DESARROLLO DE UN PROCESO DE AUTENTICACIÓN FACIAL EN UN SISTEMA ANDROID UTILIZANDO EL ALGORITMO LDA (ANÁLISIS DE DISCRIMINACIÓN LINEAL)

Francisco Emiliano Aguayo Serrano
Universidad Autónoma de Querétaro
microstudio.aguayo@gmail.com

Jesús Carlos Pedraza Ortega
Universidad Autónoma de Querétaro
caryoko@yahoo.com

Edgar Alejandro Rivas Araiza
Universidad Autónoma de Querétaro
erivas@uaq.mx

José Erik Rivas Araiza
Universidad Autónoma de Querétaro
jerivas@uaq.edu.mx

Resumen

En este trabajo de investigación se desarrolló un proceso de autenticación facial implementando primero los algoritmos PCA y LDA en una PC, evaluando sus respectivos desempeños en tiempo y nivel de autenticación con bases de datos públicas y posteriormente implementando el algoritmo LDA en una aplicación Android utilizando una base de datos propia donde las imágenes están bajo diferentes condiciones de iluminación, distancia, pose de la persona, fondo de la imagen, etc. Todo esto con el fin de seguir contribuyendo a los sistemas de reconocimiento y autenticación facial ya que esta área ha ido creciendo a lo largo de estas tres últimas décadas y se aplica en diversas áreas como la seguridad, la interacción entre hombre y máquina, video juegos, etc. Este proceso de

reconocimiento facial se divide a su vez en reconocimiento 1:n (reconocimiento facial) y reconocimiento 1:1 (autenticación facial).

Palabras Claves: Autenticación, cara, LDA, PCA, visión por computadora.

Abstract

In this work, the facial authentication process was developed by first implementing the PCA and LDA algorithms in a PC, evaluating their performance in time and level of authentication in public databases and later implementing the algorithm LDA in an Android application using an own database where the images are under different conditions of illumination, distance, pose of the person, background of the image, etc. Everything in order to continue contributing to facial recognition and authentication systems as this area has been growing throughout these three decades and is applied in various areas such as security, the interaction between man and machine, video games, etc. This facial recognition process is once divided into 1: n recognition (facial recognition) and 1: 1 recognition (facial authentication).

Keywords**:** *Authentication, Computer Vision, Recognition, LDA, PCA.*

1. Introducción

La biometría es una disciplina que estudia la identificación de una persona en base a sus características como huellas, la forma del rostro y su contorno, la voz, el iris de los ojos, cicatrices, etc. [Brumnik, 2011]. La biometría tiene muchas aplicaciones como la identificación de delincuentes, controles de acceso automático, entre muchas otras como se muestra en [Duró, 2001].

En los últimos años se ha incrementado el uso de dispositivos y tecnologías que le permiten a las personas llevar a cabo sus labores cotidianas como transacciones en la banca, acceso a un sistema mediante un usuario y contraseña e incluso con autenticación con huella o rostro, tal como se muestra en el trabajo presentado por [Hernández, 2010], donde se explica la importancia de contar con un buen sistema de autenticación facial, una metodología de trabajo y la presentación de los algoritmos más utilizados.

En la actualidad son muy pocos los sistemas que cuentan con autenticación facial, además se utilizan solamente en empresas donde se requiera un nivel de seguridad aceptable [Pentland, 2014], [Fuentes, 2011]. Con estas razones y con el objetivo de contar con más herramientas que permitan realizar autenticación facial de forma rápida y segura, se propone realizar una aplicación Android que permita realizar esta tarea y así pueda aplicarse como herramienta de autenticación móvil para validar transacciones, dar acceso a un lugar o ingresar a algún sistema.

El algoritmo PCA es muy parecido a LDA con la sutil diferencia que LDA hace una mejor reducción de la dimensionalidad de los datos y además una mejor separación de clases debido a que LDA coloca una etiqueta a los datos, lo que permite que se agrupen mucho mejor, además gracias a esta característica es considerado un algoritmo de aprendizaje supervisado, [Viola, 2001] adicionalmente explica la teoría matemática donde tomó como referencia una SVM (máquina de soporte vectorial) donde se reconoce que este algoritmo podría no resultar tan eficiente en comparación con la máquina de soporte vectorial, pero que en cuestión de costo computacional es mucho mejor [Pentland, 2014].

La primera implementación del algoritmo PCA (Análisis del componente principal) para autenticación facial fue propuesto por [Turk, 1991] donde se comprobó que este algoritmo es muy eficiente debido a que se reduce la dimensión de la imagen, una matriz formada por datos de varias imágenes a las que se les aplica una serie de operaciones matemáticas como el promedio, los valores propios, vectores propios, etc. Obteniendo así lo que se le denomina eigenfaces para clasificar las imágenes del rostro tal como se muestra en la figura 1, también se implementaron redes neuronales, al final se concluyó que gracias a la obtención de una cantidad menor de datos que representan en buena parte la matriz original, estos se pueden utilizar para autenticación, reduciendo así el costo computacional.

En [Delbracio, 2006] Se muestra una comparación entre LDA, PCA y ICA (análisis del componente independiente) se trabajó con una vasta colección de 1,176 imágenes, es decir, 49 personas con 24 imágenes cada una, se realizaron tres pruebas las cuales se basaron en seleccionar un conjunto reducido de imágenes que eran muy parecidas, también con un conjunto un poco más amplio y muy

parecidas y por ultimo un conjunto de imágenes pero con expresiones faciales distintas, este concluyo que el algoritmo LDA es mejor que PCA y ICA debido a que este algoritmo es el único que tiene un entrenamiento supervisado, también fue mejor a la hora de utilizar imágenes con distintas expresiones faciales.

Figura 1 Eigenfaces resultado del trabajo de [22].

La tesis presentada por [Mendoza, 2015] se enfoca únicamente en el algoritmo SURF modificado donde se destaca su implementación en dispositivos como teléfonos inteligentes y tabletas con sistema operativo Android y también se destaca la propuesta de una metodología que consta de seis etapas como se muestra en la figura 2 una de ellas es el procesamiento de las imágenes y su normalización, este algoritmo tiene la ventaja de usar un umbral de coincidencias, pero solo se comparan una a una cada par de imágenes lo que no resulta muy eficiente si la expresión facial de la persona va cambiando gradualmente, esto disminuye el número de coincidencias en la autenticación.

En el trabajo mostrado por [Hernández, 2010] se implementó un sistema de reconocimiento de rostros y se utilizó una metodología definida por seis etapas: Captura de la imagen, un pre procesamiento de imágenes, localización de la zona de interés, un escalamiento y ajuste, posteriormente la extracción de las

características faciales y por último la aplicación del algoritmo y la toma final de decisión.

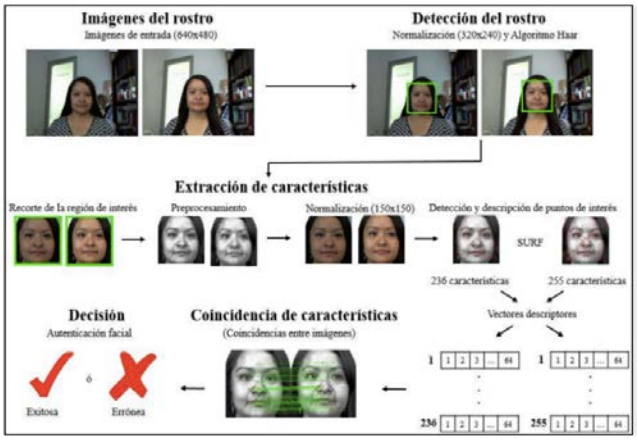

Figura 2 Metodología propuesta por [Mendoza, 2015].

2. Métodos

En este trabajo de investigación se implementaron los dos algoritmos principales LDA y PCA en la plataforma de Anaconda que permite ejecutar librerías open source y posteriormente la implementación de LDA en Android, donde la aplicación principal está escrita en java y manda llamar a las funciones escritas en C++ del algoritmo LDA el cual es un código nativo, la metodología para realizar el sistema de autenticación facial en Android es la que se muestra en la figura 3.

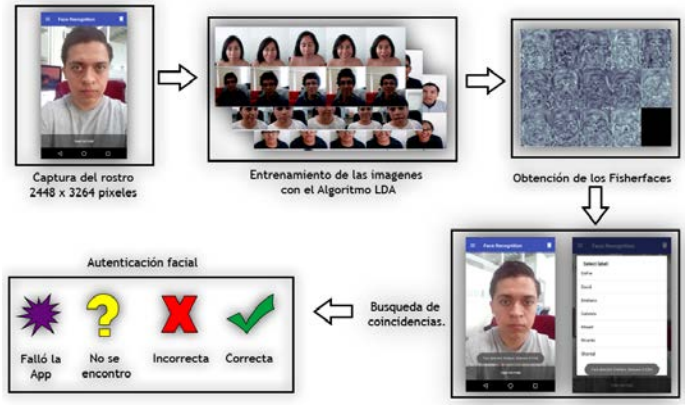

Figura 3 Metodología propuesta.

Donde se realizó la captura del rostro de todos los individuos, llevando a cabo el entrenamiento supervisado de LDA, después se obtienen las Fisherfaces y posteriormente se buscan las coincidencias con la matriz resultante dando así cuatro posibles resultados: Autenticación correcta, es decir, el usuario si se encontró; la autenticación incorrecta conocida también como falso positivo, es decir, se encontró coincidencia pero no con el usuario correcto; falso negativo donde no se encontró al usuario y si debió encontrarlo y por último donde se consideró la posible interrupción de la aplicación; el algoritmo LDA comprende los siguientes pasos:

- Construir la matriz X como en el método de eigenfaces, pero asignando a cada imagen una clase con la clase correspondiente a la matriz de clases c, ecuaciones 1 y 2.

$$X = \{X_1, X_2, \dots, X_c\} \tag{1}$$

$$X_i = \{x_1, x_2, \dots, x_n\} \tag{2}$$

- Proyectar la matriz X dentro de $(N - c)$-dimensinal sub-espacio a través de PCA con la matriz rotada. Donde N es el número de muestras en X y c es el número de clases o caras únicas, ecuación 3.

$$W_{PCA} = \arg max_W \, |W^T, S_T W| \tag{3}$$

- Calcular la matriz de dispersión entre clase y la matriz de dispersión dentro de la clase, ecuaciones 4 y 5.

$$S_B = \sum_{i=1}^{c} N_i \, (\mu_i - \mu)(\mu_i - \mu)^T \tag{4}$$

$$S_W = \sum_{i=1}^{c} \sum_{x \in X_i} (\mathrm{x}_k - \mu_i)(\mathrm{x}_k - \mu_i)^T \tag{5}$$

Donde μ es el promedio total de todas las clases y se calcula con la ecuación 6.

$$\mu = \frac{1}{N} \sum_{i=1}^{N} \mathrm{x}_i \tag{6}$$

Donde μ_i es el promedio de cada clase $i \in \{i, \dots, c\}$ y se calcula a partir de

la ecuación 7.

$$\mu_i = \frac{1}{|X_i|} \sum_{x \in X_i} x_j \tag{7}$$

- Aplicar el discriminante lineal de Fisher Y maximizar la relación entre el determinante de la matriz de dispersión entre clases y la matriz de dispersión dentro de la clase. La solución está dada por el conjunto de los vectores propios generalizados W_{FLD} de S_B y S_W, esto resulta $c-1$ valores propios diferentes a cero, ecuación 8.

$$W_{FLD} = \arg max_W \frac{|W^T W^T{}_{PCA}\, S_B W_{PCA} W|}{W^T W^T{}_{PCA}\, S_W W_{PCA} W} \tag{8}$$

- Obtener las Fisherfaces, ecuación 9.

$$W = W^T{}_{FLD}\, W^T{}_{PCA} \tag{9}$$

3. Resultados

El análisis de discriminación lineal es la técnica más común usada para reducir la dimensionalidad en el pre-procesamiento para la clasificación de patrones y aplicaciones como máquinas de aprendizaje, así como en la autenticación facial. El objetivo es proyectar un conjunto de datos dentro de un espacio de dimensión con una buena separabilidad de clases en orden de evitar el sobre ajuste y reducir el costo computacional [Raschka, 2014]. El enfoque general de LDA es muy parecido a PCA, pero LDA además de encontrar los ejes de los componentes que maximizan la varianza de los datos (lo que hace PCA), además se interesa en los ejes que maximizan la separación entre múltiples clases [Welling, 2005]. PCA maximiza la varianza entre componentes mientras que LDA además encuentra un sub-espacio de características que optimizan la separación de los datos a través de clases tal y como se muestra en la figura 4.

En este trabajo primero se realizaron pruebas de con una base de datos publica conocida como Yale Facedatabase A comúnmente conocida como Yalefaces, se decidió utilizar esta base de datos ya que se puede apreciar con más claridad la diferencia y eficiencia de los dos algoritmos principales de LDA y PCA. Estas

imágenes están en escala de grises, son 40 personas con 10 expresiones faciales distintas y sus dimensiones son de 92x112 pixeles como se muestra en la figura 5.

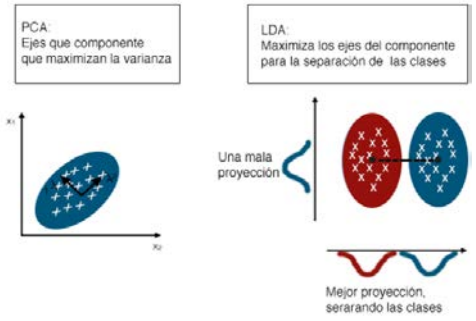

Figura 4 Comparación de PCA y LDA.

Figura 5 Rostros utilizados para las pruebas.

Para hacer las pruebas preliminares se utilizó el IDE Anaconda que es una plataforma de software libre que permite ejecutar librerías de data science en el lenguaje de programación Python con distintas librerías. En este caso se obtuvieron las 14 Eigenfaces que se muestran en la figura 6.

Debido a que el algoritmo PCA maximiza la variación entre las imágenes de entrenamiento con la imagen actual y a pesar de que las proyecciones de PCA a la hora de hacer una reconstrucción de los datos, este algoritmo no cumple con la discriminación de los datos, es decir, al tener un conjunto grande de imagen para comparar se empieza a perder información útil para poder hacer la discriminación necesaria y así lograr una autenticación más efectiva como se muestra en la figura 7, no alcanza a realizar todas las coincidencias de forma correcta. Como se explicó con anterioridad LDA maximiza la separación de los datos mediante

clases.

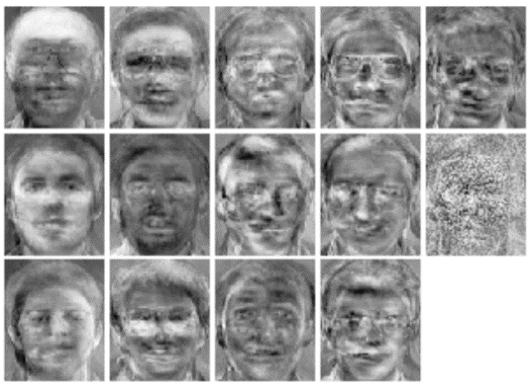

Figura 6 Eigenfaces obtenidas.

Figura 7 Coincidencias a partir del algoritmo PCA.

El método de Fisherfaces funciona a partir de una matriz de transformación específica de clase, por lo que no capturan la iluminación como el método de Eigenfaces. En cambio, el análisis discriminante lineal encuentra las características faciales para discriminar los datos entre las personas. El rendimiento de los Fisherfaces depende en gran medida de los datos de entrada, las Fisherfaces obtenidas se muestran en la figura 8.

Si los Fisherfaces se obtuvieron a partir de imágenes en condiciones bien iluminadas solamente y se intenta reconocer caras en escenas mal iluminadas, entonces el método encontrará componentes incorrectos, esto es algo lógico, ya que el método no tuvo oportunidad de aprender de la iluminación, esto no lo hace el método de Eigenfaces como ya se comprobó. De aquí la necesidad de utilizar este conjunto de imágenes de Yalefaces, ya que las 10 muestras de cada rostro tiene diferentes condiciones, es este caso la figura 9 muestra que efectivamente debido el método de Fisherfaces alcanza a encontrar todas las coincidencias dentro de la base de datos pública utilizada.

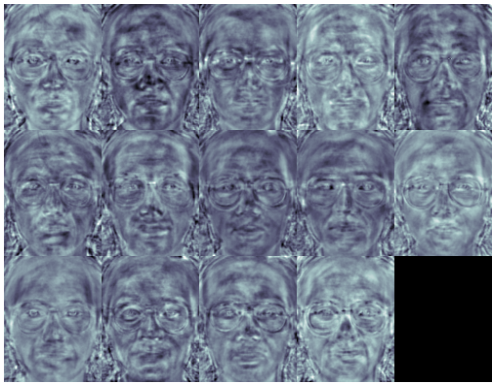

Figura 8 Fisherfaces obtenidas.

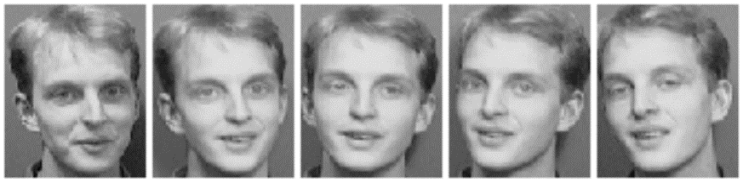

Figura 9 Coincidencias a partir del algoritmo LDA.

Como parte última de este trabajo de investigación se realizaron las pruebas de autenticación con una base de datos propia que se muestra en la tabla 1, en un teléfono inteligente con las características de harware y software que se muestran en la tabla 2.

Tabla 1 Descripción de la base de datos propia.

Descripción
10 personas, 5 imágenes distintas por persona, Imágenes en formato JPG, resolución de 4160 x 3120 pixeles, diversos fondos, iluminación, expresiones faciales, diferentes distancias entre el rostro y la cámara, distintos colores de piel y uso de lentes en algunas personas.

Tabla 2 Descripción del hardware y software del teléfono inteligente.

Descripción
Procesador ARM Cortex-A53, 1300 MHz con 4 núcleos. Procesador gráfico ARM Mali-T720 MP2, 600 MHz con 2 núcleos. Memoria RAM de 2 GB, 640 MHz. Tamaño de 12.7 cm, 720 x 1280 píxeles. Cámara de 4160 x 3120 pixeles, sensor de proximidad, acelerómetro, lector de huella dactilar.

El proceso de adquisición de las imágenes fue el siguiente: usuario se toma una selfie, es decir, un auto retrato, la primera vez que se hace esto se le asigna una

nueva etiqueta para dicho usuario (su nombre), y cada vez que el mismo usuario se tome la selfie debe seleccionar la etiqueta creada con anterioridad, las cinco expresiones faciales fueron serio, alegre, cerrando los ojos, sorprendido, y enojado, tal como se muestra en la figura 10.

Cada vez que se capturan usuarios y expresiones faciales, la aplicación procede a realizar el entrenamiento como se explicó con anterioridad. Al ser una imagen tomada desde el celular, el movimiento o la distancia no pueden ser acotados de forma estricta por lo que la aplicación muestra un mensaje de autenticación y la distancia del rostro con respecto a la cámara.

Figura 10 Base de datos propia.

Posteriormente de realizar el entrenamiento, se realizaron 10 pruebas a cada uno de los usuarios, es decir 100 pruebas en total, en este punto no se implementaron la extracción de las características faciales y el pre-procesamiento de las imágenes. En la tabla 3 se muestran los primeros resultados obtenidos, donde la columna prueba refleja las 10 pruebas de cada uno de los usuarios. La columna usuario muestra el nombre del usuario en cuestión, la tercera columna NP el número de pruebas de cada uno de ellos, la columna AC muestra el número de

autenticaciones correctas, la columna AI el número de autenticaciones incorrectas, es decir, que la aplicación no encontró al usuario, la columna FR muestra el número de falsos reconocimientos, es decir, el caso en el que al hacer el reconocimiento mostró el nombre de otro usuario y por último la columna EA que muestra el número de errores de la aplicación, es decir, donde la aplicación se cerró de forma abrupta sin poder realizar el reconocimiento.

Tabla 3 Resultados de las pruebas sin pre-procesamiento.

Prueba	Usuario	NP	AC	AI	FR	EA
1	Paulina	10	8	1	1	0
2	Dafne	10	8	2	0	0
3	Ricardo	10	9	0	1	0
4	Shantal	10	6	3	1	0
5	David	10	5	0	3	2
6	Campa	10	10	0	0	0
7	Gabriela	10	10	0	0	0
8	Misael	10	10	0	0	0
9	Emiliano	10	7	0	1	2
10	Rodrigo	10	7	0	2	1
Porcentajes de los resultados:			**80.00%**	**6.00%**	**9.00%**	**5.00%**

Hasta este momento los resultados mostrados en la tabla 4 los resultados son aproximados a otros resultados de otros trabajos de investigación del algoritmo LDA donde el porcentaje de autenticación va del 80.00% al 88.75%.

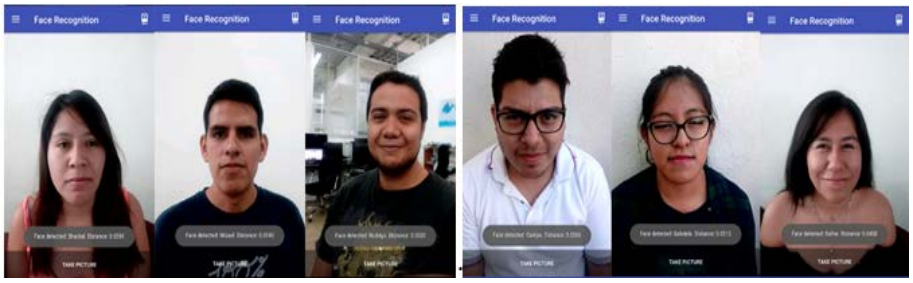

Figura 11 Resultados con la aplicación Android.

En base a los resultados obtenidos en las pruebas del algoritmo PCA y LDA que se mostraron anteriormente, se obtuvieron los porcentajes de autenticación, autenticación incorrecta, falso reconocimiento y error de aplicación.

Tabla 4 Resultados de los algoritmos PCA y LDA con 2 bases de datos distintas.

Algoritmo	DB	NP	Porcentajes			
			AC	AI	FR	EA
LDA Python	Yalefaces	14	87	6	7	0
PCA Python	Yalefaces	14	79	9	12	0
LDA Python	Propia	10	87	7	6	0
LDA Android	Propia	10	80	6	9	5

4. Discusión

El algoritmo de análisis de discriminación lineal mostro claramente ser superior al algoritmo de Análisis del componente principal, principalmente en condiciones diferentes de iluminación y distancia, aunque en la aplicación Android se mostraron algunas deficiencias de reconocimiento ya que al ser un dispositivo móvil al momento de tomar una foto, la misma puede estar distorsionada o bien en condiciones de iluminación que no son las ideales.

En trabajos anteriores de los que se habló, se mostraron buenos resultados con PCA debido a que se aplicaron acciones como procesamiento de imágenes y técnicas de visión por computadora como el face tracking, con estas acciones y a la propuesta de una metodología, es claro que es necesario contar con estos elementos para formar un buen sistema de autenticación y reconocimiento facial.

5. Conclusiones

En este trabajo de investigación se muestra la propuesta de autenticación facial utilizando el algoritmo LDA, la metodología que se siguió fue la de implementar el algoritmo de LDA y PCA en Python para poder tener una idea preliminar y contar con los elementos necesarios e implementar posteriormente el algoritmo más eficiente en una aplicación Android.

Se obtuvieron tiempos de respuesta aceptables para este algoritmo y por ser un algoritmo que funciona bien en condiciones de iluminación y distancia diferentes, a pesar de esto, se llegó a la conclusión de ser necesario una mejor metodología de sistema de reconocimiento facial.

Es posible mejorar el trabajo mostrado ya que en trabajos futuros se plantea

seguir una metodología de autenticación facial, y pre procesamiento de imágenes, este trabajo se enfocó principalmente en implementar el algoritmo en un sistema Android y realizar pruebas para probar únicamente su eficiencia.

6. Bibliografía y Referencias

[1] Aceves M. A. y J. M. Ramos, Fundamentos de Sistemas Embebidos (Ed.). Asociación Mexicana de Mecatrónica A.C. México, 2012.

[2] Anton, Howard. Introducción al algebra lineal/Por Anton, Howard. No. 512.897 A5.

[3] Brumnik, R., Podbregar, I. y Ivanuša, T., Reliability of Fingerprint Biometry (Weibull Approach). En Z. Riaz, Biometric Systems, Design and Applications, 2011.

[4] Chapra, Steven C. Canale, et al., Métodos numéricos para ingenieros. McGraw-Hill, 2007.

[5] Delbracio, M., & Mateu, M., Trabajo Final de Reconocimiento de Patrones: Identifiación utilizando PCA, ICA y LDA. Grupo de tratamiento de señales de la Universidad de de la Republica-Instituto de Ingenieria Electrica, Montevideo, Uruguay, 2016.

[6] Duró, V. E., Evaluación de sistemas de reconocimiento biométrico. Departamento de Electrónica y Automática. Escuela Universitaria Politécnica de Mataró, 2001.

[7] Duc, N. M., & Minh, B. Q., Your face is not your password face authentication bypassing lenovo–asus–toshiba. Black Hat Briefings, 2009.

[8] Embedinfo, Embedded System Development Specialist: http://www.embedinfo.com/en/ARM_Cortex-list.asp?id=15, 2014.

[9] Fuentes H. A., Recognition systems base on the facial image, Universidad Industrial de Santander, 2011.

[10] García, Gloria Bueno, et al., Learning Image Processing with OpenCV. Packt Publishing Ltd, 2015.

[11] Grossman, Stanley I., and Fernando Piña Soto, Álgebra lineal. No. 512.5 G7A4 1996 QA184. G37 1996. Grupo Editorial Iberoamericana, 1983.

[12] Hernández, R. G., Estudio de técnicas de reconocimiento facial., Departamento de Procesado de Señal y Comunicaciones. http://upcommons .upc.edu/pfc/bitstream/2099.1/9782/1/PFC_RogerGimeno.pdf, 2010.

[13] Howse, Joseph. Android Application Programming with OpenCV 3. Packt Publishing Ltd, 2015.

[14] Ifarraguerri A. y Chang C. I., Multispectral and hyperspectral image analysis with projection pursuit, IEEE Transactions on Geoscience and Remote Sensing. Approach). En Z. Riaz, Biometric Systems, Design and Applications, 2000.

[15] Introna, L., & Nissenbaum, H., Facial Recognition Technology A Survey of Policy and Implementation Issues, 2010.

[16] Martínez C., FACE recognition in the context of Smart Rooms, Department of Signal Theory and comunications UPC, 2005.

[17] Pentland A. y Adelson T., Face Recognition Demo Page, MIT Media Laboratory, 2014.

[18] Raschka, Sebastian, Implementing a principal component analysis (pca) in python step by step, 2014.

[19] Raschka, Sebastian, Linear Discriminant Analysis bit by bit, 2014.

[20] Raschka, Sebastian, Python Machine Learning. Packt Publishing Ltd, 2015.

[21] Sandoval, A. E. L., Mendoza, C., Martínez, L. Á. R. C., Rivas, E. A., Araiza, J. M. R. A., Carlos, J., & Ortega, P., Sistema de Autenticación Facial mediante la Implementación del algoritmo PCA modificado en Sistemas embebidos con arquitectura ARM. La Mecatrónica en México, 4, pp. 53-64, 2015.

[22] Serratosa, F., La biometría para la identificación de las personas. Universitat Oberta de Catalunya, pp. 8-20, 2008.

[23] Turk, M., & Pentland, A., Eigenfaces for recognition. Journal of cognitive neuroscience, 3(1), pp. 71-86, 1991.

[24] Villalba, A., Artacho, J. M., Sanchez, D., & Bernués, E., Autenticuz: Sistema de reconocimiento facial para control de acceso automático [DISK]. Zaragoza: Universidad de Zaragoza, 2004.

[25] Viola, P., & Jones, M., Rapid object detection using a boosted cascade of

simple features. In Computer Vision and Pattern Recognition, 2001, CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on IEEE, Vol. 1, pp. I-511, 2001.

[26] Vision and Modeling Group Vismod: http://vismod.media.mit.edu/vismod /demos/facerec/basic.html.

[27] Walpole, Ronald E., Raymond H. Myers, and Sharon L. Myers, Probabilidad y estadística para ingenieros. Pearson Educación, 1999.

[28] Welling, M., Fisher-lda. Technical report: http://www.ics.uci.edu/welling/ classnotes/papers_class/Fisher-LDA. Pdf, 2005.