

Primeros pasos para una aplicación móvil offline de reconocimiento facial

Barraza, Sara Lía

Facultad de Ciencias Exactas y Tecnología, Universidad Nacional de Tucumán

Thuillier, Etienne

Université Technologique de Belfort-Montbéliard, Francia

Will, Adrián

*GITIA, Universidad Tecnológica Nacional, Facultad Regional Tucumán
Facultad de Ciencias Exactas y Tecnología, Universidad Nacional de Tucumán*

Rodriguez, Sebastián Alberto

GITIA, Universidad Tecnológica Nacional, Facultad Regional Tucumán

Resumen

El reconocimiento facial es una herramienta moderna con una amplia gama de aplicaciones como vigilancia, seguridad, etc. Las necesidades de seguridad y los aumentos en la capacidad de procesamiento y resolución de las cámaras disponibles, ha hecho que estos sistemas ganen un creciente interés. Sin embargo la mayoría de este tipo de programas requieren de conexión a internet para su funcionamiento, debido entre otras razones a la potencia de cálculo requerida y al tamaño de las bases de datos involucradas. El presente trabajo representa un primer paso hacia el desarrollo de un sistema de reconocimiento facial de bajo requerimiento en procesamiento, capaz de funcionar apropiadamente en un dispositivo móvil. La posibilidad de funcionamiento offline en estos casos es crítica, ya que le permitiría funcionar eficientemente incluso dentro de edificios o en lugares sin señal.

Palabras Clave Reconocimiento facial; Tensores;

SVD; TensorFaces

1 Introducción

En los últimos años la utilización del reconocimiento facial se ha extendido por todo el mundo y ha sido ampliamente utilizada en numerosas aplicaciones como sistemas de seguridad, sistemas de control de acceso a equipos o ambientes, identificación de conductas, entre otras ([1], [2], [3], [4]). Mas aún, debido a las mejoras notables en la tecnología, es posible para una computadora reconocer un rostro rápidamente, de manera que resulta práctico utilizar un sistema de este tipo para desbloquear un dispositivo móvil o identificarse en el acceso a una parte de un edificio con mayor nivel de seguridad que el resto.

Sin embargo, la gran mayoría de estos sistemas resultan costosos en precio, instalación y mantenimiento, necesitan un gran poder de cómputo y conexión a internet al momento de

ser utilizados (*picasa*¹ o *facebook*²). Adicionalmente, sistemas como *Visidon AppLock*³ para el sistema operativo *Android* tienen severas limitaciones, tanto en la cantidad de rostros a reconocer como en la capacidad para funcionar.

Para numerosas aplicaciones es suficiente este tipo de servicios, dado que el dispositivo en que se utilizará el programa está en una ubicación fija y sin problemas de acceso a internet o procesamiento (control de acceso a un aeropuerto, o a cuartos específicos de un edificio). Sin embargo en otros casos resulta necesario utilizar un sistema de este tipo en condiciones especiales como "indoors" (en un edificio) o en lugares remotos o subterráneos, lo que generalmente significa sin conexión a internet o con una conexión de muy mala calidad. Esto se complementa con el problema, para el caso específico de desbloqueo de dispositivos móviles por ejemplo, en que el reconocimiento debe ser realizado en poco tiempo y en un aparato con bajo poder de procesamiento.

Entre otras herramientas posibles, existen investigaciones anteriores que han abordado este problema mediante el uso de tensores de orden superior [1], [4], [5]. Estas herramientas requieren en general una gran cantidad de procesamiento a la hora de entrenar, dado que se basan en una descomposición de matrices que requiere procesamiento (*HOSVD, Higher Order Singular Value Decomposition*). Sin embargo en tiempo de ejecución sólo es una proyección lineal y otros procedimientos de bajo requerimiento, lo que los hace ideales para este tipo de aplicaciones. Por otro lado, en muchas de estas investigaciones y otras similares, la experimentación, y especialmente el preprocesamiento de los datos, resulta difícil de seguir por falta de claridad y de especificación del procedimiento utilizado.

En este trabajo proponemos entonces un sistema de reconocimiento facial que dé respuesta a estos problemas, que esté preparado para funcionar en dispositivos móviles, con bajo requerimiento de memoria y procesamiento, una cámara de relativamente baja resolución, y de manera autosuficiente (sin necesidad de

conexión a internet). El sistema está basado en la descomposición HOSVD de una matriz formada por imágenes de 6 sujetos de prueba tomadas con una cámara digital en condiciones controladas, y fue validado utilizando fotos de los mismos y otros sujetos tomadas en otras condiciones incluyendo cámaras de teléfono inteligente, otras condiciones de iluminación, y otros incluyendo cambios en peinado. El procedimiento utilizado está exhaustivamente descrito en la sección 4.

Como limitaciones de la tecnología desarrollada, se encontró la necesidad de que el entrenamiento se realice en una PC o similar con alto poder de procesamiento (debido a la descomposición algebraica de matrices necesaria), que el conjunto de personas a identificar debe ser pequeño, y que las imágenes utilizadas en el conjunto de entrenamiento deben ser de buena calidad y tomadas en las mismas condiciones. Estas limitaciones surgen de los experimentos realizados, y si bien limitan la aplicabilidad del sistema, entendemos que resulta suficientemente práctico y flexible para ser utilizados en aplicaciones reales.

El trabajo está dividido en 4 secciones. En la primera sección se presenta el estado del arte. En la segunda sección se explica la parte matemática de tensores necesaria. La sección 3 presenta los datos y el procedimiento y preprocesamiento realizado a los mismos. La sección 4 presenta el algoritmo utilizado, y en la sección 5 se muestran los resultados obtenidos.

2 Trabajos relacionados

2.1 Sistemas de reconocimiento facial

Durante la década anterior, el reconocimiento facial ha sido considerado y aplicado en diversas áreas: detección de rostros en fotografía para evitar somnolencia en transportes⁴, estimación de edad y género en una multitud, análisis de clientes, reconocimiento facial para control de acceso, identificación de visitantes, ingreso en aplicaciones online, etc.

Las aplicaciones de seguridad proponen sistemas con resultados positivos casi inmediatos, pero la mayoría de estos sistemas resultan costosos tanto en el proceso de desarrollo como de instalación y mantenimiento. Presen-

¹<http://picasa.google.com/>

²<https://www.facebook.com/>

³<https://play.google.com/store/apps/details?id=visidon.AppLock&hl=en>

⁴<http://www.tcit.com.tw/?language=en>

tan resultados adecuados para las aplicaciones a las que están destinadas. Una característica común a la mayoría de estos sistemas es la necesidad de una conexión Internet debido al gran volumen de datos involucrado y el procesamiento requerido.

Luego encontramos los programas que realizan la detección de rostro en imágenes, y como se indica, solo permiten detectar la posición y límites de un rostro en una imagen, sin permitir identificar a la persona, común en cámaras fotográficas modernas ([6]). Existen también sistemas de reconocimiento facial como por ejemplo *facebook* que adquirió el servicio *face.com*, *Google* con el servicio *rekognition*⁵, *Apple* que compró *Polar rose*⁶ que desarrolló el servicio *recognizr*⁷.

Por último, se encuentra el programa *Visidon AppLock* de *Android* que permite al usuario desbloquear su teléfono inteligente con una aplicación de reconocimiento facial. El sistema funciona en dispositivos móviles sin utilizar la conexión a Internet para su funcionamiento. Se basa en una comparación de una imagen del usuario, con una o más fotos archivadas y una comparación de las características faciales. Si bien presenta problemas al utilizar el programa con bajo nivel de luz, con personas parecidas, o con cambios en el vestuario o el peinado.

2.2 Trabajos anteriores

Se realizaron trabajos que proponen tensores para reconocimiento facial. En este trabajo dos artículos importantes se utilizarán como soporte para nuestro algoritmo. Vasilescu y Terzopoulos [4] han propuesto utilizar tensor SVD para reconocer rostros. Su trabajo incluye una comparación entre PCA y TensorFaces y muestran buenos resultados al utilizar tensores. La segunda obra propuesta por Rana [1] desarrolla diferentes enfoques, en los cuales todos utilizan un algoritmo basado en SVD-tensor. En este trabajo, se realizaba un estudio de diferentes algoritmos - todos basados

⁵<http://rekognition.com/>

⁶http://news.cnet.com/8301-13579_3-20017018-37.html

⁷http://readwrite.com/2010/02/24/recognizr_facial_recognition_coming_to_android_phones

en tensores - y se extraían conclusiones sobre los algoritmos probados, con la intención de identificar el que mejores resultados presentara tanto en el reconocimiento como en el tiempo empleado.

Sin embargo, en los trabajos mencionados, aunque la matemática utilizada no presentar errores visibles, los experimentos y los resultados mostraron una falta de descripciones metodológicas, lo que resulta en experimentaciones difíciles de reproducir. Ambos trabajos, utilizan bases de datos descargados de internet: Weizmann⁸, PEAL⁹ y YaleB¹⁰, y estas producen problemas a la hora de intentar utilizarlas. Por ejemplo, la base de datos Weizmann contiene más de mil imágenes que presentan más o menos cuarenta fotos para cada persona. Esto implica un proceso complejo para agregar personas y que las condiciones utilizadas estarán lejos de las reales de utilización, lo que lo vuelve poco práctico.

3 Álgebra de tensores

Se describen en esta sección los preliminares básicos sobre Álgebra de Tensores y la descomposición HOSVD

3.1 Descripción de tensores

Un tensor es considerado una matriz multidimensional o matriz de orden N , por ejemplo, un vector es un tensor de orden 1 y una matriz es un tensor de orden 2. Un tensor A de orden tres es una matriz de tres dimensiones, y está denotado por $A \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ donde cada dimensión de la matriz es llamada *modo* del tensor. Este puede ser visualizado como un bloque en tres direcciones, como lo muestra la figura 1

Sus elementos están denotados por a_{ijk} donde $0 < i < I_1$, $0 < j < I_2$, $0 < k < I_3$ y se obtienen fijando los índices i, j y k en cada modo. Si en una matriz se fijan uno de sus índices se obtiene los vectores filas (primer índice fijo) y columnas (segundo índice fijo). El análogo en tensores se llaman *fiber*. En

⁸<http://www.wisdom.weizmann.ac.il/~vision/FaceBase/>

⁹<http://www.jdl.ac.cn/peal/>

¹⁰<http://cvc.yale.edu/projects/yalefacesB/yalefacesB.html>

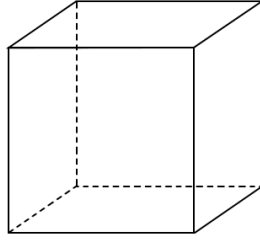


Figura 1: Representación de un tensor de orden tres.

un tensor de orden tres $A \in \mathbb{R}^{I_1 \times I_2 \times I_3}$, las fiber son llamadas columnas, filas y tubos, las cuales son denotadas por $A(:, i_2, i_3) = a_{:i_2 i_3}$, $A(i_1, :, i_3) = a_{i_1 : i_3}$ y $A(i_1, i_2, :) = a_{i_1 i_2 :}$ respectivamente.

Las fiber en un tensor de orden tres se pueden observar en la figura 2

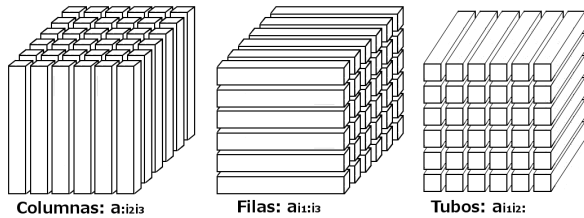


Figura 2: Vectores *Fiber* de un tensor de orden tres.

De manera análoga, las matrices de un tensor de orden tres son llamadas *slices*, y están denotadas por $A(i_1, :, :)$ (sección horizontal), $A(:, i_2, :)$ (sección lateral) y $A(:, :, i_3)$ (sección frontal). En la figura 3, se muestran las secciones de un tensor de orden tres.

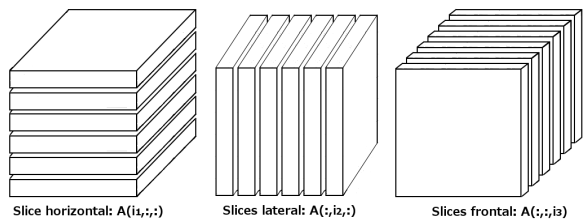


Figura 3: Matrices *slices* de un tensor de orden tres.

Se definen ahora, dos conceptos de tensores importantes para el desarrollo del algoritmo.

Matrización de un tensor: El concepto de matrización o *tensor unfolding* se refiere al proceso de transformar un tensor en una matriz. Para un tensor de orden tres este proceso se muestra en la figura 4

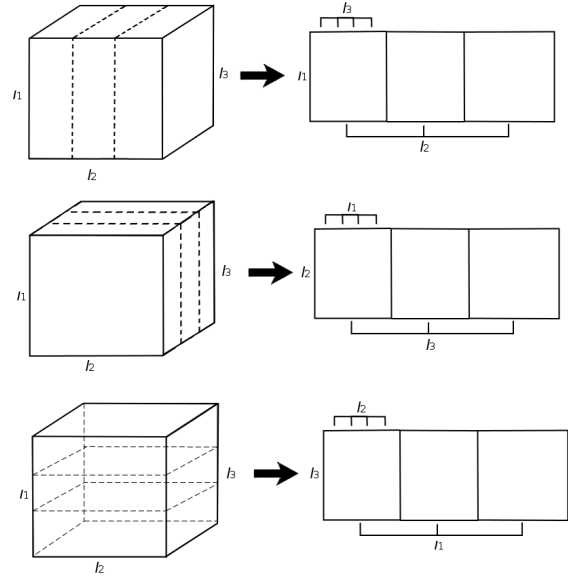


Figura 4: Proceso de matrices *unfolding*.

Producto *n*-modo: El producto *n*-modo de un tensor $A \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_n \times \dots \times I_N}$ y una matriz $U \in \mathbb{R}^{J_n \times I_n}$ denotado por $A \times_n U$, es un tensor de dimensión $I_1 \times I_2 \times \dots \times J_n \times \dots \times I_N$ definido componente a componente por

$$(A \times_n U) = \sum_{i_n} a_{i_1 i_2 \dots i_n \dots i_N} u_{j_n i_n}$$

3.2 Descomposición de tensores

Aquí, se propone describir la descomposición en valores singulares para tensores, denotada por HOSVD (Higher Order Singular Value Decomposition) [7] [5] [8] la cual es una generalización de la descomposición SVD para matrices. Para facilitar la comparación vamos a presentar la SVD matricial en la notación inducida por el producto *n*-modo entre un tensor y una matriz.

SVD de matrices:

Toda matriz $A \in \mathbb{R}^{I_1 \times I_2}$ puede ser escrita como el producto

$$A = USV^T = S \times_1 U \times_2 V^T = S \times_1 U^{(1)} \times_2 U^{(2)}$$

Donde

$U^{(1)} = U$ es una matriz unitaria de orden $I_1 \times I_1$

$U^{(2)} = V^T$ es una matriz unitaria de orden $I_2 \times I_2$

S es una matriz de orden $I_1 \times I_2$ con las siguientes propiedades:

$$S = \text{diag}(\sigma_1, \sigma_2, \dots, \sigma_{\min(I_1, I_2)})$$

$$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_{\min(I_1, I_2)}$$

Los σ_i son los valores singulares de A .

SVD de tensores (HOSVD):

Todo tensor $A \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ puede ser escrito como el producto

$$A = S \times_1 U^{(1)} \times_2 U^{(2)} \times_3 U^{(3)}$$

donde

$$\begin{aligned} U^{(1)} &\in \mathbb{R}^{I_1 \times I_1} \\ U^{(2)} &\in \mathbb{R}^{I_2 \times I_2} \\ U^{(3)} &\in \mathbb{R}^{I_3 \times I_3} \end{aligned}$$

son matrices ortogonales.

$S \in \mathbb{R}^{I_1 \times I_2 \times I_3}$ llamado tensor *core*, tiene la propiedad de ortogonalidad: cualquiera dos slices de S son ortogonales en el sentido del producto escalar:

$$\begin{aligned} \langle S(i, :, :), S(j, :, :) \rangle &= \langle S(:, i, :), S(:, j, :) \rangle \\ &= \langle S(:, :, i), S(:, :, j) \rangle \\ &= 0 \text{ para } i \neq j \end{aligned}$$

Para el modo-1 se define los valores singulares como:

$$\sigma_i^{(1)} = \|S(i, :, :)\|_f \quad i = 1, \dots, I_1$$

Y están ordenados como sigue:

$$\sigma_1^{(1)} \geq \sigma_2^{(1)} \geq \dots \sigma_{I_1}^{(1)}$$

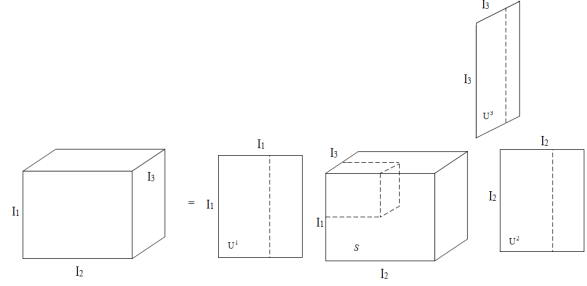


Figura 5: HOSVD de un tensor de orden tres.

Los valores singulares en los otros modos y su ordenación se definen de manera análoga. Para un tensor de orden tres, un esquema de tal descomposición se puede visualizar en la figura 5.

Esto es claramente una generalización de la descomposición SVD para matrices, si se restringe el teorema a tensores de orden dos.

El algoritmo para calcular la descomposición es:

- 1) Calcular las matrices U_n , mediante el SVD de las matrices T_n , obtenidas por matrización del tensor en cada modo.
- 2) Calcular el tensor S como:

$$S = A \times_1 U^{(1)T} \times_2 U^{(2)T} \times_3 U^{(3)T}$$

4 Datos Utilizados

Se describe en esta sección con detalle, el proceso de generación de las imágenes utilizadas en este trabajo. Se describen tanto las cámaras utilizadas como el tipo y características de las imágenes utilizadas para el entrenamiento y el proceso utilizado en las imágenes de validación.

4.1 Sesión de fotos

Intentos previos de los autores por utilizar una base de datos como YaleB, Weizmann y PEAL de manera similar a lo encontrado en los trabajos citados, resultaron infructuosos o de resultados poco reproducibles, debido principalmente a que el procesamiento de las imágenes no está correctamente descrito. Mas aún los tensores entrenados con estas bases

de datos y completados con fotografías propias, complican el funcionamiento del algoritmo, debido a que las imágenes no están tomadas en las mismas condiciones, dando malos resultados. Por esta razón, decidimos formar una base de datos propia, que nos permite un control sobre el proceso de preprocesamiento.

Para el presente trabajo se tomaron imágenes de 6 voluntarios bajo diferentes condiciones controladas de iluminación, fondo, y expresión, según el siguiente detalle:

- **Foto A1:** con expresión neutra, dos flash frontales
- **Foto A2:** con expresión sonriente, dos flash frontales
- **Foto B1:** con expresión neutra, flash a la derecha
- **Foto B2:** con expresión sonriente, flash a la derecha
- **Foto C1:** con expresión neutra, flash a la izquierda
- **Foto C2:** con expresión sonriente, flash a la izquierda

Las fotos fueron tomadas teniendo en cuenta lo siguiente:

- El fondo de la foto es neutral, sin componentes ruidosos, como formas o sombras.
- La distancia de la cámara-persona es fija en 2 mts
- La distancia focal de la cámara es tal que el ángulo formado por las lentes muestra la cara de la persona en toda la superficie de la imagen.
- Las imágenes tomadas fueron encuadradas de manera tal que la parte superior del cabello toca el borde superior de la imagen y la parte inferior del menton toca el borde inferior de la imagen.

El objetivo del proceso, es que el rostro ocupa más o menos el 70% de la imagen total. Esta configuración se utiliza con el fin de evitar que algunos factores como el fondo o la ropa de el sujeto afecten el análisis.

Pruebas preliminares demostraron que el orden de las fotos expuesto no es crítico (pueden intercambiarse poniendo por ejemplo las fotos A2 antes de las A1 con idénticos resultados), pero resulta crítico cambiar en una persona algunas de las condiciones descritas.

4.2 Preprocesamiento

Una vez tomada la serie de fotos de la base de entrenamiento, fueron tratadas con el fin de poder ser utilizadas por el algoritmo. Se utilizó para ello software de edición de fotos (*The Gimp*¹¹ y *Photoshop*¹² para el remuestreo). Cada una de las fotos de la serie se transformaron a componentes blanco y negro reduciendo la dimensión del problema. Luego, cada foto se redujo a un tamaño de 200x200 píxeles utilizando la función resampling de Photoshop. Este paso se realiza para conseguir un total de 40000 píxeles de cada foto lo que resulta en un tensor de tamaño que puede ser descompuesto en una PC de escritorio ordinaria utilizando Matlab, y al mismo tiempo con la precisión suficiente para efectuar el reconocimiento. Se obtuvo entonces una base de datos formada por seis fotografías de seis personas, un total de 36 imágenes, donde las mismas están ordenadas según la descripción efectuada arriba. Se puede ver una muestra de las 6 fotos utilizadas para algunos de los sujetos en (figura 6).



Figura 6: Parte de la base de datos utilizada.

El preprocesamiento para las fotos de validación sigue casi el mismo proceso que para las imágenes de la base de datos: se toma la foto, se transforma a blanco y negro, y luego se modificaría para que coincida en formato y características con las imágenes de la base de entrenamiento.

Toma de fotos para uso el reconocimiento

- 1) Tomar una foto retrato de la persona con el dispositivo móvil

¹¹<http://www.gimp.org/>

¹²<http://www.photoshop.com/>

- 2) Redimensionar o ampliar la foto al centro y aproximar el rostro de la persona
- 3) Establecer la foto a blanco y negro
- 4) Cambiar el tamaño de la foto para una imagen 200x200px
- 5) Probar esta imagen con el algoritmo

A.3 Especificaciones de cámaras fotográficas utilizadas

Todas las fotos de la base de datos fueron tomadas con una cámara Nikon D7000, con lente 18-105, distancia focal de 70 mm (equivalente a 105 mm si fuera una cámara tradicional de rollos de 35 mm), y dos flash colocados a ambos lados de la cámara, un Nikon SB700 y un Nissin Di622. Además se tomaron en formato jpg, bajo las mismas condiciones explicadas antes.

Se tomaron además fotos con la cámara digital incorporada de un teléfono inteligente Samsung Galaxy SII (I9100) con una distancia focal de 4,03 mm (equivalente a 30 mm en una cámara tradicional de rollos de 35 mm), un sensor de tamaño 1/3.2" con luz natural y flash incorporado de relleno, para validación del sistema.

5 Algoritmo y Experimentación

En esta sección se describe el ordenamiento como Tensor de las imágenes utilizadas, y el algoritmo utilizado.

Según lo descrito en la sección anterior, se utilizó una base de datos de 36 imágenes, tomadas a seis personas en dos expresiones diferentes y tres condiciones de iluminación, tomándose así seis imágenes por persona. Al igual que en [4], se reordenan cada imagen como un vector $1 \times 40,000$, y se las ordena de manera que las imágenes estén representadas mediante un tensor $T \in \mathbb{R}^{6 \times 6 \times 40000}$ donde los modos corresponden a modo-*persona*, modo-*exp.ilum* y modo-*pixel* respectivamente. O sea, es una matriz $6 \times 6 \times 40,000$, donde cada imagen está ordenada como un "tubo" (fila y columna constante, ver sección 3), cada slice horizontal del tensor (modo 1 constante) contiene las distintas fotos de una misma persona, y cada slice lateral del tensor (modo 2 constante), contiene las imágenes de todas las personas en una misma condición de iluminación y expresión.

Se decidió considerar en el modo-2 las seis imágenes de cada persona, ordenadas de acuerdo a la iluminación y expresión, y no de manera independiente, para trabajar con un tensor de orden tres, ya que las pruebas realizadas con tensores de orden superior separando este caso, no produjeron mejoras notables.

Al realizar la descomposición HOSVD se obtiene:

$$T = S \times_1 U_{persona} \times_2 U_{exp.ilum} \times_3 U_{pixel}$$

donde $S \in \mathbb{R}^{6 \times 6 \times 40000}$ es el llamado tensor *core* que interactúa con las matrices $U_{persona}$, $U_{exp.ilum}$ y U_{pixel} , correspondientes a los modos del tensor T , para crear una imagen.

Las columnas de $U_{persona}$, $U_{exp.ilum}$ y U_{pixel} definen los subespacios vectoriales persona, exp.ilum y píxel, respectivamente.

Las filas de la matriz $U_{persona}$ corresponden a una persona en particular, es decir, la i -ésima fila representa a la i -ésima persona, independiente de la expresión e iluminación. Lo mismo ocurre con las filas de $U_{expresion}$, donde cada una ellas corresponden a una expresión en particular.

La descomposición HOSVD puede ser considerada de diferentes maneras, de acuerdo a su utilidad. Aquí, se utiliza la siguiente expresión:

$$T = B \times_1 U_{persona}$$

donde $B = S \times_2 U_{exp.ilum} \times_3 U_{pixel}$

Como lo muestra la figura 7

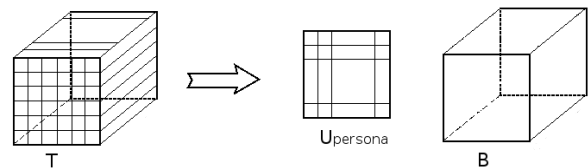


Figura 7: Descomposición HOSVD.

Si se fija una persona p y una expresión e en particular, se obtiene el vector:

$$T(p, e, :) = U(p, :) \times_1 B(:, e, :)$$

Si llamamos $t_p = T(p, e, :)$, $u_p = U(p, :)$ y $B_e = B(:, e, :)$ la ecuación se transforma en:

$$t_p = u_p B_e$$

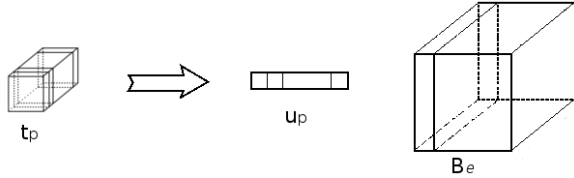


Figura 8: Vector imagen.

Esto se puede interpretar como sigue: *el vector t_p representa la imagen de la persona p en la expresión e ; las columnas de la matriz B_e forman una base para la expresión e y el vector u_p contiene las coordenadas de la imagen t_p , en dicha base.*

Por lo tanto, el vector imagen de la persona p en la expresión e está dado por:

$$I_{pe} = u_p B_e$$

Así, las coordenadas de la imagen en dicha expresión, están representadas por el vector:

$$u_p = I_{pe} \text{pinv}(B_e)$$

Ahora si se considera la imagen de una persona desconocida I_t , sus coordenadas en una cierta expresión e estarán dadas por

$$d_t = I_t \text{pinv}(B_e)$$

Luego se compara este vector, con los vectores fila $u_p = U(p, :)$ de la matriz U_p , el vector u_p que minimiza la expresión

$$\|d_t - u_p\|$$

entre todas las expresiones, identifica la imagen desconocida como la persona p .

A continuación, se describe el algoritmo:

Algoritmo: SVD-multilineal

Input: Imagen de prueba I_t

- 1) Calcular el tensor B
- 2) Dada la imagen de prueba I_t , calcular sus coordenadas en una expresión e como:
$$d_t = \text{Pinv}(B_e) * I_t$$
- 3) Evaluar $\min \|d_t - u_p\|$ donde u_p son las filas de la matriz U_{persona}
- 4) El vector u_p que minimiza la expresión $\|d_t - u_p\|$ identifica a la persona desconocida como la persona p

Se realizaron pruebas con fotos de personas desconocidas, y con fotos de personas de la base de datos, en diferentes condiciones, para optimizar el algoritmo. Los resultados obtenidos superaron nuestras expectativas y se muestran en la sección 4.

6 Resultados Obtenidos

6.1 Positivos

Las primeras pruebas se realizaron con las fotos de testeo, y con fotos de personas desconocidas. En todos los casos, los resultados fueron óptimos, aún cuando las fotos fueron tomadas en diferentes condiciones de iluminación o con una definición muy baja, como se observa en la figura 9.

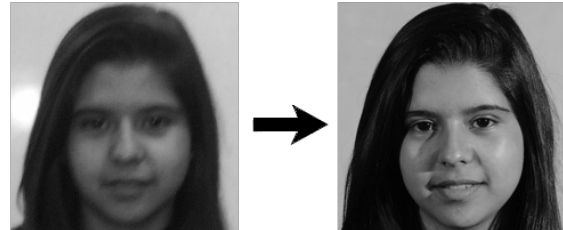


Figura 9: Resultados de reconocimiento con una foto borrosa. A la izquierda foto de prueba, a la derecha foto reconocida por el algoritmo.

6.2 Parámetros controlados

Luego se consideró, para personas de la base de datos, imágenes tomadas anteriormente en condiciones muy diferentes, como distancia focal, iluminación, condiciones físicas de la persona (barba, cabello, ropa, etc.). En este caso se pudo observar que dichas condiciones afectan, en algunos casos, al reconocimiento mientras que en otros la persona fue reconocida.

Considere la figura 10 que representa algunos casos donde el reconocimiento se obtiene sin controlar los diferentes parámetros.

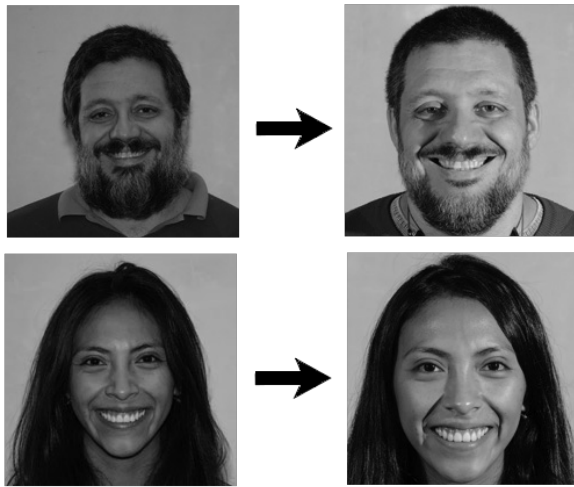


Figura 10: Resultados de reconocimiento con fotos que tienen ruido.

En otro caso fue obligatorio modificar la imagen, debido a que ésta abarca características (hombros, cuello, espacio físico) que generan ruido en el análisis. Para hacer esta modificación, se cambió el parámetro de distancia focal (es decir, un zoom sobre el rostro sin modificar el tamaño de la foto, ver figura 11)

Después de esta modificación el reconocimiento llegó a un resultado positivo (figura 12).

6.3 Resultados negativos

Al realizar las primeras pruebas, con las fotos de testeo de las personas de la base de datos, se obtuvo, que para una de ellas, el algoritmo falló, dando como resultado una compa-



Figura 11: Modificación digital de la foto testeo (zoom).

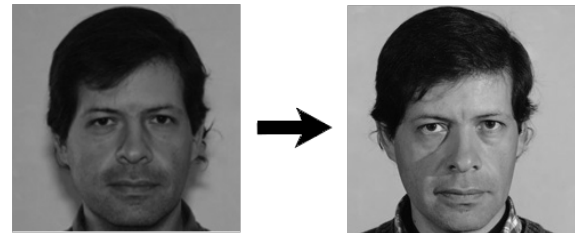


Figura 12: Resultados de reconocimiento con una foto modificada.

ración con personas parecidas. (figura 13). Observemos que en la foto, el rostro de la persona está desviado. En otra imagen similar donde la persona está correctamente orientada, el reconocimiento es positivo.

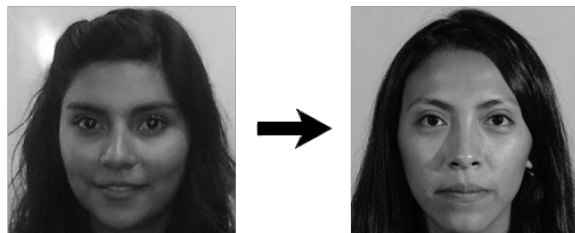


Figura 13: Resultados de reconocimiento erróneo.

6.4 Resumen

Las pruebas se realizaron con 28 fotos, de las cuales 21 corresponden a imágenes de personas de la base de datos y 7 a personas desconocidas. De estas 28 fotos, 23 de los casos die-

ron resultado positivo (82%). Los resultados negativos se obtuvieron de 5 imágenes de personas de la base de datos, de las cuales se modificaron tres de ellas, que luego de una prueba fueron reconocidas. Es decir, que el algoritmo arrojó un resultado positivo en un 93% de los casos.

Además, de las 21 fotos de las personas conocidas, sólo 16 de estas fueron reconocidas, más las 3 imágenes modificadas nos da un 90% de resultado positivo con personas de la base de datos.

En el caso de las personas desconocidas, el 100% de los casos fueron positivos.

La figura 14 muestra la tabla de resultados.

Fotos \ Resultados	Pruebas	Conocidas	Desconocidas
Positivos	82 %	76.2 %	100 %
Controlados	11 %	14.3 %	---
Negativos	7 %	9.5 %	---

Figura 14: Resultados de reconocimiento.

7 Conclusiones

En este trabajo presentamos un método para reconocimiento facial mediante un modelo basado en tensores, diseñado para ser ejecutado en dispositivos móviles. Como características principales, es bastante robusto a cambios en apariencia, resolución, cambios en la vestimenta o peinado, etc. El uso de más de una persona y una cámara de buena calidad en el entrenamiento han producido resultados muy superiores a los obtenidos entrenando con fotos de calidad normal. En las fotos probadas con baja iluminación o menor calidad, el sistema funcionó igualmente de manera correcta.

El algoritmo desarrollado evalúa la proximidad lineal de una imagen con un conjunto de imágenes presentes en la base de datos, que dan un resultado positivo (el reconocimiento o no) en más de un 93% de los casos. Se encontró sin embargo que resulta crítica la formación de un conjunto de fotos de entrenamiento de alta calidad y de poses similares. La introducción de fotos de mala calidad, o con variaciones en la posición o expresión en el conjunto de entrenamiento, disminuye sensiblemente el nivel de los resultados obtenidos. También

es sensible al encuadre de la foto, lo que hace imprescindible la incorporación de un sistema para reconocer el rostro y adecuar el encuadre, como trabajo futuro.

Los trabajos futuros incluyen la aplicación de este proceso y algoritmo en dispositivos inteligentes, así como la mejora del proceso de entrenamiento, que según la experimentación realizada debe ser efectuada en una PC debido al alto poder de cómputo requerido. También la mejora del reconocimiento en bajo nivel de iluminación, problema común en este tipo de programas, sería un avance importante. Entre las posibilidades del sistema a futuro, es perfectamente posible entrenarlo para reconocer otras personas, permitiendo que estas (familiares o similar) tengan acceso al sistema, o que personas de características faciales similares sean distinguidas. Estas posibilidades son complicadas de realizar en otros de los sistemas actualmente en uso.

Agradecimientos

Agradecemos al Ingeniero Jorge Bustos de la F.A.C.E.T., U.N.T., por su trabajo profesional en la toma y preprocesamiento de las imágenes

Referencias

- [1] Santu Rana, Wanquan Liu, Mihai Lazarescu, and Svetha Venkatesh. A unified tensor framework for face recognition. volume 42, pages 2850 – 2862, 2009.
- [2] Santu Rana, Wanquan Liu, Mihai Lazarescu, and Svetha Venkatesh. Efficient tensor based face recognition. In *ICPR*, pages 1–4. IEEE, 2008.
- [3] Matthew Turk and Alex Pentland. Eigenfaces for recognition. *J. Cognitive Neuroscience*, 3(1):71–86, jan 1991.
- [4] M.A.O. Vasilescu and D. Terzopoulos. Multilinear image analysis for facial recognition. 2:511–514 vol.2, 2002.
- [5] Lars Eldén. *Matrix Methods in Data Mining and Pattern Recognition*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2007.

- [6] Paul Viola and Michael Jones. Robust real-time object detection. In *International Journal of Computer Vision*, 2001.
- [7] Lieven De Lathauwer, Bart De Moor, and Joos Vandewalle. A multilinear singular value decomposition. *SIAM J. Matrix Anal. Appl.*, 21:1253–1278, 2000.
- [8] Tamara G. Kolda and Brett W. Bader. Tensor decompositions and applications. *SIAM REVIEW*, 51(3):455–500, 2009.