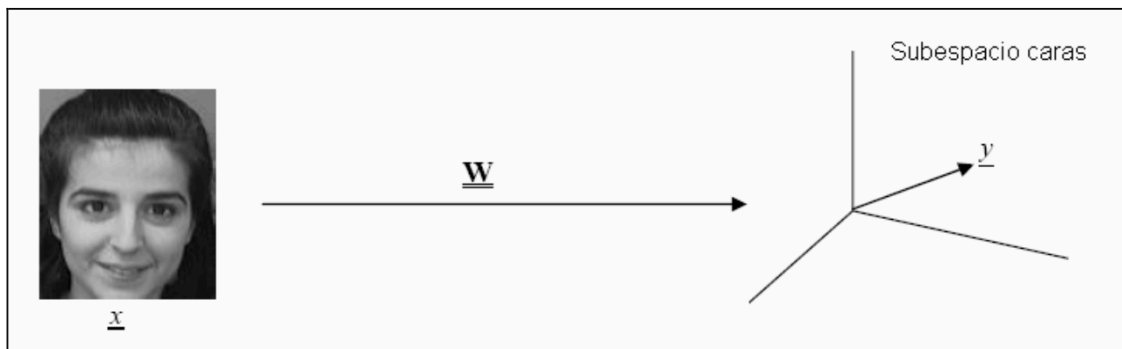


Figura 3.5 - Esquema de proyección de una imagen de referencia al nuevo subespacio.

- 2) Cuando las imágenes de referencia están correctamente proyectadas, se realiza el mismo procedimiento con las imágenes de test (Figura 3.6). Estas pueden pertenecer a personas que estaban incluidas entre las imágenes de referencia.

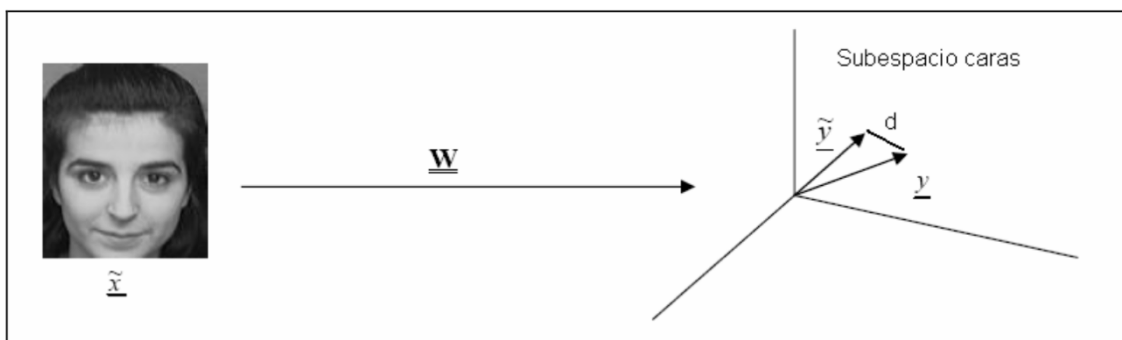


Figura 3.6 - Ejemplo de proyección de una imagen de test.

- 3) Finalmente se dirá que ha habido reconocimiento si se cumple alguna condición, en general se considera que hay un reconocimiento positivo si se cumple una condición:

$$d \Rightarrow |\tilde{y} - y| < d_{th}, \text{ donde } d_{th} \text{ es una distancia threshold. (1)}$$

En concreto para el caso de este estudio, una imagen de test pertenece a otra del conjunto de imágenes de referencia si es la que cumple distancia mínima de manera que siempre se tiene que asignar un sujeto a una imagen de entrada.

A continuación se introducen los análisis más comunes que nos permiten reducir la cantidad de datos de la que disponemos a la hora de analizar imágenes de rostros o de características ya que en muchas ocasiones son redundantes o requieren de una gran cantidad de cómputo, lo que nos facilita su estudio y su procesamiento.

3.2 Reducción de dimensionalidad y extracción de características

3.2.1 PCA (Principal Component Analysis)

PCA es un método que transforma un número de variables posiblemente correladas en un pequeño número de variables incorreladas llamadas componentes principales, es decir, es un algoritmo de reducción dimensional que permite encontrar los vectores que mejor representan la distribución de un grupo de imágenes.

PCA está basado en la Transformada de Karhunen-Loeve (KLT) [11], que consiste en la representación de un proceso estocástico no periódico a través de una base de vectores obtenidos completamente del proceso en sí mismo, es decir, PCA permite representar una imagen de una cara usando una base que se ha conseguido a partir de muchas observaciones de diferentes caras.

El objetivo de este método consiste en representar una imagen en términos de un sistema de coordenadas óptimo reduciendo el número final de componentes que tendrá la imagen. Un ejemplo de PCA lo podemos ver en la Figura 3.7.

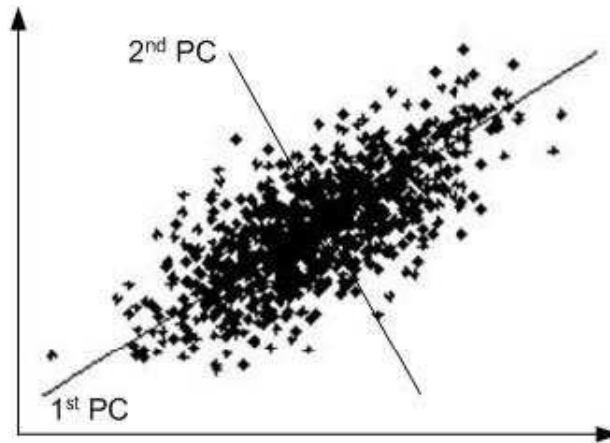


Figura 3.7 - Componentes principales de un conjunto de puntos bidimensional. La primera componente ofrece una reducción lineal óptima de 2D a 1D en cuanto a error cuadrático medio se refiere.

Esta base de transformación depende de las observaciones y se forma a partir de los d autovalores más significativos de la matriz de covarianza, los cuales representan las componentes principales que son más comunes en imágenes de diferentes caras. La reducción dimensional realizada por PCA es equivalente al número de autovectores que se utilicen. Por lo tanto, la imagen proyectada por PCA tendrá una dimensión de valor d , como se puede ver en la Figura 3.8.

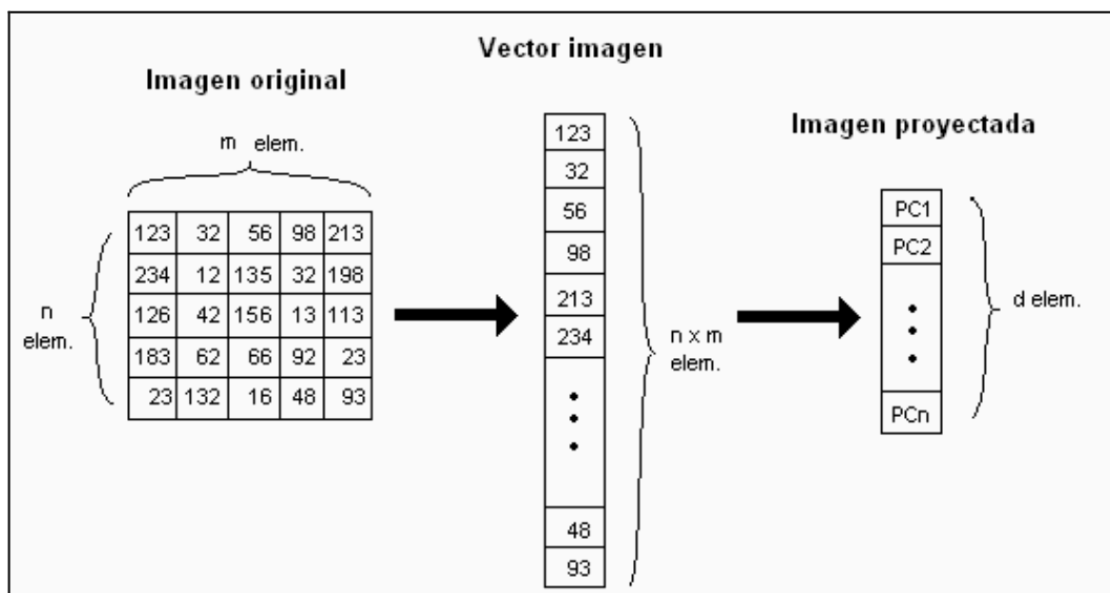


Figura 3.8 - Ejemplo de reducción dimensional al aplicar PCA.

Estos autovectores representan las componentes principales que son más comunes en imágenes de diferentes caras, otra forma de verlo, es que son los ejes del subespacio donde proyectaremos las imágenes de caras humanas los cuales se corresponden con las direcciones de máxima varianza de los datos. La matriz de transformación está formada por los autovectores correspondientes a los d autovalores más significativos.

Al realizar el análisis PCA lo que estamos obteniendo es la siguiente expresión [12]:

$$x = \bar{x} + \Phi_s b_s \quad (1),$$

siendo x , el vector que contiene los puntos de forma originales, \bar{x} la media de todos los puntos de forma, Φ_s la matriz de covarianza con los *eigenvectors* (autovectores) principales y b_s un vector de parámetros característicos de cada individuo que representa los pesos y que usaremos de codificación individual.

Otra de las cualidades de PCA, es que realiza una discriminación entre clases, es decir, todas las caras correspondientes a un mismo individuo están cerca entre sí, mientras que las imágenes correspondientes a individuos diferentes están más alejadas. Esta propiedad es más destacable en las primeras componentes de las imágenes proyectadas.

PCA es una técnica muy recurrida en análisis de datos masivos ya que resulta muy útil su propiedad que nos permite reducir la dimensionalidad de dichos datos. Nos ofrece grandes ventajas de computación reduciendo la carga necesaria, así como otras características de interés. El modelo basado en ejes (Axis Model)[1], a su vez basado en AAM [2], utiliza también PCA en uno de sus pasos, sin embargo de una manera algo distinta a la explicada en este método. Más adelante se detallará en profundidad.

A continuación se describirán brevemente otros métodos de extracción de características y de reducción de dimensionalidad muy utilizados en sistemas de reconocimiento de patrones.

3.2.2 LDA (Linear Discriminant Analysis)

LDA tiene como objetivo convertir un problema de alta dimensionalidad en uno de baja [13][15]. Para ello LDA proyecta los datos (imágenes) en un espacio vectorial de baja dimensionalidad de manera que el ratio entre la distancia entre clases y la distancia dentro de la clase se maximiza. De este modo se garantiza una máxima discriminación entre las clases.

A diferencia de PCA, que es un método de reducción dimensional sin supervisión, LDA sí que es supervisado y utiliza información de los datos.

Se obtiene una matriz de proyección que permite pasar de un problema de alta dimensionalidad a uno de baja. Cada una de las columnas que forman esta matriz serán las bases del nuevo subespacio, las cuales se conocen como *FisherFaces* (Figura 3.9). Estas bases son aquellas que garantizan una máxima discriminación entre clases por encima de ser las que mejor describen los datos, a diferencia de PCA [14].

Una de las limitaciones que tiene LDA, es que se requiere de matrices de dispersión no-singulares. Esto en aplicaciones de reconocimiento facial es complicado debido a que en muchas ocasiones la alta dimensionalidad de las imágenes es mucho mayor

que el número de imágenes lo que desemboca en un problema de matrices singulares, algo que como se verá más adelante también puede suceder en LPP, conocido como *undersampled* o *singularity problem*. Para evitar este efecto, se utilizan diferentes aproximaciones como *pseudo-inverse* LDA [15], que utiliza la pseudo-inversa de la matriz de covarianza en lugar de su inversa, y PCA + LDA [15], que realiza una reducción previa de los datos utilizando PCA.



Figura 3.9 - Las 6 Primeras *FisherFaces* calculadas en la base de datos ORL

3.2.3 LPP (Locality Preserving Projections)

LPP [16] es un algoritmo lineal que del mismo modo que PCA realiza una reducción dimensional de los datos. Al tratarse de un algoritmo lineal es rápido y útil para aplicaciones prácticas.

Una de las propiedades que lo diferencian de PCA es que, en lugar de conservar la estructura global de los datos, conserva la estructura local. De este modo los ‘vecinos’ para un dato en concreto serán los mismos en el espacio original, de alta dimensionalidad, y en el nuevo subespacio de baja dimensionalidad [17]. Al conservarse la estructura local de los datos, las imágenes pertenecientes a un mismo individuo estarán cercanas entre si y alejadas de las de otros individuos, es decir, hay una discriminación entre clases.

Para conservar la estructura local de los datos se hace uso de un grafo de adyacencias que incluye información de la estructura de los datos. Este grafo consiste en la creación de una matriz de tamaño $N \times N$, donde N es el número de imágenes, que tiene asignados unos pesos dependiendo de si los elementos i y j son vecinos o no. En el momento de crear este grafo cabe la opción de crearlo de manera automática (caso no supervisado), utilizando métodos de búsqueda de ‘vecinos’ como *K-nearest neighbors* o *ϵ -neighborhoods*, o hacerlo de manera manual (caso supervisado) asignando manualmente quienes son o no vecinos.

Este método presenta algunos inconvenientes. Uno de ellos es la dificultad de recuperar los datos originales a partir de los datos proyectados al nuevo subespacio, esto se debe al hecho de hacer uso de bases no ortogonales, este problema no ocurre con PCA, por ejemplo.

Otro inconveniente que se puede dar es el de las matrices singulares, el cual se produce cuando la dimensionalidad de los datos es mucho más elevada que el número de muestras ($n \gg N$). Para solucionar este problema se hace uso de técnicas de reducción dimensional de los datos antes de utilizarlos de manera que $n = N$ o $n > N$.

Por otro lado, este método es interesante por diferentes perspectivas:

1. Los mapas están diseñados para minimizar algún criterio objetivo procedente

- de las técnicas lineales clásicas.
2. Como LPP preserva la estructura local de los datos, el sistema tendrá los mismos 'vecinos' en el espacio de baja dimensionalidad que en el de alta.
 3. LPP es un método lineal, lo que hace que sea rápido y conveniente para aplicaciones prácticas.
 4. LPP está definida para todos los casos, a diferencia de las técnicas de reducción no lineales que solo están definidos para el conjunto de datos de entrenamiento, de manera que al tener un nuevo dato LPP es capaz de representarlo en el nuevo subespacio sin tener que recalcularse las matrices de proyección.

3.2.4 DCT (Discrete Cosine Transform)

La DCT es una transformación que representa una secuencia finita de datos como la suma de una serie de funciones cosenoidales oscilando a diferentes frecuencias. Esta técnica es muy utilizada en aplicaciones de procesamiento de señal, desde compresión de audio e imágenes hasta métodos espectrales para la solución numérica de ecuaciones diferenciales. Una de estas aplicaciones es el reconocimiento facial [19].

En este trabajo, la DCT será utilizada para obtener características de la imagen y luego usarlas para poderlas clasificar, como se hace con PCA [18]. A diferencia de PCA, este método no necesita ser entrenado con imágenes del mismo tipo a las que se van a usar, sino que simplemente se transforman directamente las imágenes, es decir, la base de la transformación es independiente de las imágenes. Otra de sus ventajas es su bajo coste computacional en relación con PCA [18].

Se puede tratar este algoritmo del mismo modo que PCA o LPP, es decir, como un sistema que tiene una matriz de proyección (en este caso única para cada imagen) gracias a la cual se puede proyectar las imágenes a un subespacio de baja dimensionalidad de igual modo que se hace en PCA.

3.2.5 DCT por Bloques

Este método realiza una mezcla de las técnicas basadas en la apariencia y las basadas en modelos, en concreto, hace uso de la misma metodología que el método DCT pero aplicado de forma distinta.

Lo que se hace es aplicar DCT-II sobre bloques de 8x8 píxeles de la imagen [20]. Una vez calculada la DCT sobre cada uno de estos bloques de 8x8 píxeles, se selecciona un determinado número de coeficientes por bloque con el fin de crear la nueva representación de la imagen.

Este sistema hace uso parcial de los sistemas basados en modelos porque requiere de la localización de alguna característica facial. En este caso para el buen funcionamiento del sistema se necesita que las imágenes estén recortadas y totalmente alineadas []. Por esta razón previamente a aplicar el algoritmo, se debe detectar la posición de los ojos y alinear la imagen según un criterio. En caso de que no se realice esta alineación previa de las imágenes, el sistema no funcionará correctamente.

Esta introducción a diferentes sistemas de reducción de dimensionalidad han sido expuestos muy resumidamente al no ser este el objetivo del trabajo, la intención es que el lector se familiarice con distintas técnicas de reducción y sea consciente de la importancia de la reducción de datos a la hora de analizar imágenes que nos permitan realizar la identificación de un individuo. El análisis expuesto más detalladamente ha sido el PCA, puesto que es el análisis que implementaremos en nuestro modelo.

3.3 Técnicas de reconocimiento facial actuales

En la actualidad existen varios sistemas de reconocimiento facial que han demostrado su eficacia bajo ciertas condiciones. A continuación, mostraremos algunos de los principales y como progresivamente se han ido mejorando unos a otros hasta llegar a nuestro sistema.

Una de las principales motivaciones a la hora de realizar un sistema de reconocimiento facial es la robustez. Un modelo debe además ser capaz de adquirir parámetros de una imagen que le permitan discernir entre sí es un rostro lo que se está observando o no. Estos parámetros son utilizados a menudo para la identidad de una persona, su pose o incluso su expresión facial. Para poder interpretar una nueva imagen es fundamental un método eficiente para encontrar una coincidencia válida entre una imagen y un ejemplo sobre la imagen que se quiere buscar.

Existen múltiples modelos que permiten sintetizar caras completas, que son los que nos interesan en este estudio.

En primer lugar, destacaremos el modelo Eigenface desarrollado por Turk and Pentland en 1991[7]. Este modelo consiste en realizar un análisis PCA sobre un banco de imágenes de rostros correctamente alineadas. Cada componente PCA representa una imagen llamada *eigenface* (Figura 3.10). Dichas componentes corresponden a las componentes con mayor variabilidad entre sus valores, es decir, la primera componente obtenida tras el análisis PCA (PC1), nos mostrará aquellas características de la imagen que más varían respecto a todos los individuos de la base de datos (por ejemplo el pelo).



Figura 3.10 - Representación a modo de imágenes de cada 12 componentes obtenidas tras un análisis PCA.

Este método parte de un rostro medio obtenido tras la suma de todas las imágenes de rostros alineadas (Figura 3.11) a la cual le añade una combinación lineal de cada componente obtenida tras el análisis PCA. (Figura 3.12).



Figura 3.11 - Imagen media de todos los rostros de la base de entrenamiento



Figura 3.12 - Representación de un rostro a partir de la combinación lineal de eigenfaces

Como se puede observar, a partir de este método podemos codificar cualquier individuo con un simple vector de número que nos indique en qué medida debemos aplicar cada uno de los componentes principales, lo que supone un nivel de compresión y de codificación muy elevado.

Sin embargo, eigenfaces no es robusto frente a cambio de forma en los rostros, así como podrían ser cambios producidos por expresiones faciales y variaciones de pose y orientación [8].

Por ello Ezzat and Poggio [21], en 1996, sintetizaron un nuevo modelo que añadía nuevos puntos de vista sobre un mismo rostro, permitiendo así la identificación del mismo a pesar de ser una foto presentada de diferentes perfiles. Sin embargo, este modelo no respondía adecuadamente a las vistas de rostros que no había observado con anterioridad, lo que limitaba su uso a unas orientaciones determinadas.

C. Nastar y Pentland [22], en 1996, propusieron un nuevo modelo 3D que obtenía información sobre el nivel de grises (*gray-level*) de la superficie de la imagen, lo que permitía un análisis completo de la misma tanto en su forma como en su apariencia, sentando las claves de los actuales modelos más robustos de identificación facial. Sin embargo, este modelo presenta problemas de mínimos locales a la hora de buscar la mejor aproximación, no encontrando en ocasiones la mejor coincidencia posible.

G.J. Edwards, C.J Taylor y T.Cootes propusieron en 1997 el Active Shape Model (ASM)[23] partiendo de la idea de que era necesario obtener información tanto de la forma como de la apariencia de cada rostro. Extendieron la propuesta inicial de C. Nastar modelando también la correlación entre la forma y la apariencia. Partiendo de que la mejor manera de ajustar modelos a imágenes era minimizando el error en la medida entre la apariencia predicha y la imagen requería mucho consumo. Proponen realizar una media de todas las formas de cada cara obteniendo unos puntos característicos de cada imagen. A continuación, se reducen todas las imágenes de manera que sus puntos característicos coincidan con los de la media. Cuando cada rostro ha sido modificado a dicha media, se obtiene entonces su nivel de grises en esa posición.

Este método es realmente efectivo, pero el ASM no aprovecha toda la información disponible y por lo tanto no siempre es robusto. Es por ello que los propios autores del ASM desarrollaron una serie de modificaciones en su propio modelo desarrollando así el Active Appearance Model (AAM)[8].

El AAM es uno de los sistemas más robustos de la actualidad en cuanto a identificación de individuos. El modelo a estudiar en este trabajo se basa precisamente en este último añadiendo modificaciones a la hora de interpretar la información que se obtiene de él. Durante la descripción del proceso, se detallará más en profundidad los pasos necesarios para realizar un buen AAM, además, como se ha comentado en el resumen, se comparará con el primero de los algoritmos, el Eigenfaces, para dar constancia de la mejora de un método con respecto al otro.

Es necesario remarcar que todos los modelos expuestos anteriormente parten de una idea que se conoce como *exemplar model*[24](modelo de ejemplos). Esto quiere decir que nuestro modelo tiene una réplica, o ejemplo, de lo que se quiere identificar ‘en su interior’, por lo que, a la hora de identificarlo, compara los resultados obtenidos con el ejemplo que tiene disponible. Por ejemplo, si el sistema debe reconocer el rostro de Brad Pitt cuando se le enseña un rostro de Brad Pitt, debe tener almacenado un rostro modelo de cómo es en realidad Brad Pitt.

Si asumimos este hecho, debemos asumir que, por lo tanto, en nuestro cerebro tenemos una neurona que identifica específicamente a Brad Pitt y una para cada rostro que hubiéramos memorizado a lo largo de nuestra vida. Dicha neurona se activaría cada vez que obtuviéramos una identificación positiva sobre cierto individuo. No parece razonable que tanto los primates como los seres humanos identifiquemos individuos asignando una neurona por cada uno, por lo que Le Chang y Doris T. Tsao desarrollan el Axis Model [25] analizando la forma análoga en la que lo hacemos.

3.4 Modelo de ejes (Axis Model)

A pesar de que todos los modelos presentados obtienen buenos resultados en lo que a identificación se refiere, hasta la fecha, no parecía prudente afirmar que un sistema de reconocimiento facial fuera igual de fiable que el sistema que, de manera natural,

utilizamos los humanos, y de manera más general, los primates. A los primates no parece afectarles a la hora de reconocer un rostro su orientación. Tampoco parece afectarles en exceso las diferentes condiciones de iluminación. Estos son quizá los dos grandes problemas del actual reconocimiento facial. Persiguiendo el objetivo de reproducir un sistema capaz de replicar el comportamiento de los primates a este respecto, los doctores Le Chang y Doris Y. Tsao, de la Universidad de Caltech, han propuesto recientemente el modelo de ejes (*Axis model*) [25] al que también llaman *linear projection onto a single appearance STA axis*.

El estudio realizado sostiene que podemos obtener una simple codificación de cualquier rostro a partir de los impulsos neuronales recogidos en las neuronas que se encuentran en el middle lateral (ML), middle fundus (MF) y en el anterior medial (AM) [26](Figura 3.13).

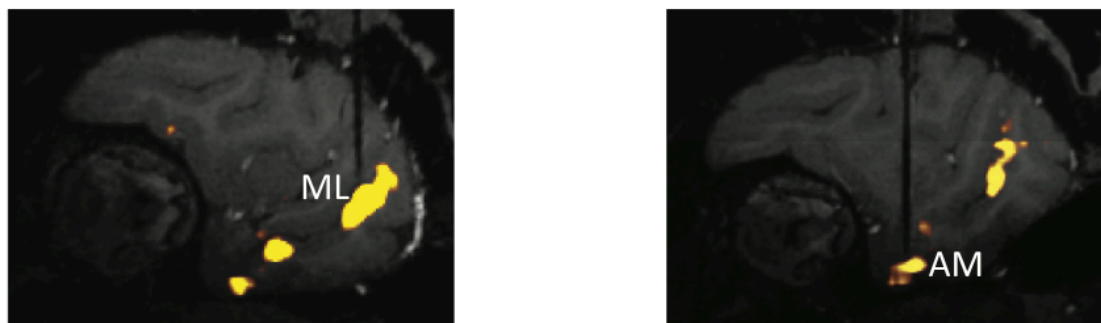


Figura 3.13 - Cortes sagitales mostrando la localización de los grupos de neuronas correspondientes a la zona ML y la zona AM.

Decodificando dichos impulsos, podemos ser capaces de transformarlos a un subespacio cara de manera que cada característica principal de cada rostro esté codificada en un eje específico, independiente y ortogonal al resto de características. Este modelo predice que cada célula debe tener un espacio lineal nulo y ortogonal en el cual todas las caras obtienen la misma respuesta. Uno de los objetivos de este trabajo es conseguir dicho vector de codificación que seamos capaces de interpretar como una respuesta neuronal [27].

Para la realización del experimento, fue necesario entrenar un modelo que nos permita generar, reducir y representar, cualquier rostro en un vector de características especiales. Para ello, como se ha comentado con anterioridad, se ha optado por el AAM, ya que cumple con los requisitos que estamos buscando.

3.5 Active Appearance Model

Los Modelos de Apariencia Activa (AAM) [8] permiten reproducir de forma sintética imágenes de superficies que incluyen deformaciones no rígidas y cambios de apariencia. Están basados en la obtención, mediante una fase de entrenamiento, de un modelo estadístico de la forma y la apariencia del objeto de interés.

En un AAM la forma es descrita mediante un conjunto de N puntos característicos (los llamados Parámetros de forma), que determinan una malla similar a la representada

en la Figura 3.14.

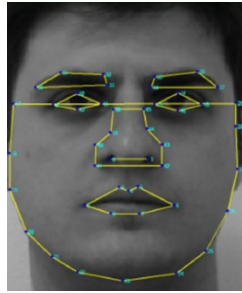


Figura 3.14 - Ejemplo marcado de puntos en AAM

Mediante el análisis de componentes principales (PCA) sobre los puntos de entrenamiento se obtiene un vector medio s_0 y un subespacio $B_s = [s_1, \dots, s_n]$ formado con n componentes principales, con una dimensionalidad menor que la del conjunto de entrenamiento. La expresión obtenida es la siguiente:

$$x = \bar{x} + \Phi_s b_s \quad (2)$$

Siendo x , el vector que contiene los puntos de forma originales, \bar{x} la media de todos los puntos de forma, Φ_s la matriz de covarianza con los eigenvectores (autovectores) principales y b_s un vector shape de parámetros característicos de cada individuo que representa los pesos.

Cualquier instancia de la forma del modelo se obtiene a partir de una combinación lineal de los vectores de la base de forma, B_s , mediante la siguiente expresión:

$$s(p) = s_0 + \sum_{i=1}^n p_i s_i \quad (3)$$

La inclusión de los parámetros de forma en el modelo se realiza mediante una transformación afín definida a trozos denominada función warp $W(x;p)$ [28]. Esta función se encarga de transformar los puntos interiores de un vector concreto (normalmente s_0) en donde se define la apariencia, a cualquier vector $s(p)$ generado a partir de (3). Es decir:

$$x' = W(x;p) \quad (4),$$

Donde x son puntos en el interior de s_0 y x' está definido en el interior de $s(p)$.

La apariencia se describe a partir del mapa de bits definido en la imagen deformada hacia los puntos del vector s_0 , son los llamados parámetros de apariencia.

Mediante (4) se transforman las imágenes de entrenamiento, con el fin de normalizarlas en forma. De la misma manera que con los parámetros de forma,

mediante PCA se obtienen tanto la apariencia media A_0 , como la base de un subespacio $B_A = [A_1(x), A_2(x), \dots, A_m(x)]$, de dimensión menor al conjunto de entrenamiento, que está formada por las m componentes principales del entrenamiento, obteniendo:

$$g = \bar{g} + \Phi_g b_g \quad (5),$$

siendo g , el vector que contiene la imagen en escala de grises original, \bar{g} la media de todas las imágenes, Φ_g la matriz de covarianza con los eigenvectores (autovectores) y b_g , un vector shape-free de parámetros característicos de cada individuo que representa los pesos.

A partir de estos elementos, se obtiene un modelo de apariencia lineal, que es capaz de generar una instancia de apariencia a partir de una combinación lineal de la media y las componentes de la base, ponderadas por un conjunto de parámetros $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)$:

$$A(x; \lambda) = A_0(x) + \sum_{i=1}^m \lambda_i A_i(x) \quad (6)$$

Estos parámetros serán los utilizados para codificar cada rostro de cada individuo y serán sus proyecciones sobre el espacio-cara generado, las que nos permitan identificarlos.

El modelo AAM es un buen método estadístico para ajustes de plantillas, el cual utiliza toda la información disponible de la cara, es decir, no solo los puntos de los bordes que representan la forma de la cara, sino también los puntos interiores que definen la apariencia. El problema de este algoritmo es el elevado coste computacional a la hora de deformar imágenes para ajustarlas al modelo [12].

En el sistema tradicional de AAM se sugiere en muchas ocasiones concatenar ambos bancos de datos ya analizados y reanalizar con un PCA ambos de nuevo, esta vez juntos. En este estudio no se realiza de esa manera, puesto que no es necesario y ambas matrices son de tamaños distintos. Se mantiene el análisis por separado únicamente.

Una vez entendido que hace exactamente el análisis de componentes principales, se hace patente la necesidad de calcular el término b_s y b_g de ambas ecuaciones. Dicho vector es el que nos dará la identidad numérica única de cada individuo que estamos buscando. Para ello, basta con realizar el proceso inverso:

$$b_s = \Phi_s^T (x - \bar{x}) \quad (7)$$

El vector b_s define un set de parámetros de deformación que identifica una forma determinada de cada individuo. Cada vez que cambiemos un elemento de este vector, estaremos cambiando en última instancia una característica referente a la forma de su rostro. El mismo proceso inverso debe realizarse para obtener el vector b_g con

características shape-free.

Normalmente, cuando se realiza un modelo AAM, se utilizan todas, o casi todas, las componentes principales que nos arroja el análisis PCA, puesto que la reconstrucción del rostro será más fiable. Sin embargo, el 95% del total de la variación entre rostros puede obtenerse únicamente con las primeras 5 componentes, como puede verse en la Figura 3.15. Esto sugiere que no será necesario utilizarlas todas para una correcta identificación. Por ello, en el estudio realizaremos experimentos para comprobar hasta qué punto podemos reducir esas componentes y, con ello, el vector codificación.

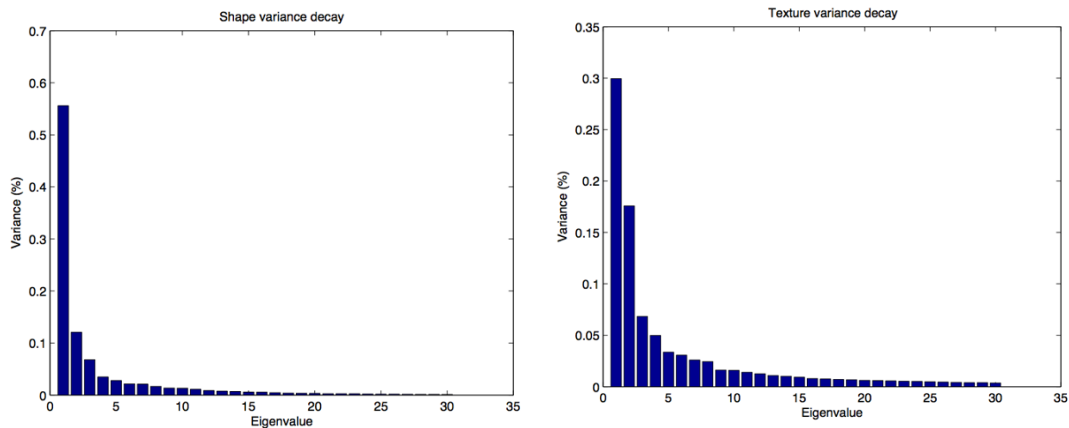


Figura 3.15 - Variación de cada PC para forma y apariencia respectivamente.

A la hora de hacer al sistema robusto frente a cambios en la orientación, sería necesario repetir el proceso, como se ha comentado, con cada una de las orientaciones. Una vez realizado, podemos utilizar *linear regression* (regresión lineal) para establecer una relación entre cada perfil con cada frontal siguiendo la siguiente expresión [25]:

$$\begin{pmatrix} p_i^1 \\ \dots \\ p_i^{200} \end{pmatrix} = \begin{pmatrix} 1 & f_1^1 & \dots & f_n^1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & f_1^{200} & \dots & f_n^{200} \end{pmatrix} \begin{pmatrix} \beta_i^0 \\ \dots \\ \beta_i^n \end{pmatrix} + \begin{pmatrix} \varepsilon_i^1 \\ \dots \\ \varepsilon_i^{200} \end{pmatrix} \quad (8)$$

Donde p representa el vector del rostro de perfil, f la matriz de coordenadas frontales, β , los coeficientes de regresión y ε el error cometido.

Se realizaría este proceso para las características de forma y apariencia independientemente. Una vez realizado, podríamos obtener las coordenadas frontales de un individuo a partir de un perfil y viceversa, obteniendo en ambos casos las mismas características que le definen y protegiéndonos de los cambios producidos por la orientación del rostro.

4. Metodología

4.1 Base de datos

Para la realización de este estudio se utilizó la base de datos propuesta por Le Chang y Doris Y. Tsao [25], autores del estudio que motivaron este trabajo, la FEI Face Database [29], una base de datos realizada por el Artificial Intelligence Laboratory of FEI de Sao Paulo, Brasil. Esta base de datos es de uso libre para proyectos de investigación y es a partir de la cual entrenamos nuestros modelos de reconocimiento.

La base de datos contiene 200 individuos, hombres y mujeres de diferentes edades, con diferentes orientaciones. Lo cual es necesario para entrenar el modelo y hacerlo robusto a cambios de orientación. Además, dispone también de una imagen de cada individuo sonriendo, lo cual es necesario para entrenar el modelo y hacerlo robusto frente a cambios en las expresiones faciales. En la Figura 4.1 se puede ver una muestra de ejemplo.



Figura 4.1- Muestra de individuo contiene FEI Face Database.

La base de datos contenía imágenes de 260x360 a color. Dichas imágenes no se encontraban perfectamente alineadas, por lo que fue necesario alinearlas para mejorar los resultados de entrenamiento del modelo. Además, se convirtieron las imágenes a escala de grises para manejar un número más reducido de datos y mejorar los tiempos de cómputo, ya que no es relevante el color a la hora de identificar individuos (Figura 4.2).



Figura 4.2- Rostro centrado y convertido a escala de grises

4.2 Software utilizado

Para el desarrollo del trabajo se utilizó Matlab en su versión de estudiante. Matlab es un lenguaje interpretado, sencillo de manejar y rápido de implementar, lo que lo convertía en un software ideal para un trabajo de estas características. Sin embargo, es un lenguaje lento, lo que deberá tenerse en cuenta en la toma de tiempos. A pesar de ello, se prefirió debido a su facilidad para obtener gráficas ilustrativas y a funciones ya implementadas que podemos encontrar en sus *toolboxes*.

Se utilizaron para este trabajo, el *Statistics and Machine Learning Toolbox* [31], con funciones estadísticas, el *Image Processing Toolbox*[30], con funciones de tratamiento y manejo de imágenes, el *Symbolic Math Toolbox* [32], con funciones de cálculo de autovalores y autovectores, y el *Neural Network Toolbox* [33], que proporciona funciones como distancias euclídeas necesarias en la identificación.

4.3 Desarrollo algoritmo Eigenfaces

Una vez adecuadas las imágenes para su procesamiento, se comenzó con el desarrollo del primero de los dos algoritmos, el Eigenfaces.

El algoritmo se compone de tres fases claramente diferenciadas. La primera es la creación de la base de datos y la extracción de características de la misma. La segunda fase es la obtención de la imagen test que se pretende identificar y su proyección en nuestro espacio de características obtenido en la primera fase. La última fase es la fase de decisión. En la misma, se proyectan todas las imágenes de la base de datos en el espacio obtenido en la primera fase del mismo modo que hemos hecho con la imagen test y calculamos la distancia euclídea obtenida de cada imagen con la proyección de test obtenida en la fase dos. La mínima distancia obtenida será la que defina la identidad del individuo. La Figura 4.3 muestra un diagrama donde se observan las tres fases del proceso.

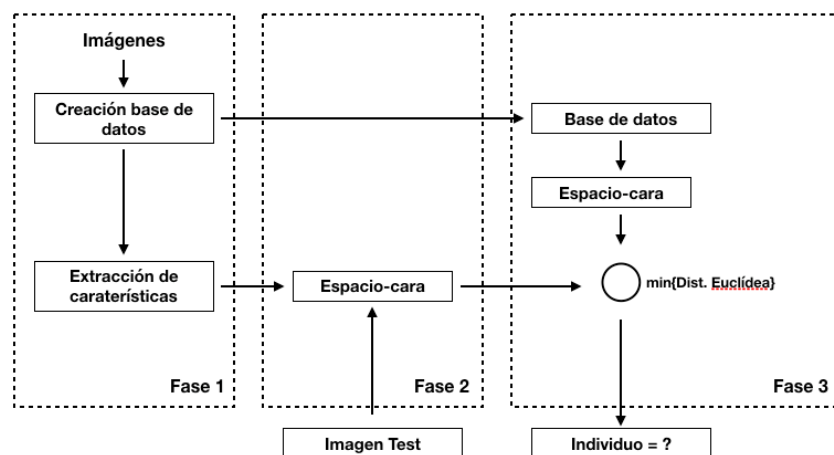


Figura 4.3- Diagrama de funcionamiento del algoritmo Eigenfaces.

4.3.1 Creación del face-space

Para la fase primera, se implementó una función en Matlab que leyera de nuestra base de datos todas las imágenes y las almacenara en una misma matriz de 260x360. Después, proseguimos realizando un análisis PCA para extraer las características más relevantes del banco de datos. Para ello, se calculó la media de todos los individuos, sumando todas las imágenes de la base de datos y dividiendo por el número total de imágenes. Seguidamente, a cada imagen, le sustrajimos la media calculada y creamos otra matriz A que contenía todas las caras a las que le habíamos restado su media. La Figura 4.4 muestra el resultado de sustraer la media a un individuo.



Figura 4.4 – Imagen de un rostro al que se le ha sustraído la media de todos los rostros.

A continuación, se calculó la matriz de covarianza multiplicando la matriz A obtenida por su inversa. A partir de esta matriz de covarianza fueron calculados los autovalores (*eigenvalues*), que se correspondían a los elementos de la diagonal. Obtuvimos dichos valores y los agrupamos eliminando aquellos que tuvieran un valor inferior a uno. Una vez fue realizado este paso, calculamos los autovectores multiplicando la matriz de imágenes a la que le fue sustraída su media, la matriz A, por los autovalores calculados, obteniendo así el espacio formado por los autovectores, también conocido como *eigenvectors* en inglés, o *eigenfaces*, dando nombre al algoritmo. Estos *eigenfaces* agrupados conformaron nuestro espacio-cara (*facespace*) y tenía un tamaño de 93600x199. 93600 se obtuvo de concatenar las imágenes de 260x360 y 199 correspondía al tamaño del banco de imágenes menos uno. Por lo tanto, se pudo reducir la dimensionalidad del banco de datos en un orden de magnitud. Esto fue debido a que teníamos muchas más variables (93600 correspondientes a cada píxel de la imagen) que muestras (200), lo que nos daba una reducción de dimensionalidad baja. Sin embargo, no era nuestro objetivo reducirla aquí.

4.3.2 Proyección imagen de test

Una vez se definió nuestro *facespace* compuesto por nuestros *eigenfaces*. El siguiente paso fue proyectar una imagen de test en el mismo espacio generado. Cualquier imagen de 260x360 podía ser proyectada a este espacio y obtener así un vector proyección, el vector de características, de 199 elementos. Sin embargo, el modelo no identificaría ninguna imagen al no disponer almacenada la proyección de la imagen

nueva. Por lo tanto, la imagen de test debía de ser una imagen perteneciente a la base de datos utilizada.

La fase dos requería obtener una imagen y proyectarla sobre el *facespace* obtenido. Para ello, bastaba con multiplicar la cara que queríamos identificar, concatenada en un vector de dimensión 93600×1 , frente a nuestro espacio de 93600×199 traspuesto. Obtuvimos de esta manera un nuevo vector proyección de 199×1 que contenía la codificación para esa cara y ese espacio.

4.3.3 Identificación

La fase tres es la fase de identificación, para ello, proyectamos todas las imágenes originales de nuestra base de datos sobre el espacio-cara calculado obteniendo así todas las codificaciones de cada una. Por último, calculamos la distancia euclídea entre cada elemento del vector con cada elemento de cada cara original y calculamos la mínima obtenida en todo el banco de imágenes. La mínima obtenida se corresponderá con la imagen que el algoritmo consideraba más próxima. Para corroborar que la mínima distancia euclídea obtenida es la correspondiente al rostro que queremos identificar, se han ordenado las imágenes de manera que coincidan sus índices, así, con una simple comparación de los índices arrojados por el programa, contabiliza automáticamente cuántas veces la identificación es satisfactoria.

4.4 Desarrollo algoritmo Active Appearance Model

Una vez fue terminado el desarrollo del algoritmo Eigenfaces se procedió al desarrollo de su evolución, el algoritmo Active Appearance Model. Este modelo, presenta mejoras con respecto al anterior ya que recoge más información sobre las imágenes de entrenamiento, lo que requirió un proceso de entrenamiento más laborioso y largo.

4.4.1 Etiquetado del banco de imágenes

Como se ha comentado en la introducción teórica, el AAM tiene en cuenta en su modelo tanto la forma del rostro del individuo como su apariencia. Para ello, el primer paso en su desarrollo fue etiquetar cada una de las imágenes que conforman nuestra base de datos. Este etiquetado consistió en un marcado a mano de todos los puntos que fueron considerados relevantes a la hora de definir una cara. La Figura 4.5 muestra un ejemplo de dicho etiquetado.



Figura 4.5- Ejemplo de etiquetado de dos individuos de la FEI Face Database.

Para este proceso fue creado un GUI (Grafic User Interface) en Matlab para poder realizarlo de la manera más automática posible. Fueron etiquetadas con 48 puntos cada una de las 200 caras frontales que conforman nuestra base de datos. Esta información fue almacenada en forma de vector concatenado, es decir, vectores de 96×1 , que contenían la información referente a la forma de la cara de cada individuo.

Tras finalizar el etiquetado, se calculó una media de todos los puntos de la cara obteniendo así un rostro medio de todos los individuos de la base de datos. Este rostro medio fue utilizado para ajustar cada imagen frontal de cada individuo a un rostro medio común, de manera que podamos obtener información relativa a la apariencia de cada individuo sin que afecte a la apariencia los cambios en la forma del rostro del individuo.

4.4.2 Deformación del banco de imágenes

Para ajustar cada cara a la cara media se utilizó la función *Warp* del *Image Processing Toolbox* de Matlab que deforma suavemente las imágenes según unos puntos de referencia. El algoritmo utilizado para el cálculo de dichas deformaciones se hizo mediante *local weighted mean* ajustando con un polinomio de grado 35, que fue el que mejor ajustaba la imagen en este caso, ya que aplicando grados inferiores las deformaciones no eran suaves y corríamos el riesgo de desvirtuar la apariencia. En la Figura 4.6 puede observarse el efecto de la deformación ante polinomios de grados bajos. En la Figura 4.7, se muestra el óptimo resultado de la mencionada transformación.



Figura 4.6 - Deformación del rostro aplicando 'lwm' de grado 6 y 12 respectivamente. Como se puede observar, no puede considerarse que la deformación sea suave con grados de polinomios tan bajos.

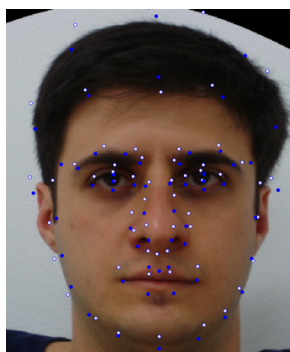


Figura 4.7- Figura que muestra el proceso final de deformación y ajuste a la cara media de un individuo. En blanco, los puntos de la forma original de su rostro. En azul, los puntos del rostro medio hacia los que hemos deformado la cara. Se muestra en color para una mejor apreciación.

El proceso de deformación fue un proceso costoso computacionalmente hablando. Para poder realizarlo con éxito, el ordenador debió calcular el marco de referencia al cual ajustarse buscando la diferencia entre los puntos del rostro original y el rostro medio. Una vez calculado dicho marco de referencia en forma de vector, se le pasó como parámetro a la función de deformado junto con la imagen que se deseaba ajustar. El proceso se realizó para cada una de las 200 caras. Una vez terminado, se almacenaron todas estas nuevas caras deformadas (*warped faces*) redimensionadas a 140x140, para aligerar las operaciones con ellas, en una nueva matriz. Esta matriz era el descriptor de apariencia que estábamos buscando. Donde no influye la forma específica de la cara de cada individuo.

De igual manera a cuando obtuvimos todos los descriptores de forma de la cara anteriormente, calculamos la imagen media a partir de la suma de todas las imágenes deformadas. Esta media fue utilizada más adelante para la correcta reconstrucción de las imágenes y para la identificación. En la Figura 4.8 podemos observar la cara media de los individuos de nuestra base de datos. Llegados a este punto, teníamos ya preparada nuestra base de datos para entrenar un modelo frontal de AAM, la parte más costosa del trabajo por su manualidad. Cabe destacar, que, si se hubiera deseado realizar dicho modelo robusto a variaciones en las expresiones faciales o a la orientación de las caras, debió haberse etiquetado también cada una de las caras correspondientes, sin embargo, la carga de trabajo que eso conlleva sobrepasaba los límites de un TFG. Por lo tanto, se compararon ambos algoritmos únicamente con imágenes frontales.



Figura 4.8 – Imagen media de los 200 individuos con sus caras deformadas a la forma media.

4.4.3 Análisis PCA

A continuación, se procedió a la extracción de características para conformar nuestro nuevo *facespace*. Para ello, seguimos el mismo procedimiento que para el algoritmo Eigenfaces y realizamos un análisis PCA del banco de imágenes deformadas. Esta vez, se optó por utilizar la función PCA que implementa el *Statistics and Machine Learning Toolbox* de Matlab. Esta función recibe una matriz y nos devuelve los autovectores y autovalores calculados y ordenados automáticamente. La función de autovectores tiene un tamaño de 19600x199, por el mismo motivo descrito en Eigenfaces, pero además nos devolvió las proyecciones de dichas imágenes automáticamente. Esta matriz de proyecciones tenía un tamaño de 200x199 como cabía esperar. 200 rostros y 199 valores de codificación. Además, se realizó también el análisis PCA de los vectores de la forma del rostro, obteniendo una matriz de autovectores que codifican la forma del individuo de 200x91.

4.4.4 Reconstrucción a partir del vector de codificación

De nuevo, se obtuvo un vector 199-d que codificaba individualmente cada cara. Para comprobar que habíamos realizado correctamente el proceso, se deshizo el camino inversamente. Partiendo del vector de características, se proyectó dicho vector sobre nuestro espacio *shape-free* (espacio apariencia, que no se ve afectado por la forma de la cara) y obtuvimos de nuevo la cara deformada. Realizamos el mismo proceso con los puntos de la forma de cada individuo y los aplicamos a la función warp de manera inversa. En la anterior ocasión el objetivo era partir de los puntos originales hacia los puntos medios, y en esta ocasión partíamos de la cara media y debíamos devolver la imagen a su estado original.

4.4.5 Identificación

Del mismo modo que con Eigenfaces, el método de decisión sobre la identidad de un individuo se basó midiendo distancias euclídeas. En esta ocasión, seleccionamos una imagen test perteneciente a nuestra base de datos y la deformamos para adecuarla a nuestra base de datos. Tras la deformación, se proyectó frente al *facespace* generado y

se obtuvo un vector de codificación. A continuación, para realizar una correcta comparación con el algoritmo Eigenfaces se volvieron a proyectar todas las imágenes deformadas de la base de datos frente al mismo *facespace* y obtuvimos sus vectores de codificación. Por último, calculamos la distancia euclídea entre el vector test y los vectores de cada cara y seleccionamos la menor de todas ellas, obteniendo así el rostro que más se aproxima al de test en nuestra base de datos. Para corroborar que la mínima distancia euclídea obtenida es la correspondiente al rostro que queremos identificar, se han ordenado las imágenes de manera que coincidan sus índices, así, con una simple comparación de los índices arrojados por el programa, contabiliza automáticamente cuántas veces la identificación es satisfactoria.

4.5 Mediciones de tiempos

A la hora de evaluar un algoritmo no solo es necesario corroborar su correcto funcionamiento, sino también su eficiencia a la hora de realizar su tarea. Para ello se midieron tiempos de las funciones más relevantes de cada algoritmo. Los tiempos que se obtuvieron más significativos fueron los de tiempo de arranque del sistema e identificación del primer individuo y el tiempo que tardaba cada algoritmo en identificar a un individuo una vez el sistema ya se encontraba arrancado. El resultado que esperábamos obtener era el de un mayor tiempo de cómputo para el AAM, ya que requiero de procesos de deformación mucho más costosos, mientras que el Eigenfaces realiza la identificación comparando las imágenes directamente.

4.6 Reducción de la codificación

Para confirmar que podíamos seguir reconociendo caras a pesar de reducir la codificación, como sigue el estudio de Le Chang y Dosis Y. Tsao[25], hemos repetido el proceso de identificación reduciendo el número de componentes utilizados en el vector de características. Realizamos iteraciones proyectando la imagen de test desde sólo dos componentes, hasta que volviera a obtener el algoritmo una tasa de acierto similar a la obtenida con todos los componentes del vector. Este proceso resultó ser un proceso lento y costoso ya que significaba repetir la fase de identificación al menos las 199 veces, una por cada característica que sustraemos, correspondientes a cada elemento del vector.

Para ello, se proyectó la imagen test con un número determinado de elementos menos, dependiendo de la iteración. Esta proyección sobre un espacio reducido proporcionalmente, nos devolvía una imagen reconstruida correspondiente a la que obtendríamos utilizando ese número de componentes. Una vez obtenida dicha imagen, era proyectada nuevamente sobre el *facespace* original para obtener su vector de características en el espacio cara original. Esto nos devolvía un nuevo vector de características de 199 elementos producidos por una imagen que había sido generada con menos componentes. La Figura 4.9 muestra un diagrama del proceso realizado y la Figura 4.10 muestra un ejemplo de caras reconstruidas con menos componentes.

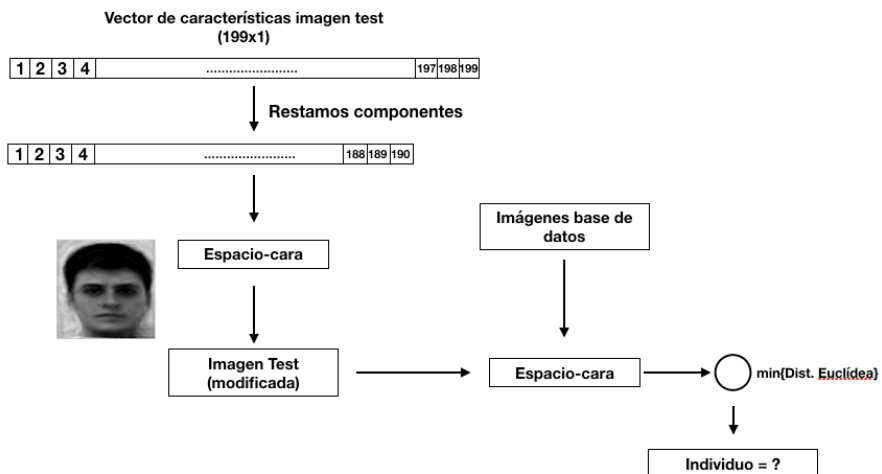


Figura 4.9 – Diagrama del proceso de eliminación de componentes e identificación posterior.

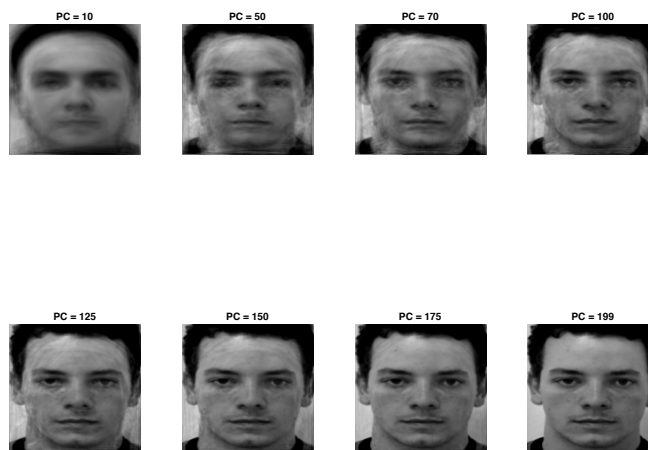


Figura 4.10 – Imágenes reconstruidas eliminando componentes. Sobre cada imagen se muestra el número de componentes utilizados en la reconstrucción.

Tras realizar la nueva proyección debíamos comparar, de la misma manera que antes, la distancia euclidía entre la cara de test frente al resto. Iterando este proceso para cada número de componentes entre 2 y 199, obtuvimos el porcentaje de acierto para cada iteración. Se puso como cota un 80% de acierto en la identificación, a partir de la cual podíamos afirmar que nuestro algoritmo seguía siendo fiable a pesar de la reducción en las componentes de su vector de características.

4.7 Identificación bajo condiciones de ruido y luminosidad adversas

Para poner a prueba la robustez de nuestros algoritmos, se realizaron pruebas sometiendo a la imagen de test a condiciones adversas ya que no podíamos suponer que siempre obtendremos una óptima calidad en la imagen a comparar.

4.7.1 Luminosidad adversa

Se sometió a la imagen de test a problemas de luminosidad. Para ello se siguió un procedimiento similar al descrito en el apartado anterior. Se modificó la imagen de test aumentando el brillo de la imagen por iteraciones, aumentándolo progresivamente. Esta imagen test modificada se proyectó sobre el facespace y se comprobó con las proyecciones de los rostros originales. En la figura 4.11 se puede ver un ejemplo de la imagen test con la luminosidad aumentada.



Figura 4.11 – Rostro de ejemplo en condiciones de iluminación adversas

4.7.2 Imagen test bajo condiciones de ruido

Con el objetivo de evaluar la eficacia de los métodos bajo estudio frente al ruido, se realizaron diversas pruebas de iluminación, se empeoró la imagen test con ruido blanco gaussiano para estudiar como decrece el nivel de acierto conforme aumentamos el ruido. Para introducir ruido, se utilizó la función *imnoise* de tipo *gaussian* disponible en el *Image Processing Toolbox* de Matlab. El ruido se introdujo con una media 0 e iterando la desviación típica del mismo en 0,1 por iteración.

Del mismo modo que en procesos anteriores, se tomó la imagen test modificada y se proyectó sobre el espacio *facespace* para comparar dicha imagen test con el resto de imágenes proyectadas sobre el espacio. En la Figura 4.13 puede observarse un ejemplo de imagen test con ruido gaussiano.



Figura 4.12 - Ejemplo de ruido blanco gaussiano añadido a una imagen test. Izquierda, imagen original en escala de grises. Derecha, imagen con ruido blanco gaussiano des. tip. = 0.1

5.Resultados

En el presente capítulo se exponen los resultados obtenidos a partir de la metodología desarrollada para la implementación del algoritmo Eigenfaces y AAM. Los resultados son obtenidos al probar el sistema para la base de datos descrita en la sección xx.

5.1 Reconstrucción facial en algoritmo AAM

A partir de los métodos descritos en la sección 4, hemos sido capaces de reconstruir cualquier rostro de nuestra base de datos a partir de su vector característico proyectado en nuestro 'espacio-cara'. Se pudo demostrar así que éramos capaces de obtener una codificación de la identidad de cada individuo facilitando así su almacenamiento en bases de datos. En las Figuras 5.1 y 5.2, podemos observar un ejemplo de reconstrucción utilizando el algoritmo AAM.

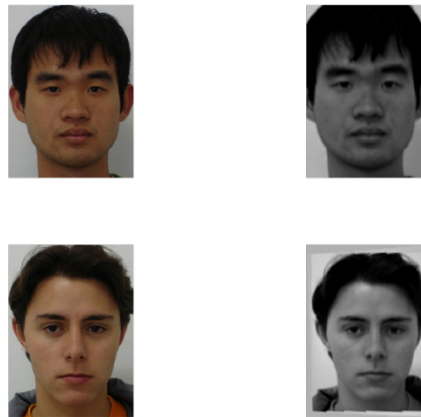


Figura 5.1 – Ejemplo de la reconstrucción de dos rostros de la base de datos. En primer lugar se decodifica su apariencia y seguidamente, se devuelve el rostro a su forma original.

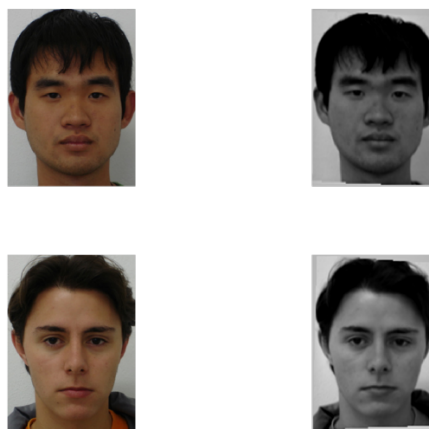


Figura 5.2- Reconstrucción de los mismos rostros devolviéndolos a su forma original.

Como puede observarse, la reconstrucción de los rostros se ha realizado de manera exitosa, confirmando así la validez de la codificación para cada individuo.

El número de elementos del vector de codificación es igual al número de caras disponibles en la base de datos, y a partir de las cuales obtenemos nuestro espacio cara, menos uno. Es decir, al disponer de 200 caras con las que hemos construido nuestro modelo, cada individuo estará codificado por 199 números característicos.

Sin embargo, se ha podido reducir este número de componentes ya que para la identificación facial no es necesario una reconstrucción perfecta del rostro. En la Figura 5.3, podemos observar el efecto de sustraer dichas componentes al vector de codificación 199-d, pudiendo reducir así su extensión. En siguientes apartados comprobaremos hasta qué punto el sistema es capaz de seguir reconociendo caras.

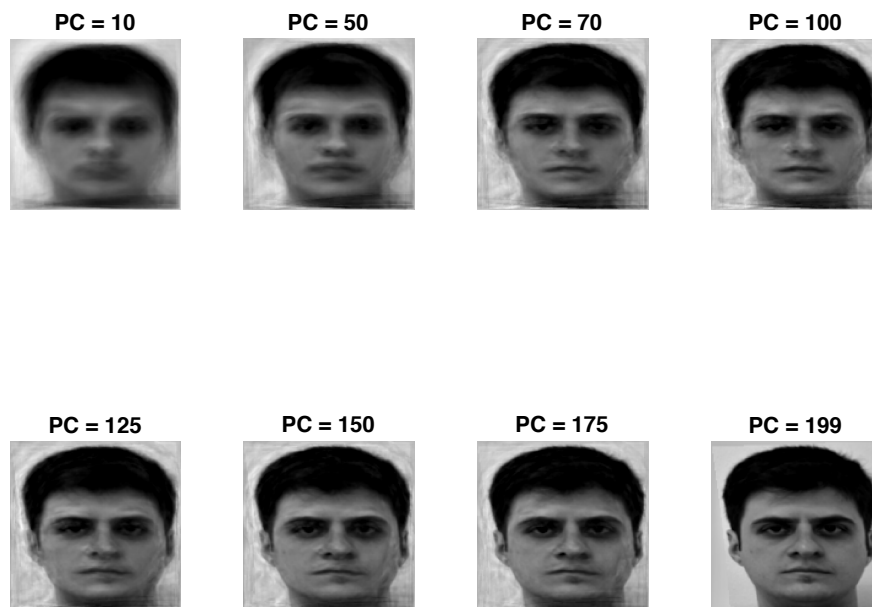


Figura 5.3- Efecto de reconstruir el rostro de un individuo con distinto número de componentes. Como se puede observar, no es necesario todo el vector para poder reconstruir un rostro que un humano pueda reconocer.

5.2 Identificación facial

Se han evaluado por separado ambos algoritmos para determinar su precisión a la hora de identificar individuos dentro de nuestra base de datos. Los resultados obtenidos bajo condiciones normales, es decir, utilizando las imágenes tal y como están almacenadas en nuestra base de datos, se muestran en la Tabla 5.1.

	Porcentaje de acierto	Error
AAM	99.5%	0,5%
Eigenfaces	87.94%	12.06%

Tabla 5.1 – Resultados obtenidos en la identificación positiva de individuos en imágenes sin modificar.

Como se puede observar, los resultados son satisfactorios para condiciones normales en ambos algoritmos, aunque se aprecia una mayor precisión en el algoritmo AAM como era de esperar, pues realiza la comparación sin que afecte a la misma la forma de cada cara, lo que le convierte en un algoritmo más robusto en cuanto a identificación. Al evaluar los rostros de los individuos igualándolos a una cara media, sus variaciones en cuanto forma no son significativas en el análisis, otorgando una mejora de más de un 11%.

Sin embargo, esta ganancia en cuanto a fiabilidad en la identificación tiene un coste de computación que repercute en el tiempo que tarda el algoritmo en resolver una identidad. En la tabla 5.2 podemos observar el tiempo de identificación obtenido para un solo individuo y el tiempo por cada individuo extra. En esta ocasión el algoritmo Eigenfaces es significativamente más rápido pues el AAM requiere de una fase previa de deformado de imágenes para su posterior análisis. Este proceso es un proceso costoso en términos de computación y tiempo.

	AAM	Eigenfaces
Tiempo de arranque e identificación	19.247 s.	3.5877 s.
Tiempo por cada individuo extra	0,42 s/individuo	0.02 s/individuo

Tabla 5.2 – Tiempo de identificación para algoritmo AAM y Eigenfaces

Según los datos mostrados en la tabla, Eigenfaces es un 18.64% más rápido cuando se trata de identificar a un solo individuo, pero también lo es una vez toda las imágenes de la base de datos han sido deformadas. Esto es debido a que AAM también debe deformar el rostro de entrada para compararlo con el resto de imágenes deformadas. Por cada individuo extra, el algoritmo Eigenfaces es un 4.7% por rápido, que para bases de datos con un número elevado de individuos puede resultar muy significativo. En la Tabla 5.3 se muestra el tiempo total que han necesitado ambos algoritmos para realizar el cómputo y la identificación de toda la base de datos de 200 caras.

Esto nos da una muestra de lo importante que puede ser sacrificar un cierto margen de error en la identificación a cambio de ganar velocidad en la detección, sobre todo para bases de datos extensas. Sin embargo, ambos algoritmos son susceptibles de paralelización, lo que disminuiría considerablemente los costes de transformación de caras que penalizan al algoritmo AAM.

5.3 Resultados eliminando componentes

Además de una identificación con todos los componentes disponibles, se ha procedido a la realización de una prueba de identificación sustrayendo componentes de identificación de cada individuo. Así, hemos sido capaces de comprobar hasta qué punto somos capaces de reducir el vector de identificación observando hasta qué punto el algoritmo puede seguir identificándolos con un margen aceptable.

El procedimiento es el que se ha descrito en el apartado 4 del presente trabajo y el efecto producido en las caras es el que puede observarse en la Figura 5.3 mostrada anteriormente. La Figura 5.4 ha mostrado como el nivel de acierto va disminuyendo conforme restamos componentes, como era de esperar. Se ha considerado también que el algoritmo reconoce adecuadamente a los individuos cuando su nivel de acierto es superior al 80%.

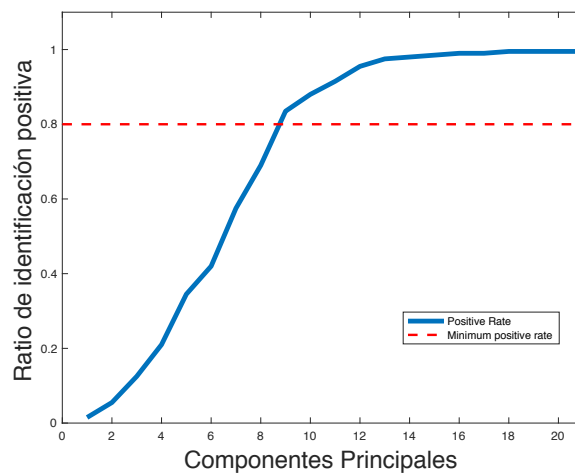


Figura 5.4- Gráfica que muestra como varía el porcentaje de acierto en la identificación de un individuo. En el eje x, el número de componentes utilizadas en la identificación. En el eje y, el ratio de acierto conseguido.

Por lo tanto, tras esta prueba, hemos sido capaces de confirmar que reduciendo los coeficientes a 9-10, seguimos siendo capaces de reconocer individuos. Demostrando así que no todos los componentes son estrictamente necesarios y que contienen información sin relevancia para el algoritmo, reduciendo así la codificación del rostro en más de un 95%. Este resultado puede parecer sorprendente pero tiene su justificación en que son las primeras componentes de nuestro vector las que almacenan la información más variable entre los diferentes rostros, es decir, aquellos rasgos que más distinguen unos individuos de otros son los almacenados en las componentes 1 a 10.

A continuación, se muestra el mismo proceso para el algoritmo Eigenfaces en la Figura 5.5. De manera similar, se ha observado que el nivel de acierto disminuye conforme restamos componentes a al vector cara. Al tratarse de un algoritmo menos robusto en la detección, la reducción obtenida es menor que la conseguida en el AAM, el algoritmo necesita más componentes principales para poder realizar una óptima identificación.

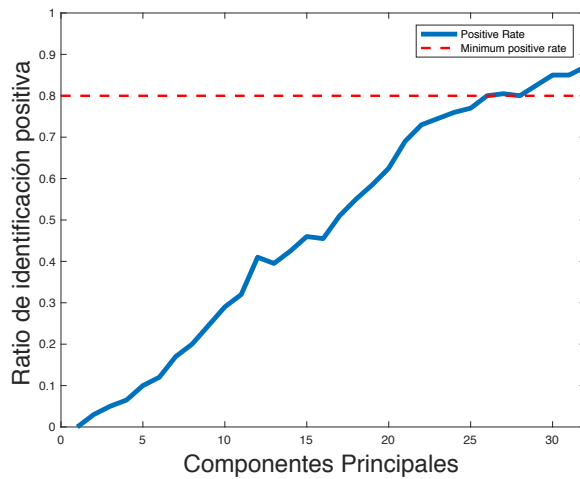


Figura 5.5 - Gráfica que muestra como varía el nivel de acierto en la identificación de un individuo. En el eje x, el número de componentes utilizadas en la identificación. En el eje y, el ratio de acierto conseguido.

5.4 Identificación en condiciones de ruido e iluminación adversas

Para poner a prueba la robustez de ambos algoritmos bajo condiciones de obtención de imágenes adversas, se han realizado pruebas sometiendo a las imágenes de test a condiciones ruidosas y de iluminación desfavorable. La Figura 5.6 muestra la tasa de acierto variando las condiciones de iluminación, mientras que la Figura 5.7 muestra la tasa de acierto conforme añadimos ruido blanco gaussiano.

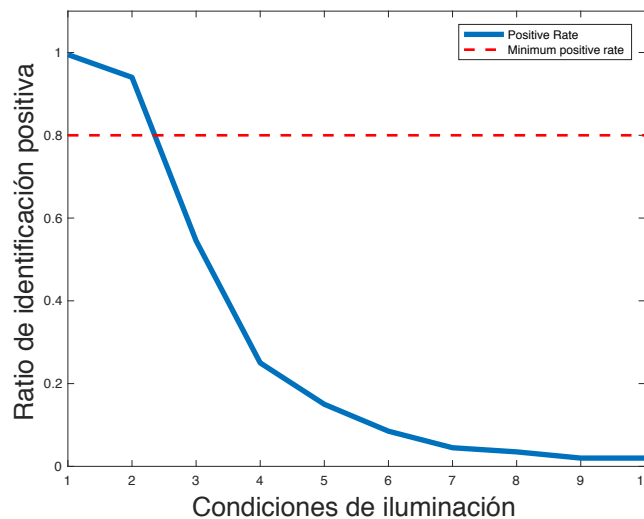


Figura 5.6- La figura muestra el decaimiento en la tasa de acierto del algoritmo AAM a medida que empeoran las condiciones de iluminación.

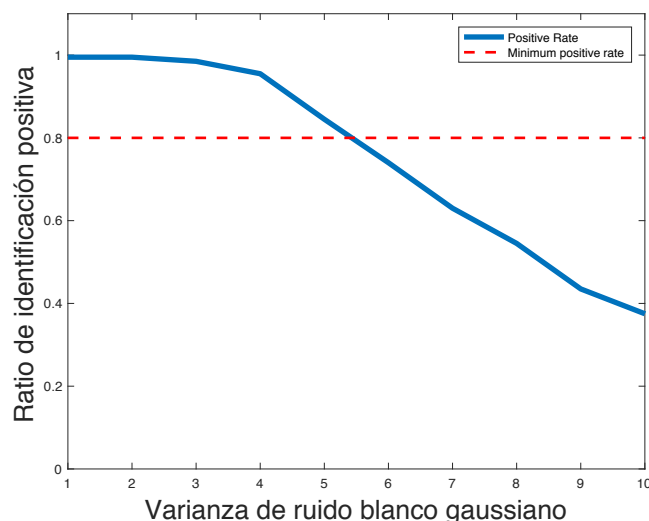


Figura 5.7- La gráfica muestra el decaimiento de la tasa de identificación a medida que aumenta el ruido blanco gaussiano.

Como era de esperar, a medida que las condiciones se vuelven más adversas, el algoritmo comete más errores, como demuestran ambas gráficas. Se ha comprobado que el algoritmo es más robusto frente a interferencias de ruido que frente a una mala iluminación. En la Figura 5.8 y en la Figura 5.9, se muestra el mismo procedimiento realizado, en esta ocasión para el algoritmo Eigenfaces.

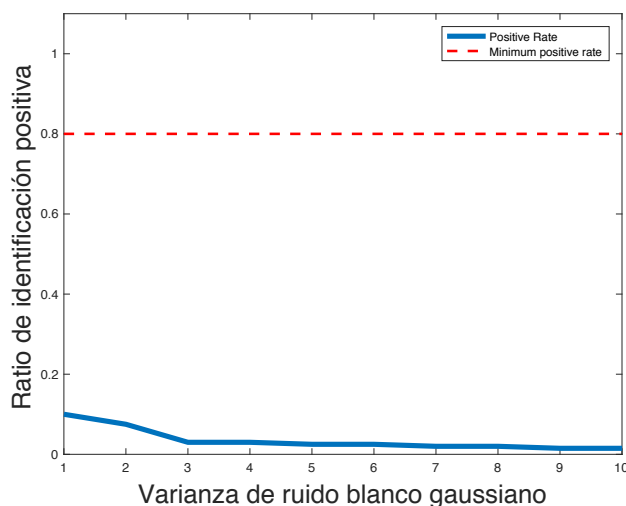


Figura 5.8 - La gráfica muestra el decaimiento de la tasa de identificación del algoritmo Eigenfaces a medida que aumenta el ruido blanco gaussiano. Como se puede observar, nunca llega a alcanzar unos ratios de identificación aceptables.

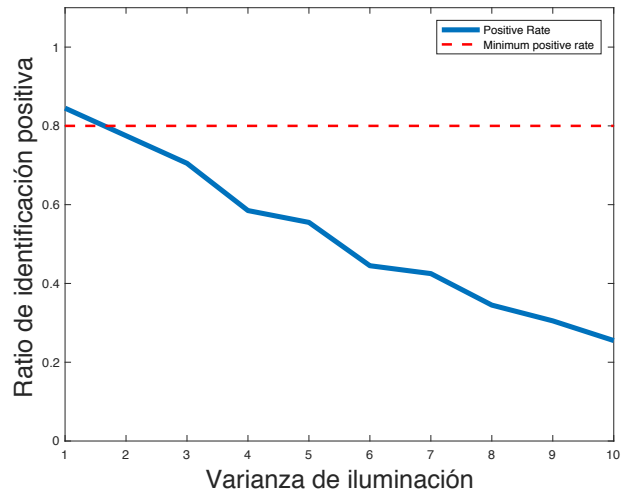


Figura 5.9 - La figura muestra el decaimiento en la tasa de acierto del algoritmo Eigenfaces a medida que empeoran las condiciones de iluminación.

6. Presupuesto de desarrollo

Universidad Politécnica de Valencia
Escuela Técnica Superior de Ingenieros de Telecomunicación
PRESUPUESTO DESARROLLO SISTEMA DE RECONOCIMIENTO FACIAL

MATERIAL	UNIDAD	PRECIO
Ordenador Portátil	1x MacBook Pro 13"	1.200 €
Software desarrollo	1x Licencia Matlab (uso comercial)	2.000 €
	1x Statistics and Machine Learning Toolbox	1.000 €
	1x Neural Network Toolbox	1.000 €
	1x Image Processing Toolbox	1.000 €
	1x Symbolic Math Toolbox	1.000 €
MANO DE OBRA	HORAS	
Graduado Telecomunicaciones	180	4.500 €
	TOTAL	11.700 €

7. Discusión

Hemos sido capaces de codificar cualquier rostro humano en un vector de características individuales. Esto supone una enorme reducción del espacio de almacenamiento frente a almacenar una imagen entera y un paso más en cómo se almacenan las identidades faciales en el cerebro de los primates, confirmando así que podemos reconocer patrones faciales a partir de ejes de números que representan características únicas y aplicarlo a una inteligencia artificial. Hemos demostrado que a partir de dicha codificación podemos reconstruir el rostro con una gran semejanza al rostro real incluso frente a pérdidas de información dentro del vector de características, puesto que se ha demostrado que no es necesario reconstruir el rostro con todas las componentes disponibles. Además, hemos demostrado que ambos algoritmos, el Eigenfaces y el Active Appearance Model, funcionan con una alta fiabilidad en condiciones óptimas de iluminación e interferencias. Se ha comprobado durante la realización del trabajo que el algoritmo Eigenfaces ha requerido de una puesta en marcha y fase de entrenamiento mucho menos costosa que el AAM, puesto que únicamente ha requerido almacenar las imágenes correspondientes al test en una matriz de individuos. Sin embargo, el AAM ha requerido mucho tiempo de preparación para definir el modelo. Ha sido necesario un etiquetado previo de 46 puntos de cada cara de las 200 disponibles en nuestra base de datos y se debe realizar manualmente, lo que ha consumido mucho tiempo del trabajo. No obstante, es precisamente este hecho el que ha permitido hacerlo más robusto frente al Eigenfaces ya que se ha ‘ayudado’ al modelo a predecir cuáles son los puntos a tener en cuenta en un rostro, mientras que el Eigenfaces lo ha interpretado por sí mismo. Este etiquetado es el que nos permitiría hacer al sistema robusto frente a cambios de expresión y cambios en la orientación del individuo, pero supondría marcar manualmente todos los puntos necesarios en cada una de las expresiones y cada una de las orientaciones necesarias para cada una de los rostros de la base de datos, lo que supone demasiada carga de trabajo para un TFG. Sería, por tanto, una buena propuesta de trabajo futuro. Si etiquetamos cada rostro en sus diferentes orientaciones y expresiones faciales, podemos mapear cada punto de la nueva orientación con su homólogo frontal, utilizando técnicas de inteligencia artificial como *linear regression* [25]. Por el contrario, hemos comprobado como el algoritmo Eigenfaces sufre de manera significativa cuando pretendemos que identifique a un individuo con una expresión en su cara (sonriendo en este caso), disminuyendo a un 58,5% su nivel de acierto, lo que lo hace poco fiable en esas condiciones.

Sin embargo, es esta robustez del AAM la que ha perjudicado significativamente su rendimiento. Se ha comprobado como dicho algoritmo es más lento que el Eigenfaces debido a que ha necesitado deformar las imágenes previamente a su análisis. El Eigenfaces, por el contrario, ha extraído las principales componentes de la imagen directamente, sin ningún procesado previo. Se ha podido concluir que, para unas condiciones óptimas de recepción de imágenes frontales, aunque este hecho en la vida real no se produce a menudo, puede ser más adecuado decantarse por el algoritmo Eigenfaces, ya que es capaz de procesar más rostros más rápidamente. Queda para un trabajo futuro la paralelización del algoritmo AAM. Distribuyendo la carga que supone deformar cada imagen entre varios procesadores podemos ser capaces de reducir

considerablemente el tiempo de arranque del modelo y de procesamiento de la imagen de entrada, convirtiendo así al algoritmo AAM en indiscutiblemente mejor algoritmo.

Suponiendo que las imágenes a reconocer son siempre frontales y con expresiones neutras, hemos podido comprobar también como responden ambos algoritmos a condiciones de captación de imágenes adversas. Como se ha podido observar, el exceso de iluminación tiene un efecto más negativo sobre el reconocimiento en AAM mientras que se muestra robusto frente a interferencias de tipo gaussiano. Por su parte, el Eigenfaces se ha mostrado extremadamente ineficaz frente a interferencias de ruido blanco gaussiano, pero se ve ligeramente menos afectado ante peores condiciones de iluminación. El efecto de contrarrestar el exceso de iluminación podría combatirse con filtros de luz y AAM es un sistema razonablemente robusto ante interferencias del tipo ruido blanco gaussiano, lo que le da cierto margen para reconocer rostros con cámaras de baja calidad.

8. Conclusión

Tras la realización del estudio, se ha concluido que el algoritmo AAM es más fiable en cuanto al reconocimiento de individuos y más codificable. Dispone de un mayor porcentaje de éxito en condiciones normales que el Eigenfaces, tiene una mayor tolerancia a ruido y a malas condiciones de iluminación, es robusto frente a cambios de expresión y frente a variaciones en la orientación y permite una mayor codificación. Sin embargo, la preparación del modelo es lenta y trabajosa y el procesado de imágenes puede ser demasiado lento. Por el contrario, el algoritmo Eigenfaces es menos robusto y ha penalizado en exceso los cambios en las expresiones faciales. A su favor, es un algoritmo mucho más rápido que el AAM. Además, la codificación en vectores de pocos elementos se corresponde con el modelo planteado sobre la identificación facial en cerebro de primates, alejada de la creencia de que almacenamos una imagen de un rostro determinado de cada individuo que conocemos.

9. Bibliografía y referencias

- [1] Cognitec Systems GmbH, <http://cognitec-systems.de/contact/contact.html>
- [2] Identix, <http://identix.com>
- [3] Apple Inc, <https://apple.com/lae/iphone-x/>
- [4] International Biometric Group, <http://biometricgroup.com>
- [5] R.Hietmeyer, "*Biometric identification promises fast and secure processing airline passengers*", The Int'l Civil Aviation Organization Journal, vol. 55, no. 9, pp. 10-11, 2000.
- [6] Face Recognition Homepage, <http://face-rec.org>
- [7] M.Turk, A. Pentland, "*Eigenfaces for Recognition*", Journal of Cognitive Neuroscience, Vol. 3, No.1, 1991, pp. 71-86.
- [8] G.J. Edwards, T.F. Cootes and C.J. Taylor, "*Face Recognition Using Active Appearance Models*", in Proc. European Conference on Computer Vision 1998 (H.Burkhardt & B. Neumann Ed.s). Vol. 2, pp. 581-695, Springer, 1998
- [9] Xiaoguang Lu, Dirk Colbry, and Anil K. Jain, "*Three-Dimensional Model Based Face Recognition*", ICPR August 2004.
- [10] Claudi Martínez "*FACE recognition in the context of Smart Rooms*", PFC Department of Signal Theory and communications UPC.
- [11] M. Kirby and L. Sirovich, "*Application of the Karhunen-Loeve procedure for the characterization of human faces*" IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 12, no. 1, pp. 103–108, Jan. 1990.
- [12] Pedro Alexandre Dias Martins "*Active Appearance Models for Facial Expression, Recognition and Monocular Head Pose Estimation*" MSc Thesis. Department of Electrical and Computer Engineering. University of Coimbra.
- [13] Jieping Ye, Ravi Janardan, Qi Li. "*Two-Dimensional Linear Discriminant Analysis*", NIPS'04
- [14] Naotoshi Seo. "*Eigenfaces and Fisherfaces*". University of Maryland
- [15] Duda R., Hart P., Stork D.. "*Pattern Classification*" Chapter 4.
- [16] Xiaofei He & Partha Niyogi. "*Locality Preserving Projections*", NIPS'03 volume 16.
- [17] Yong Xu and David Zhang. "*A New Solution Scheme of Unsupervised Locality Preserving Projection Method for the SSS Problem*", SSPR'08 (International Workshop on Structural, Syntactic, and Statistical Pattern Recognition),775-781.
- [18] Ziad M. Hafed and Martin D. Levine. "*Face Recognition Using the Discrete Cosine Transform*". International Journal of Computer Vision 43(3), 167–188, 2001.
- [19] Abdallah S. Abdallah, A. Lynn Abbott, and Mohamad Abou El-Nasr. "*A New*

Face Detection Technique using 2D DCT and Self Organizing Feature Map".
Proceedings of World Academy of Science, Engineering and technology Vol. 21
May 2007 ISSN 1307-6884

- [20] Hazim Kemal Ekenel, Rainer Stiefelhagen *"Local appearance based face recognition using discrete cosinus transform"*, 13th European Signal Processing Conference (EUSIPCO 2005).
- [21] T. Ezzat and T. Poggio. *"Facial Analysis and Synthesis Using Image-Based Models"*. In International Workshop on Automatic Face and Gesture Recognition 1996, pages 116–121, Killington, Vermont, 1996.
- [22] C.Nastar,B.Moghaddam,andA.Pentland.*"Generalized Image Matching: Statistical Learning of Physically-Based Deformations"*. In European Conference on Computer Vision, volume 1, pages 589–598, Cambridge, UK, 1996.
- [23] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. *"Active Shape Models - Their Training and Application"*. Computer Vision, Graphics and Image Understanding, 61(1):38– 59, 1995.
- [24] Quiroga, R.Q., Reddy, L., Kreiman, G., Koch, C., and Fried, I. (2005). *"Invariant visual representation by single neurons in the human brain"*. Nature 435, 1102–1107.
- [25] Le Chang, Doris Y. Tsao (2017). *"The Code for Facial Identity in the Primate Brain"*. Cell 169, 1013-1028.
- [26] Freiwald, W.A., Tsao, D.Y., and Livingstone, M.S. (2009). *"A face feature space in the macaque temporal lobe"*. Nat. Neurosci. 12, 1187–1196.
- [27] Schwartz, O., Pillow, J.W., Rust, N.C., and Simoncelli, E.P. (2006). *"Spike-triggered neural characterization"*. J. Vis. 6, 484–507.
- [28] Matlab Documentation: *imwarp* function.
<https://es.mathworks.com/help/images/ref/imwarp.html>
- [29] FEI Face Database, <http://fei.edu.br/~cet/facedatabase.html>
- [30] Matlab Image Processing Toolbox,
<https://es.mathworks.com/help/images/index.html>
- [31] Matlab Statistics and Machine Learning Toolbox,
<https://es.mathworks.com/help/stats/index.html>
- [32] Matlab Symbolic Math Toolbox,
<https://es.mathworks.com/help/symbolic/index.html>
- [33] Matlab neural Network Toolbox,
<https://es.mathworks.com/help/nnet/index.html>
- [34] Mejda Chihaoui, Akram Elkefi, Wajdi Bellil and Chokri Ben Amar, *"A Survey of 2D Face Recognition Techniques"*(2016). Computer vol.5, Issue 4.