Reconocimiento de las personas por los rasgos de la cara

PID_00215065

Francesc Serratosa

Universitat Oberta de Catalunya

Índice

Introducción

El reconocimiento de las personas a través de la cara es el método de identificación de las personas más usado por los humanos. Otro método natural es el olor, que es uno de los métodos más usados por muchos animales, pero esta técnica de reconocimiento dejó de ser usada por los humanos hace muchos miles de años. El reconocimiento de las personas por la cara es una tarea que llevamos a cabo constantemente y sin un esfuerzo aparente. No obstante, en la inteligencia artificial, sigue siendo una tarea de gran esfuerzo computacional y de recursos. La aparición de dispositivos de bajo coste y con mucha capacidad de cálculo, como portátiles o teléfonos inteligentes, ha despertado un enorme interés por el procesado de imágenes y vídeos en muchas aplicaciones, entre ellas está la identificación a través de la biometría y el seguimiento de las personas o de sus caras. Una aplicación típica es el seguimiento automático de la cara de un conferenciante.

La investigación en el reconocimiento de caras está motivada no solo por el desafío de la metodología en sí, sino también por numerosas aplicaciones prácticas donde la identificación de las personas es necesaria. El reconocimiento de caras es uno de los métodos biométricos más usados y se ha vuelto cada vez más importante debido a los avances tan rápidos en las tecnologías de las cámaras digitales, Internet, dispositivos móviles y la demanda cada vez más grande de medidas de seguridad. La capacidad de cálculo del procesador de una cámara de fotos de bajo coste es superior a los ordenadores de sobremesa de principios de los ochenta. Así han podido aparecer los sistemas de detección de caras en cámaras de bajo coste.

Las principales ventajas del reconocimiento de las personas por la cara son dos:

1) es el método más natural, al que estamos acostumbrados y, por lo tanto, es menos intrusivo; y

2) es el método más fácil de usar de cara al usuario, simplemente una cámara tiene que capturar la cara con poca o ninguna colaboración del usuario.

Además, es un sistema biométrico que puede trabajar en un proceso de verificación (o autentificación) o en un proceso de identificación (o reconocimiento). Han aparecido aplicaciones que ejecutan de forma automática estos dos procesos (por ejemplo, Picassa o iPhoto). El objetivo de estas es ir buscando caras conocidas en fotos personales y así poder mostrar en qué fotos aparece una persona concreta. Hay un proceso de aprendizaje donde la aplicación selecciona las caras y pregunta de qué persona son. Cuantas más caras se verifican, más aprende el sistema y, por lo tanto, más caras encuentra y más clasifica en nombres concretos. Uno de los aspectos prácticos interesantes es que

el aprendizaje se puede hacer en cualquier momento, es decir, que siempre se está a tiempo de corregir caras mal clasificadas o clasificar caras de las que el sistema duda y pide ayuda. Además, cuando se van añadiendo fotos nuevas, el sistema va detectando automáticamente las nuevas caras que puedan aparecer.

El orden en el que se tratan las técnicas y procesos en este módulo tiene cierta relación con el flujo de la información en el proceso general de la comparación de dos caras o la busca de una cara en una base de datos. Vamos a comentar estos métodos de la forma siguiente:

- En el apartado "Etapas de los sistemas de reconocimiento de personas por la cara" tratamos la existencia de las cinco etapas fundamentales de un sistema de reconocimiento de caras.

- Después, pasamos a detallar cada una de las etapas. De este modo, en el apartado "Detección de las caras" explicamos cómo detectar una cara dentro de una imagen, es decir, cómo saber que en alguna parte o algunas partes de la imagen hay una cara o varias caras.

- En el apartado "Normalización de la cara", explicamos un proceso de normalización de la parte de la imagen donde creemos que hay una cara. Esta normalización tiene el objetivo de tener la información de la cara independientemente de la posición de la persona y de la luz.

- Una vez tenemos la imagen normalizada, en el apartado "Extracción de las características" extraemos las características principales.

- En el apartado "Verificación de una cara" ya tenemos el proceso específico de los sistemas biométricos, que se basa en verificar si dos imágenes de caras pertenecen a la misma persona.

- Como continuación natural, en el apartado "Identificación de una cara" se explica cómo identificar a una persona a través de la cara.

- Finalmente, en el apartado "Generación de caras" se explica cómo generar una cara de forma sintética a partir de un conjunto de caras. Esta técnica es interesante cuando no tenemos la imagen completa de la cara de una persona y necesitamos tener una buena aproximación de esta.

Objetivos

Los objetivos básicos de este módulo son los siguientes:

1. Conocer las etapas básicas de los sistemas de reconocimiento de las personas basados en la cara.

2. Conocer las técnicas de la visión por computador aplicadas a la extracción de las características principales de la cara.

3. Representar una cara en un registro de una base de datos.

4. Comparar dos caras.

5. Generar una cara de forma sintética usando un conjunto de caras.

1. Etapas de los sistemas de reconocimiento de personas por los rasgos de la cara

El primer sistema hecho público de reconocimiento de personas por la cara fue la tesis doctoral presentada por **Kanade** en 1974. Desde aquel primer sistema hasta ahora, el rendimiento de los sistemas de reconocimiento de caras ha mejorado significativamente. Además, ahora este proceso se puede llevar a cabo en tiempo real con imágenes o vídeos captados en situaciones más o menos favorables. No obstante, aunque la mejora en estos sistemas ha sido alentadora, esta tarea ha resultado ser un esfuerzo difícil, en especial por situaciones no controladas donde puede haber cambios de iluminación, expresiones, puntos de vista u oclusiones. Los humanos y algunos animales tienen una capacidad de reconocimiento muy superior a los sistemas artificiales. Hoy por hoy, no se sabe cuándo un sistema artificial será mejor que uno natural.

Hoy en día, las etapas principales de los sistemas de reconocimiento de personas a través de la cara no han variado demasiado respecto a la primera solución propuesta por Kanade. Lo que ha variado es la solución aplicada dentro de estas etapas. Todos los sistemas suelen estar formados por cinco grandes módulos que pasamos a describir.

1) Detección y seguimiento de la cara[1]: Extrae la zona donde puede estar la cara dentro de la imagen. En el supuesto de que trabajemos directamente con una secuencia de vídeo, también tendremos que hacer el seguimiento de la cara, es decir, ir sabiendo en qué zona de la imagen se encuentra la cara, imagen a imagen. No vamos a explicar nada más sobre los sistemas de seguimiento de caras[2], simplemente mencionamos que la ventaja de trabajar en vídeo es que se dispone de una información pasada (posición, velocidad, aceleración, orientación, tamaño) que se usa para predecir estos datos en imágenes futuras.

(1)En inglés, *face detection.*

(2)En inglés, *face tracking.*

2) Normalización de la cara[3]: El objetivo es normalizar la imagen de la cara respecto a las propiedades geométricas tales como tamaño y posición usando transformaciones geométricas o morfológicas. También se normaliza respecto a la iluminación. El objetivo de estas transformaciones es que la imagen resultante sea lo más invariante posible a todos los agentes externos (como focos, iluminación, posición relativa de donde está la cámara respecto a donde está la cara, zoom de la cámara). Los componentes faciales como los ojos, las orejas, la nariz o la boca se detectan y se usan para hacer las transformaciones necesarias.

(3)En inglés, *face alignment.*

3) Extracción de las características[4]: El objetivo es obtener información efectiva de la imagen de la cara que sea útil para distinguir entre dos caras diferentes pero en cambio que sea invariante a dos imágenes de la misma persona. Además, se tiene que lograr que sea estable respecto a los agentes externos.

(4)En inglés, *face extraction*

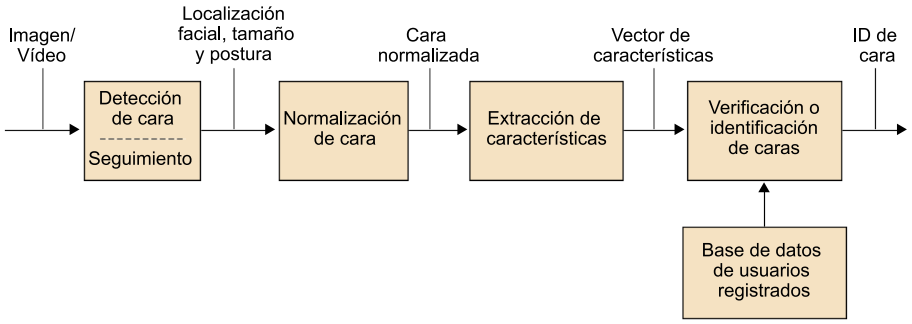
4) Verificación o identificación de caras[5]: El objetivo de este módulo depende de si estamos en un proceso de identificación o de verificación. En un proceso de identificación, compara las características extraídas de una nueva imagen con las características de varias imágenes de la base de datos y devuelve la más parecida (o conjunto de más parecidas). En un proceso de verificación, compara las características de dos imágenes y devuelve si son la misma persona o no. También puede devolver una distancia o probabilidad de que sean la misma persona.

(5)En inglés, *feature matching*.

5) Base de datos[6]: Este módulo almacena las características de todas las imágenes que nos han servido para matricular a las personas. En los procesos de identificación, va ofreciendo características de caras para que sean comparadas. Suelen tener unos mecanismos internos denominados *indexado de base de datos*, que sirven para que no sea necesario recorrer toda la base de datos y así reducir el tiempo de busca.

(6)En inglés, *database of unrolled users*.

Figura 1. Proceso de identificación o verificación de personas por la cara



2. Detección de las caras

La detección de la cara es el primer paso del proceso de identificación de las caras. Su fiabilidad tiene una influencia capital en todo el sistema de reconocimiento puesto que, si la cara no se detecta, entonces no se aplica ningún otro proceso en aquella parte de la imagen. Dada una sola imagen o una secuencia de vídeo, un detector de caras ideal tendría que tener la capacidad de localizar todas las caras, a pesar de la posición, el escalado, la edad o la expresión, por ejemplo. Además, tendría que ser independiente de iluminaciones extrañas y del contenido de la imagen o el vídeo. Notad que detectar una cara quiere decir que el sistema sepa que en aquellos píxeles de la imagen hay una cara pero todavía no sabe qué persona es.

Picassa y el iPhone

En las aplicaciones Picassa o iPhone, si no se detecta la cara, no hay manera de seleccionar manualmente dónde hay una cara.

Se han diseñado varios métodos para localizar caras en una imagen y también para el seguimiento y detección de caras en secuencias de vídeo.

Reflexión

Aquí trataremos uno de los métodos más usados para reconocer caras en imágenes estáticas debido a su sencillez y eficacia. No haremos ningún comentario respecto a los sistemas de seguimiento de caras en vídeos.

La localización de la cara por el **método basado en la aparición**[7] se basa en la idea de buscar si la cara aparece en cualquier punto de la imagen. El proceso se lleva a cabo de la manera siguiente: la imagen de entrada se escanea en todas las posiciones posibles y escalados diferentes por una subventana. Esta subventana normalmente es un cuadrado de diferentes dimensiones. La detección de la cara se plantea como un sistema que analiza solo la subventana y sabe clasificar la zona de la imagen que cae dentro de la subventana en solo dos clases. Una clase es "cara" y la otra clase es "no cara". Los píxeles externos a la subventana no afectan a esta decisión. La subventana sirve para aislarnos del resto de la imagen. El clasificador que decide si es una cara o no es una cara ha sido enseñado con muchos ejemplos de subventanas que son caras y muchos ejemplos de subventana que no son caras. Tal como vamos a ver a continuación, estos ejemplos sirven para aportar información estadística. Si las medidas de la subventana son, por ejemplo, de 20 × 20 píxeles, entonces detectaremos caras en la imagen que ocupan 20 × 20 píxeles. Y para entrenar el clasificador, tendremos que pasar imágenes de 20 × 20 que sean caras e imágenes de 20 × 20 que no sean caras.

(7)En inglés, *appearance-based detection*.

En la figura 2, se ven diez imágenes de 20 × 20 píxeles usadas para aprender la clase "cara" y diez imágenes de 20 × 20 píxeles usadas para aprender la clase "no-cara".

Figura 2. Imágenes usadas para aprender la clase "cara" e imágenes usadas para aprender la clase "no cara"

Aquí vamos a presentar el clasificador más sencillo que se ha diseñado, se denomina **clasificador de Bayes**. Existe otro clasificador también muy usado llamado **Adaboost**, que normalmente funciona algo mejor. Este otro clasificador complica un poquito la idea inicial de Bayes añadiendo unas ponderaciones o pesos. Las muestras que vamos deduciendo de forma iterativa que están mal clasificadas se tienen muy en cuenta puesto que el peso con que se las pondera va aumentando. Estos dos clasificadores de "cara" y "no cara" aprenden por qué en las subventanas que contienen una cara aparece una correlación entre sus píxeles. Por ejemplo, siempre que una zona está muy iluminada, también lo está esta otra zona o, al revés, siempre que está muy iluminada, detectamos que esta otra zona es muy oscura. Por otro lado, se supone que esta correlación no se detecta en las subventanas donde hay cualquier otro objeto o parte de objeto que no sea una cara.

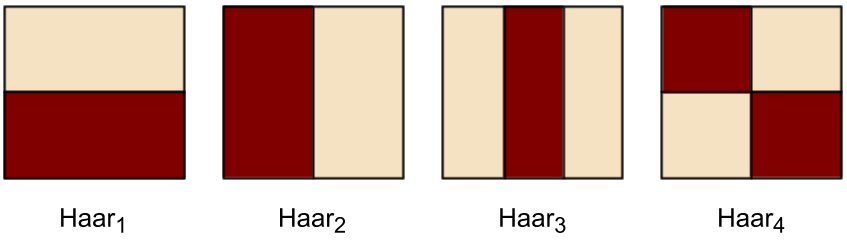
Estos clasificadores no reciben directamente la subventana de la imagen, sino un conjunto de características (normalmente es un vector de números reales) extraídas de esta subventana de la imagen. Hay varios métodos para generar estas características. Los más usados son las **características de Haar** propuestas por **Viola y Jones** en el año 2001. Otro método consiste en extraer las características a través de **filtros de Gabor**. Pasemos a describir las características de Haar.

Reflexión

El método que utiliza los filtros de Gabor no se explica en este material didáctico.

Las características de Haar (figura 3) están compuestas por cuatro imágenes de varias dimensiones con formas rectangulares bicolores.

Figura 3. Las cuatro características básicas de Haar



$F_i(I_j)=a_i^j$

Cuando se ubica una característica de Haar sobre una subventana, los píxeles de la subventana que caen sobre la parte blanca se suman y los píxeles de la subventana que caen sobre la parte negra se restan. El valor final de aplicar un filtro concreto en una posición concreta es el valor resultante de sumar las partes blancas y restar las partes negras.

Cada tipo se puede concretar en una figura según los parámetros siguientes:

- Ubicación de la característica de $Haar_i$ en la imagen (x, y). Normalmente, el centro de coordenadas de los píxeles de la imagen original así como la subventana donde se aplica Haar se localiza en el extremo inferior izquierda y las coordenadas empiezan por el punto (0, 0).

- Diferentes dimensiones horizontales y verticales (h, w) de la característica de $Haar_i$.

Es decir, que dada una subventana aplicaremos $Haar_i(x, y, h, w)$ y devolverá un valor entero.

Ejemplo

Dada esta pequeña imagen original de 6 × 6:

23	56	67	78	56	32
23	55	12	90	234	105
245	211	178	190	176	87
23	54	78	189	232	190
23	123	239	123	56	32
203	23	29	12	116	132

$Haar_2(1, 1, 4, 4)$ devuelve el valor –(55 + 12 + 211 + 178 + 54 + 78 + 123 + 239) + (90 + 234 + 190 + 176 + 189 + 232 + 123 + 56) = –950 + 1290 = 340.

Por lo tanto, el vector de entradas del clasificador de Bayes está compuesto por los cuatro tipos de características en posiciones diferentes y dimensiones diferentes. Hay muchas combinaciones posibles, por eso se tiene que decidir un subconjunto.

Ejemplo

En subventanas de 24 × 24, se puede obtener un vector de 80 valores. Es decir, 20 combinaciones diferentes por cada tipo de Haar. Si establecemos filtros de Haar de 6 × 6, entonces podemos poner 16. Y si también consideramos filtros de Haar de 12 × 12, entonces podemos poner cuatro. Lógicamente, se pueden poner otras muchas combinaciones, por ejemplo, aceptando el solapamiento entre filtros o que no tengan que ser cuadrados.

Una vez tenemos convertida la subventana en un vector de n características x, pasamos a explicar cómo montamos un clasificador de Bayes. Antes de nada, tenemos que deducir cuál es la probabilidad de que el valor de la i-ésima característica concreta sea de la clase "cara" $P_i^c(x_i)$ y también que sea de la clase "no cara" $P_i^{nc}(x_i)$. Una vez tengamos estas probabilidades, asumiremos que son independientes entre ellas y entonces consideramos que una muestra x es de clase "cara" o de clase "no cara" según el producto de probabilidades de todas las características de la forma siguiente:

La muestra *x* es "cara" si $\prod_{i=1}^{n} P_i^c(x_i) > \prod_{i=1}^{n} P_i^{nc}(x_i)$. De lo contrario, es "no cara".

Ahora nos queda comentar cómo calculamos las probabilidades $P_i^c(x_i)$ y $P_i^{nc}(x_i)$. En el método de Bayes, estas probabilidades se llaman **probabilidades a posteriori** y se calculan a través de otras probabilidades marginales y condicionales. La probabilidad a posteriori de la clase "cara" se define así:

$$P_i^c(x_i) = \frac{P(x_i|"cara") \cdot P("cara")}{P(x_i)} \qquad 4.1$$

De manera similar, la probabilidad a posteriori de la clase "no cara" se define como:

$$P_i^{nc}(x_i) = \frac{P(x_i|"no \quad cara") \cdot P("no \quad cara")}{P(x_i)} \qquad 4.2$$

Las probabilidades condicionales $P(x_i|"cara")$ y $P(x_i|"no \quad cara")$ se denominan **probabilidades a priori** y son la función de posibilidad (*likelihood*) de la clase "cara" y de la clase "no cara" respecto al valor x_i de la *i*-ésima característica. Las probabilidades de las clases $P("cara")$ y $P("no\,cara")$ son la probabilidad a priori de la clase "cara" y de la clase "no cara". La probabilidad de que esta muestra x_i aparezca, $P(x_i)$, se calcula con el teorema de la probabilidad conjunta:

$$P(x_i) = P(x_i|"cara") \cdot P("cara") + P(x_i|"no \quad cara") \cdot P("no \quad cara") \qquad 4.3$$

Solo nos queda definir las probabilidades de las clases y las probabilidades a priori. Normalmente, se considera que existe la misma probabilidad de que en la subventana aparezca una cara que de que no aparezca. Por ese motivo, se puede asumir que:

$$P("cara") = P("no \quad cara") = 1/2 \qquad 4.4$$

Con la definición de estas probabilidades, podemos redefinir la fórmula general de la probabilidad a posteriori para las dos clases de la manera siguiente:

$$P_i^c(x_i) = \frac{P(x_i|"cara")}{P(x_i|"cara") + P(x_i|"no \quad cara")} \qquad 4.5$$

y

$$P_i^{nc}(x_i) = \frac{P(x_i|"no\ \ cara")}{P(x_i|"cara") + P(x_i|"no\ \ cara")} \qquad 4.6$$

Esta función nos indica que para considerar que un elemento sea de tipo "cara" (o "no cara") es importante que su probabilidad a priori sea alta, pero también es importante que la probabilidad a priori de la otra clase sea baja.

Ahora pasemos a definir las probabilidades a priori. Primero empezamos por la probabilidad de ser una clase de tipo "cara". Dado el conjunto de *k* subventanas en el que sabemos que hay una cara, calculamos los vectores de sus *n* características y obtenemos el conjunto de vectores $\{a^1, a^2, \ldots, a^k\}$ donde:

$$a^j = \left[a_1^j, \ldots a_n^j\right] \qquad 4.7$$

El elemento a_i^j es el valor de la característica *i*-ésima de la muestra *j*-ésima en la que sabemos que hay una cara. Esta característica se calcula con una función de Haar aplicada a la muestra *j*-ésima con unos parámetros (*x*, *y*, *h*, *w*) concretos. Con estos vectores de características, vamos a calcular el valor medio de cada característica:

$$\mu_i^c = \frac{1}{k}\sum_{j=1}^{k} a_i^j \qquad 4.8$$

y también su desviación estándar:

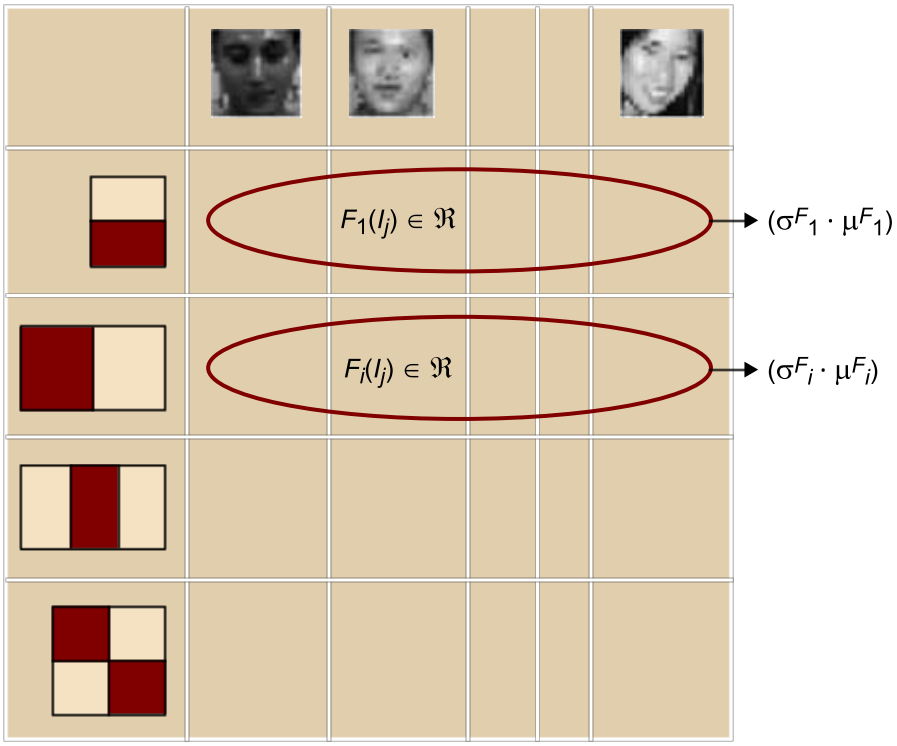
$$\sigma_i^c = \sqrt{\frac{1}{k-1}\sum_{j=1}^{k}\left(a_i^j - \mu_i^c\right)^2} \qquad 4.9$$

Entonces, asumimos que estas características se pueden aproximar por una función normal y obtenemos la probabilidad a priori de ser cara dada la muestra siguiente:

$$P(x_i|"cara") = \frac{1}{\sigma_i^c\sqrt{2\pi}}\, e^{\frac{-\left(x_i-\mu_i^c\right)^2}{2\left(\sigma_i^c\right)^2}} \qquad 4.10$$

La figura 4 muestra de forma gráfica la obtención de la desviación estándar y la media de cada característica dadas las imágenes de la clase "cara". En la matriz, hay tantas columnas como imágenes de caras y tantas filas como filtros de Haar se hayan aplicado dependiendo de los cinco parámetros de la función Haar $_i(x, y, h, w)$. En la figura, se muestran cuatro posibles valores del parámetro *i* y no se concretan los otros cuatro parámetros.

Figura 4. Extracción de la desviación estándar y la media dadas las imágenes de caras y cada combinación del filtro de Haar



El proceso para calcular la probabilidad de clase "no cara" es similar al descrito con la clase "cara". Del mismo modo, calcularemos los vectores $\{b^1, b^2, \ldots, b^k\}$ que representan lo mismo y se calculan del mismo modo que los vectores $\{a^1, a^2, \ldots, a^k\}$ pero las muestras iniciales son de subventanas de la clase "no cara". Con estos vectores de características, vamos a calcular el valor medio de cada característica:

$$\mu_i^{nc} = \frac{1}{k}\sum_{j=1}^{k} b_i^j \qquad 4.11$$

y también su desviación estándar:

$$\sigma_i^{nc} = \sqrt{\frac{1}{k-1}\sum_{j=1}^{k}\left(b_i^j - \mu_i^{nc}\right)^2} \qquad 4.12$$

Entonces, asumimos que estas características se pueden aproximar por una función normal y obtenemos la probabilidad de no ser cara dada la muestra:

$$P(x_I|"no \quad cara") = \frac{1}{\sigma_i^{nc}\sqrt{2\pi}} e^{\frac{-\left(x_I-\mu_i^{nc}\right)^2}{2\left(\sigma_i^{nc}\right)^2}} \qquad 4.13$$

La figura 5 muestra una fotografía donde se han detectado automáticamente las caras. Fijaos en que aparecen dos tipos de errores de detección que son parecidos a los errores tratados en el módulo "Evaluación de los sistemas biométricos en aplicaciones reales". Algunas caras no han sido detectadas, **falso rechazo**[8], y algunas partes que no son caras han sido consideradas caras, **falsa aceptación**[9]. No obstante, el número de aceptaciones correctas[10] es muy superior al de errores.

Figura 5. Fotografía donde se han detectado las caras de forma automática

[8]En inglés, *false rejection*.

[9]En inglés, *false acceptance*.

[10]En inglés, *correct acceptance*.

3. Normalización de la cara

Tal como ya hemos destacado, los métodos basados en la aparición operan en subventanas de tamaño fijo. Por esa razón, es necesario ajustar el tamaño de todas las ventanas a un tamaño estándar, por ejemplo 20 × 20. Esto se debe a que los métodos que vamos a comentar más adelante para el reconocimiento de caras necesitan tener un número de píxeles concreto. Además, el método de Haar pretende ser independiente a los cambios de iluminación. Por ello, es necesario un proceso posterior de normalización de las subventanas donde se ha detectado la cara según su intensidad. Si este proceso no se ejecutara, el reconocimiento de caras se podría llevar a cabo pero no funcionaría tan bien. Estas operaciones son globales desde el punto de vista de que todos los píxeles de la subventana se modifican.

3.1. Normalización de las dimensiones de la subventana

Normalmente, las subventanas son cuadradas, ya que las caras se pueden encuadrar bastante bien en un cuadrado. Supongamos que la subventana que ha detectado la cara tiene unas dimensiones de (l_1, l_1) y nosotros deseamos que todas las subventanas donde hay posibles caras tenga unas dimensiones de (l_2, l_2). Cada píxel de la nueva subventana tomará el valor siguiente:

$$I_2(x, y) = I_1\left(\left\lfloor x \cdot \frac{l_1}{l_2} \right\rfloor, \left\lfloor y \cdot \frac{l_1}{l_2} \right\rfloor\right) \qquad 4.14$$

Las dimensiones normalizadas suelen ser las de la ventana más pequeña. Esto es así porque, en caso contrario, nos encontraríamos con que tendríamos que llenar algunos píxeles de la nueva subventana de forma artificial puesto que algunos píxeles de la nueva ventana no serían llenados por ningún píxel de la ventana original. Otro método más inteligente para llenar las nuevas posiciones se basa en encontrar la media ponderada de los píxeles de alrededor. Este método es el más usado cuando se modifican las dimensiones de imágenes aunque tiene el inconveniente de que tiene un efecto de filtrado.

3.2. Normalización de la intensidad de la subventana

Hay varias técnicas para normalizar la imagen desde un punto de vista de la intensidad. Aquí vamos a tratar el método basado en la **ecualización del histograma**. Este método se basa en aumentar el contraste de la imagen ajustando los valores del histograma. Mediante este ajuste, las intensidades de los píxeles se distribuyen más equitativamente en el histograma. Primero, tendremos que calcular la intensidad de píxel mínima, $v_{\min}$ y máxima, $v_{\max}$. Es decir, los valores de los píxeles a partir de los cuales ningún píxel de la imagen tiene

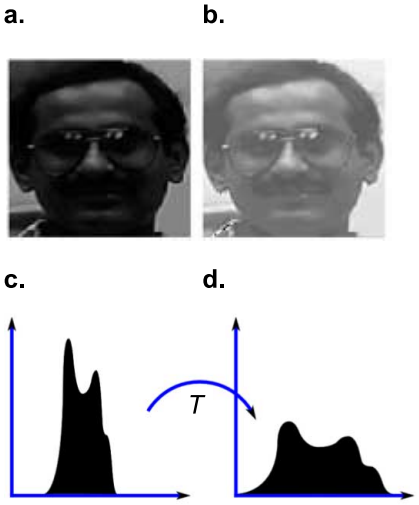
un valor inferior o superior (a veces, en lugar de ninguno se considera un número pequeño). Entonces, aplicaremos una transformación global a todos los píxeles de la imagen según la ecuación:

$$I_2(x, y) = \frac{I_1(x, y) - V_{min}}{V_{max} - V_{min}} \cdot MAX \qquad 4.15$$

donde *MAX* representa el valor máximo posible del píxel (habitualmente es 255).

Esta transformación de la imagen supone que el histograma resultante llena todo el rango posible de valores. La figura 6 muestra una imagen original y la misma imagen en la que se ha ecualizado su histograma. También se muestra el histograma original y el histograma ecualizado.

Figura 6. Imagen ecualizada



a. Imagen original; b. imagen tratada; c. histograma antes del proceso; d. histograma ecualizado

4. Extracción de las características

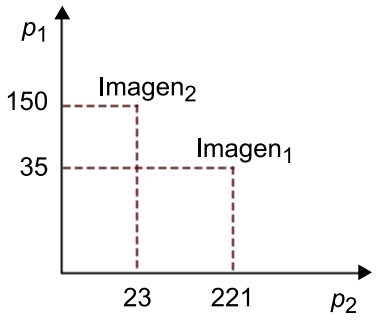
Una **imagen** se puede ver como un punto en un espacio discreto que tiene tantas dimensiones como píxeles tiene la imagen y el número de valores de cada dimensión es la profundidad del píxel (normalmente 256 × 3 canales; rojo, verde y azul).

Notad que este espacio es enorme. Supongamos una webcam de baja resolución (640 × 480). Esto quiere decir que nuestro espacio tiene 307.200 dimensiones y cada dimensión está discretizada en 256 × 3 valores.

Ejemplo

Imaginemos una cámara de muy baja resolución. Tan baja que las imágenes que toma solo tienen dos píxeles. En este caso, las imágenes se pueden definir en un espacio de dos dimensiones. La dimensión x_1 representa el píxel p_1 y la dimensión x_2 representa el píxel p_2. Como esta cámara de dos píxeles toma solo fotografías en escala de grises, tenemos que la profundidad de los píxeles es de 256 y, entonces, las dimensiones de este nuevo espacio discreto son de 256 posiciones. Supongamos que tomamos dos fotos con esta cámara: $\text{Imagen}_1 = [35|221]$. Esto quiere decir que el valor del píxel $p_1 = 35$ y el valor del píxel $p_2 = 221$. Y la segunda foto es $\text{Imagen}_2 = [150|23]$. Si representamos estas fotografías en el nuevo espacio tendremos (figura 7):

Figura 7. Representación de dos fotografías hechas con una cámara de solo dos píxeles y con escala de grises



El método que vamos a tratar considera este espacio e intenta reducir el número de dimensiones de este espacio. Normalmente, se lleva a cabo una reducción drástica, ya que se acostumbra a pasar de 307.200 dimensiones a unas 7. Esta reducción se puede llevar a cabo por dos motivos principales:

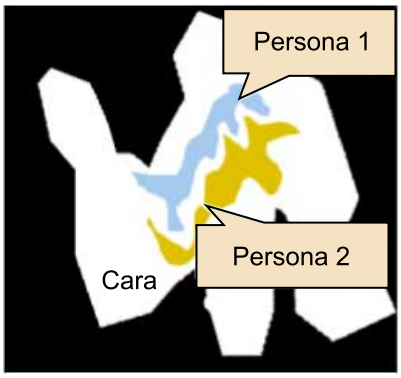
1) solo una pequeña zona de este enorme espacio es la que contiene imágenes con caras, y

2) se lleva a cabo una transformación del espacio y este nuevo espacio donde vamos a parar es el espacio de la covarianza entre los píxeles de las imágenes.

Llevaremos a cabo la reducción con el método de **análisis de los componentes principales (PCA)** que se explica más adelante.

En este nuevo espacio, podemos analizar qué fotografías han sido tomadas a caras, cuáles no e, incluso, podemos agrupar las fotos de caras dependiendo de la persona. De este modo, nos podría quedar un espacio como el que muestra la figura 8.

Figura 8. Espacio bidimensional donde se representa la zona donde hay fotografías de caras y, dentro de esa zona, dónde están las caras de dos personas



Debido a que en este espacio es difícil separar de forma lineal las zonas donde hay caras de las personas, transformamos este espacio en un espacio donde se considera la variabilidad entre los píxeles, es decir, la variabilidad entre las coordenadas de este nuevo espacio.

El **análisis de los componentes principales (PCA)** es una técnica de reducción de las dimensiones del espacio basada en obtener solo un número impuesto como parámetro de entrada de componentes principales de datos multidimensionales.

Ved también

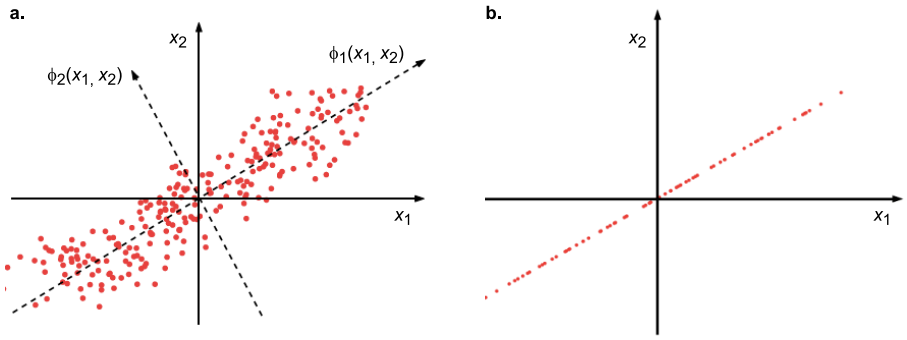
En el anexo de este módulo, hay dos enlaces a vídeos que explican en detalle esta técnica.

El primer componente principal es la combinación lineal de las dimensiones originales que tiene la máxima varianza. La *n*-ésima componente principal es la combinación lineal con la máxima varianza y con la restricción de que debe ser ortogonal en los $n - 1$ primeros componentes principales.

La idea de los PCA se ilustra en la figura 7. Los ejes x_1 y x_2 representan la covarianza entre dos píxeles. $\varnothing^1(x_1, x_2)$ corresponde a la dirección de la máxima varianza y, por lo tanto, se elige como primer componente principal. En el caso de dos dimensiones, el segundo componente principal, $\varnothing^2(x_1, x_2)$, se determina unívocamente por las restricciones de ortogonalidad. En espacios de más dimensiones, la selección de componentes seguiría, guiada por las varian-

zas de las proyecciones. Si se quisiera reducir el espacio a una sola dimensión, entonces pondríamos todos los puntos en la línea formada por el eje $\varnothing^1(x_1, x_2)$ imponiendo $\varnothing^2(x_1, x_2) = 0$ en este nuevo espacio $(\varnothing^1, \varnothing^2)$, tal como muestra la figura 9.

Figura 9.

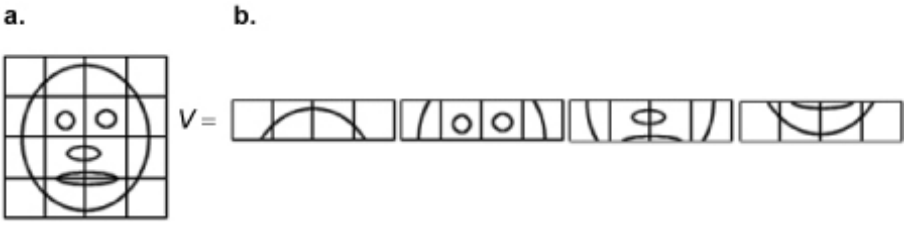


a. Representación de los dos ejes principales; b. reducción a una sola dimensión

Para encontrar estos nuevos ejes, usaremos la técnica de la **transformación de Karhunen-Loève**. No obstante, para usar esta técnica, debemos suponer que la media de los datos sea cero. Esta imposición no es ningún inconveniente, ya que lo único que nos exige es que inicialmente restemos la media de las caras a todas las caras.

Supongamos que las imágenes de las caras tienen $m \times n$ píxeles. Entonces, describimos cada imagen como un vector v de $N = m \cdot n$ elementos, simplemente concatenando las columnas de la imagen para formar el vector. v_i representa el valor de la posición i-ésima del vector v. En la figura 10, mostramos una cara y el vector transpuesto que se genera (normalmente, los vectores se representan de forma vertical).

Figura 10.



a. Imagen de 16 × 16 píxeles; b. vector transpuesto que representa la imagen

También suponemos que tenemos M imágenes diferentes de la misma cara o persona. La cara j-ésima se describe por el vector v^j. La posición i-ésima del vector v^j se describe con el símbolo v_i^j. Lo primero que debemos hacer es volver a calcular los vectores de las caras tal que la media de los nuevos vectores x^j sea cero. Cada una de las posiciones del vector media se define así:

$$\overline{v_i} = \frac{1}{M}\sum_{j=1}^{M} v_i^j \qquad 4.16$$

Este vector representa la cara media, es decir, como si las imágenes fueran transparencias y las pusiéramos todas en un montón. Entonces, los vectores normalizados se obtienen sustrayendo la media:

$$x_i^j = v_i^j - \overline{v_i} \qquad 4.17$$

para toda $1 \leq i \leq N$ y para toda $1 \leq j \leq M$. La variable x_i^j representa el valor normalizado de la posición *i*-ésima de la imagen *j*-ésima.

Con este conjunto de vectores normalizados formaremos una matriz **X** de *N* filas y *M* columnas:

$$X = \begin{bmatrix} x_1^1 & \cdots & x_1^M \\ \vdots & \ddots & \vdots \\ x_N^1 & \cdots & x_N^M \end{bmatrix} \qquad 4.18$$

Deseamos estudiar la variabilidad entre los píxeles que componen el conjunto de las imágenes de la cara de una misma persona. Por ejemplo, estudiar si siempre que un píxel aumenta el color también aumenta el color de otro píxel. En el supuesto de que la imagen no sea de la misma persona, asumimos que esta variación en concreto ya será menos importante y así detectaremos que no es la misma persona. Como lo que deseamos es estudiar la variabilidad entre los datos, con la matriz **X** calcularemos la matriz de covarianza **Σ** de *N* × *N* dimensiones de la manera siguiente:

$$\sum_{[i,j]} = \frac{1}{M}\sum_{k=1}^{M} x_i^k x_j^k \qquad 4.19$$

para todo *i*, *j* de 1 a *N*.

En esta ecuación, hemos asumido que los vectores que representan las caras son verticales. Fijaos en que el cómputo de la covarianza es muy sencillo debido a que los datos se han normalizado previamente y la media ya se ha sustraído. La matriz de covarianza se puede calcular de forma vectorial:

$$\sum_{i,j} = \frac{1}{M}\sum_{k=1}^{M} x_i^k x_j^k \qquad 4.20$$

t representa la función transpuesta, es decir que genera un vector horizontal.

Una vez calculada la matriz de covarianza, vamos a calcular los vectores propios $\varnothing = [\varnothing^1, \ldots, \varnothing^N]$ y los valores propios $\lambda = [\lambda^1, \ldots, \lambda^N]$. Cada vector propio $\varnothing^i$ tiene *N* elementos. Fijaos en que los vectores propios son los nuevos ejes representados en la figura 9. Y los valores propios representan la variabilidad de los datos proyectados en el espacio de vectores propios. Los vectores propios y valores propios se ordenan tal que:

$$\lambda^1 \geq \lambda^2 \geq \ldots \geq \lambda^N \qquad 4.21$$

Es decir, que el vector propio $\varnothing^i$ corresponde al *i*-ésimo valor propio λ^i más grande.

Finalmente, la cara de la persona *p*-ésima, que en la base de datos original se representaba por *M* imágenes de $N = m \times n$ píxeles, aquí se representa con una matriz y dos vectores. La matriz de $N \times K$ elementos son los *K* primeros vectores propios $\varnothing^p = [\varnothing^{1^p}, \ldots, \varnothing^{k^p}]$. Y el primer vector está compuesto por los *K* valores propios $\lambda^p = [\lambda^{1^p}, \ldots, \lambda^{k^p}]$. Si *K* es exactamente *N*, entonces no habremos aplicado ningún tipo de reducción de los datos. Cuanto menor sea *K*, menos ocupa la representación de cada persona pero más probabilidad habrá de que el sistema de reconocimiento se equivoque. Y el segundo vector es la media de las caras usadas para generar el modelo $\bar{v}^p = [\bar{v}^{1^p}, \ldots, \bar{v}^{N^p}]$. Este vector siempre es de *N* elementos. La representación de la persona *p* quedará pues:

1) $\varnothing^p = \begin{bmatrix} \varnothing_1^{1^p} & \cdots & \varnothing_1^{k^p} \\ \vdots & \ddots & \vdots \\ \varnothing_N^{1^p} & \cdots & \varnothing_N^{k^p} \end{bmatrix}$ donde $\varnothing_i^{q^p}$ representa el elemento *i* del vector propio *q*-ésimo de la persona *p* con una compresión de *K* vectores propios, que es $K \leq M$.

2) $\lambda^p = [\lambda^{1^p}, \ldots, \lambda^{k^p}]$ donde λ^{q^p} representa el valor propio *q*-ésimo más grande de la persona *p*. El vector *q*-ésimo se relaciona con el valor propio *q*.

3) $\bar{v}^p = [\bar{v}^{1^p}, \ldots, \bar{v}^{N^p}]$ donde $\bar{v}^{i^p}$ representa la media del píxel *i*-ésimo de todas las imágenes usadas para modelar la persona *p*.

Resumen de los índices:

- *i:* Posición del elemento dentro de los vectores, $i \in [1, \ldots, N]$.
- *j*: Índice que identifica una cara de una persona en concreto, $j \in [1, \ldots, M]$.
- *p*: Índice que identifica a una persona, $p \in [1, \ldots, Z]$.
- *q*: Índice que identifica un vector propio y un valor propio, $q \in [1, \ldots, K]$.

La figura 11 muestra la cara media de la persona p, $\bar{v}^p$, a la izquierda. Y los siete primeros vectores propios $\varnothing 1^p$, …, $\varnothing 7^p$ de la persona p. Los vectores propios se suelen denominar *eigenfaces*[11] y se traducen como *eigencares*.

(11) *Eigenvector es un vector propio y face quiere decir cara.*

Figura 11. Representación de una persona a través de su media y los siete vectores propios

5. Verificación de una cara

Supongamos que tenemos una imagen nueva de una cara en formato de vector v y la representación de la persona p expresada de la forma que hemos descrito en el apartado anterior. Deseamos saber si esta imagen es de la persona p.

El proceso de verificación ejecutará los pasos siguientes:

1) Normalizar los datos. Obtenemos el vector x a partir del vector imagen v, donde hemos extraído el vector media $\bar{v}^p$ de la persona p. Por lo tanto:

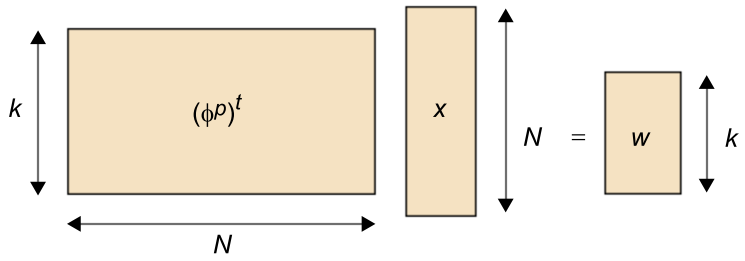
$$x = v - \bar{v}^p \qquad 4.22$$

2) Aplicamos un cambio de base a este vector x donde la nueva base son los vectores propios $\varnothing^p$ y obtenemos el mismo dato w, que se conoce como *pesos*, pero en las nuevas coordenadas. El vector w es de k elementos y se calcula así:

$$w = (\varnothing^p)^t \cdot x \qquad 4.23$$

El superíndice t representa la matriz transpuesta. Gráficamente, en la figura 12.

Figura 12



3) Calcularemos la distancia euclídea entre los pesos obtenidos w y los valores propios de la persona p, λ^p. Formalmente:

$$d^p(v) = \sqrt{\sum_{q=1}^{k} \left(\lambda^{qp} - w^q\right)^2} \qquad 4.24$$

4) Por último, si el valor obtenido es inferior a un umbral T impuesto por el administrador del sistema, consideraremos que la cara v pertenece a la persona p. Si $d^p(v) < T$ entonces devuelve SÍ. De lo contrario, devuelve NO.

6. Identificación de una cara

Tal como se ha explicado en el módulo "La biometría para la identificación de las personas", para identificar a una persona, se tienen que comparar sus datos con los datos de la base de datos. Ya dijimos que hay métodos que permiten evitar algunas comparaciones y no explorar toda la base de datos. No es el objetivo de esta asignatura hablar de temas de base de datos, por eso supondremos que realmente se exploran todos los registros de la base de datos.

Supongamos que tenemos una imagen nueva de una cara en formato de vector v que deseamos identificar y supongamos también que tenemos una base de datos de caras con Z personas y que de cada persona hemos extraído los datos explicados en el apartado "Extracción de las características". El proceso de identificación seguirá los pasos siguientes:

1) Para todas las personas p inscritas en la base de datos, calcularemos la distancia entre la cara representada por el vector v y la representación de las personas, $d^p(v)$. Obtendremos un vector de distancias:

$$D = [d^1(v), \ldots, d^p(v), \ldots, d^Z(v)] \qquad 4.25$$

2) Reordenaremos el vector D de menor a mayor generando el vector:

$$D' = [d'^1(j, v), \ldots, d'^e(i, v), \ldots, d'^Z(t, v)] \qquad 4.26$$

Mantendremos la información de cada distancia con la persona a la que correspondía, ya que:

$$d'^e(i, v) = d^i(v) \qquad 4.27$$

y la distancia es $d^i(v)$ la e-ésima menor.

3) Devolveremos la identificación de las E primeras personas con distancia menor. En algunos casos, se desea solo saber la persona que más se asemeja, entonces $E = 1$. Pero en algunas aplicaciones, es habitual no ser tanto restrictivo y E puede variar, por ejemplo $E = 10$.

Ved también

La representación de las personas se ha explicado en el apartado 5 de este módulo.

7. Generación de caras

Dada una fotografía de una persona, la podemos proyectar sobre los modelos de componentes principales y después recuperar esta nueva persona con la información de la covarianza de la otra persona. La imagen obtenida se suele denominar **imagen recuperadaR**. El proceso se denomina **generación de caras**. Este proceso se compone de dos etapas:

1) proyectamos la cara en el espacio de covarianzas; y

2) volvemos a proyectar los datos obtenidos en el espacio de las imágenes.

Este procedimiento se usa en dos situaciones diferentes. En la primera situación, deseamos generar una nueva cara con la información de la persona que queramos. Es similar a imponer los rasgos característicos de una persona a otra persona. La segunda situación aparece cuando tenemos imágenes con mucho ruido o con oclusiones parciales.

> Imaginad que hemos generado el modelo de una persona con varias fotos correctas donde los ojos siempre están abiertos y mirando a la cámara. Pero la foto que tenemos de la persona tiene los ojos cerrados. Este proceso puede servir para que podamos ver la nueva foto pero con los ojos abiertos.

Si v es el vector que representa la imagen de la persona p^1 y queremos generar una imagen recuperada de la cara R (R es un vector de N posiciones) como si fuera la persona p^2, los pasos son los siguientes:

1) Normalizar los datos. Obtenemos el vector x a partir del vector imagen v de la persona p^1 donde hemos extraído el vector media $\bar{v}^{p^1}$ de la persona p^1.

Ved también

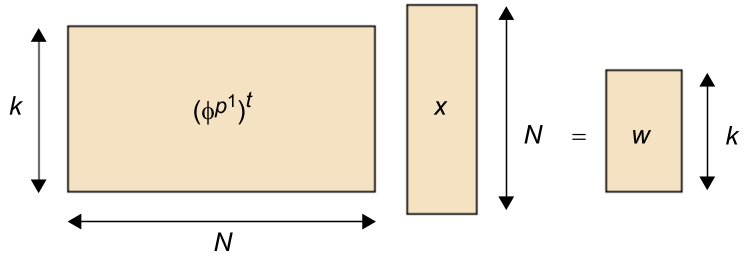
Este paso se ha explicado en apartados anteriores.

2) Aplicamos un cambio de base a este vector x, donde la nueva base son los vectores propios $\varnothing^{p^1}$. Hasta este punto, lo único que hemos hecho es convertir la imagen de la cara v el espacio de las covarianzas. Se calcula:

$$w = \left(\varnothing^{p^1}\right)^t \cdot x \qquad 4.28$$

El superíndice t representa la matriz transpuesta. Gráficamente, en la figura 13.

Figura 13



3) Ahora, lo que haremos es obtener el vector de la imagen recuperada *R* volviendo a la base original pero a través de los vectores propios de la persona p^2 y sumando la media de la persona p^2. Es decir:

$$R = \varnothing^{p2} \cdot w + \bar{v}^{p2} \qquad 4.29$$

Gráficamente, en la figura 14.

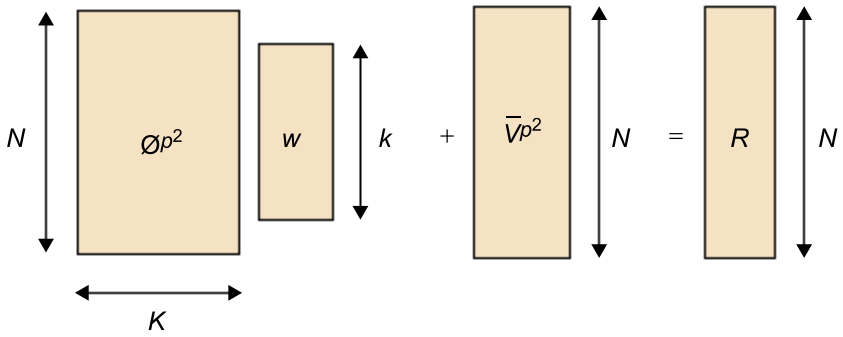
Nota

Este paso se puede llevar a cabo debido a que la matriz inversa de la matriz transpuesta de una matriz ortogonal es la propia matriz:

$$M = (M')^t \qquad 4.30$$

Los superíndices *t* y ' representan las matrices transpuesta e inversa, respectivamente.

Figura 14



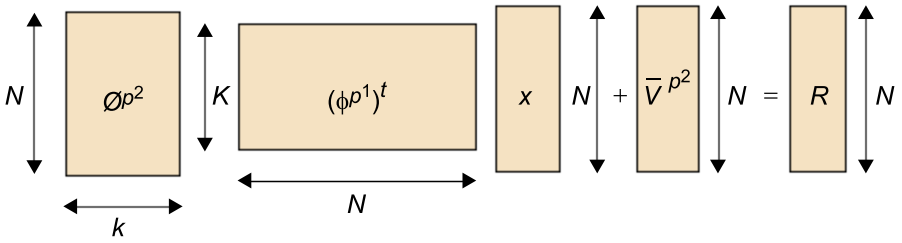
En el supuesto de que $p^1 = p^2$ y cuando no se ha llevado a cabo reducción de vectores propios, *k* = *N*, entonces el resultado final es exactamente la imagen original. En el supuesto de que *k* sea menor que *N*, entonces hay pérdida de información, puesto que la base de los vectores propios es de dimensión inferior. En este caso, la imagen original no es la misma que la recuperada. Pero este hecho, a veces es positivo, ya que sirve para extraer la información más importante, la que tiene más covarianza.

Si agrupamos todos los pasos en una sola ecuación matemática, tenemos:

$$R = \varnothing^{p1} \cdot \left(\varnothing^{p2}\right)^t \cdot x + \bar{v}^{p1} \qquad 4.31$$

Gráficamente, en la figura 15.

Figura 15



En el supuesto de que quisiéramos recuperar una imagen de una misma persona, entonces $p = p^1 = p^2$; por lo tanto:

$$R = \varnothing^p \cdot \left(\varnothing^p\right)^t \cdot x + \bar{v}^p \qquad 4.32$$

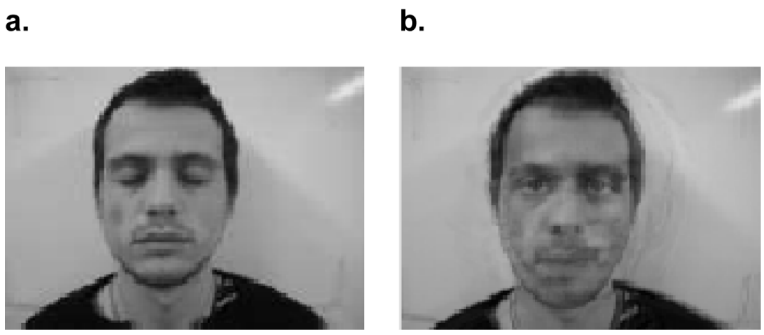
Las primeras imágenes de la figura 16 muestran la cara de una persona donde el punto de vista ha ido cambiado. La última imagen es la imagen media. Fijaos en que en todas las imágenes se pueden ver los ojos.

Figura 16.

a. Varias imágenes de una persona; b. imagen media de una persona

La imagen izquierda de la figura 17 muestra una nueva imagen. Con esta imagen, que tiene los ojos cerrados, deseamos generar una imagen con los ojos abiertos. Usando la información de las otras imágenes y aplicando el sistema de generación de imágenes obtenemos la imagen derecha de la figura 17.

Figura 17.



a. Imagen usada para la generación automática; b. imagen resultado

La figura 18 muestra las *eigenfaces* ordenadas según la magnitud del vector propio. Fijaos en cómo la primera *eigenface* resalta el movimiento de rotación de la cabeza (píxeles más claros).

Figura 18. *Eigenfaces* ordenadas según el valor propio

Resumen

En este módulo, hemos explicado algunas técnicas relacionadas con la identificación de las personas a través de sus caras. La cara es el rasgo biométrico más usado por los humanos para identificar a las personas, por eso la sociedad lo acepta extensamente. Este hecho ha influido de forma notable en la puesta en marcha de aplicaciones civiles basadas en el reconocimiento de la cara. Además, el sensor para capturar caras no es nada más que una máquina de tomar fotografías y es que todo el mundo tiene una y todo el mundo está generando fotografías con ese dispositivo. Fijaos en que este es el motivo por el que los sistemas de identificación de personas como los que usa Picassa o el iPhone tienen sentido. Si la gente no fuera generando miles de fotografías y las fuera almacenando, no habría el deseo de buscar personas en las fotografías.

Es importante destacar que la identificación de personas por las caras es un método no intrusivo y pasivo. Por lo tanto, no se pide al usuario que toque nada y así se evita el rechazo de la sociedad a tocar aparatos que otros tocan y también se evitan posibles contagios. Además, el hecho de ser pasivo se usa para identificar a personas sin su colaboración o, incluso, sin que se den cuenta.

No solo hemos hablado de identificar a personas, sino de que, gracias al análisis de los componentes principales, hemos logrado generar imágenes nuevas con caras. Esta es una aplicación útil para reducir ruido en imágenes o completar imágenes de caras parcialmente ocultas.

Finalmente, para explicar el proceso de identificación o verificación de caras nos ha hecho falta explicar el clasificador de Bayes, las características de Haar así como el análisis de componentes principales. Estas técnicas se usan ampliamente en muchas otras aplicaciones y sistemas.

Actividades

1. Describid las etapas del sistema de reconocimiento de personas por la cara. Dibujad el esquema completo.

2. Poned en funcionamiento el sistema de reconocimiento de caras de Picassa y, si disponéis de un Mac, el del iPhone.

3. ¿Cómo se llama el método que explicamos para detectar caras? ¿Por qué creéis que se llama así? ¿Creéis que es un método inteligente o es un método que usa la fuerza bruta?

4. ¿Cómo lo haríais para definir un sistema que solo detectara caras riendo? ¿O solo caras de mujeres? ¿O solo caras de una persona en concreto?

5. ¿Creéis que podríamos definir otras características de Haar? ¿Cuál es la idea básica de estas características? ¿Cómo influyen los cambios de iluminación en estas características?

6. Imaginad que deseamos establecer un clasificador de caras de mujeres y hombres. Tenemos dos opciones, una es crear dos clasificadores. El primero detecta la clase "cara mujer" y la clase "otras imágenes" y el segundo detecta la clase "cara hombre" y la clase "otras imágenes". En el supuesto de que los dos clasificadores tomen la decisión de que hay una cara en el mismo punto se tiene que tomar una decisión de qué clasificador se considera el correcto. El segundo método es crear un clasificador directamente que genere tres clases: "cara mujer", "cara hombre" y "otras imágenes". ¿Podéis definir las probabilidades de este clasificador?

7. En la definición de las probabilidades hemos supuesto que se pueden modelar usando una normal. Si fuera el caso de que no deseáramos esta modelización, ¿cómo podríamos definir estas probabilidades de forma que no fuera necesario suponer ningún tipo de modelo?

8. Detectad falsas aceptaciones y falsos rechazos en la imagen de la figura 5. En algunas falsas aceptaciones no se puede intuir la forma de la cara. ¿Por qué creéis que puede haber generado esta falsa aceptación? Y los falsos rechazos, ¿por qué creéis que no ha detectado la cara?

9. Describid un algoritmo en alto nivel que realice una normalización tanto de los tamaños como de la intensidad dados los parámetros necesarios.

Abreviaturas

PCA *principal component analysis* (análisis de los componentes principales)

Bibliografía

Jain, Anil; Flynn, Patrick; Ros, Arun (ed.) (2008). *Handbook of biometrics*. Springer.

Li y otros (2005). *Handbook of face recognition*. Springer.

Anexo. Análisis de los componentes principales, PCA

La técnica de análisis de los componentes principales, PCA o transformación de Karhunen-Loève se aplica en muchos campos diversos. Tiene una base matemática muy fuerte en la que no hemos entrado en detalle. En los vídeos siguientes, se explica la base matemática de esta técnica.

"Lec-32 Introduction to Principal Components and Analysis". *Neural Network and Applications* (http://www.youtube.com/watch?v=H0HjNuNvFVI)

Lec-33 Dimensionality reduction Using PCA". *Network and Applications* (http://www.youtube.com/watch?v=HnVYF6VQryU)