UNIVERSIDAD
DE ANTIOQUIA
1 8 0 3

Verificación biométrica de identidad usando reconocimiento de rostros y patrones de tecleo.

Luis Felipe Gómez Gómez

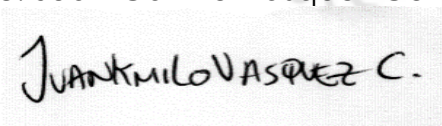
Universidad de Antioquia
Facultad de ingeniería
Departamento de Ingeniería Electrónica y Telecomunicaciones
Medellín, Colombia
2018

Verificación biométrica de identidad usando reconocimiento de rostros y patrones de tecleo.

Luis Felipe Gómez Gómez

Trabajo de grado presentado para optar por el título de:
Ingeniero de Telecomunicaciones.

Asesor:
MSc. Juan Camilo Vásquez Correa.



Línea de investigación:
Procesamiento Digital de Señales
Grupo de investigación:
Grupo de investigación en Telecomunicaciones Aplicadas GITA

Universidad de Antioquia
Facultad de ingeniería
Departamento de Ingeniería Electrónica y Telecomunicaciones
Medellín, Colombia
2018

Resumen

Los sistemas de verificación biométrica de identidad son un constante tema de estudio en los sistemas de aprendizaje automático, debido a su complejidad y sus múltiples variantes como el estudio de características fisiológicas o características conductuales. Los sistemas de verificación biométrica de identidad son usualmente usados en sistemas de control de acceso a zonas restringidas, en aplicaciones de detección de fraudes en sucursales bancarias y en sistemas de detección de fraudes exámenes en modelos de educación virtual. En este trabajo se propone un método de verificación biométrica de identidad a través de dos fuentes de información: (1) reconocimiento de rostros, y (2) análisis de patrones de tecleo. Para el reconocimiento de rostros se exploran y se implementan técnicas clásicas como aquellas basadas en el análisis de componentes principales (PCA) y máquinas de soporte vectorial (SVM) y técnicas del estado del arte como redes neuronales convolucionales (CNN). Para el análisis de patrones de tecleo se utilizan SVMs y redes neuronales (NN) totalmente conectadas entrenadas usando características temporales di-graficas como la latencia, duración en el proceso de tecleo, el tiempo entre soltar una tecla y presionar la siguiente, entre otras.

Los resultados indican que es posible verificar la identidad de una persona usando los rostros con aciertos de hasta el 80%, y del 76% usando patrón de tecleo. Este trabajo encontró las mejores métricas para obtener un buen desempeño del sistema de verificación, el cual se puede emplear en sistemas de seguridad o en sistemas de detección de fraudes en tiempo real.

Tabla de contenido

1. Introducción.

1.1. Contexto y motivación.

El presente proyecto tiene como objetivo el desarrollo de software que permite la verificación a través del reconocimiento de rostros y el reconocimiento de patrones de tecleo basado en reconocimiento de patrones, la cual es una de las técnicas de aprendizaje automático con mayor aplicabilidad en aspectos de seguridad y autenticación de usuarios.

La elección del tema se debe a la migración que se está presentando hacia la inteligencia artificial, principalmente para actividades de autenticación y verificación de la identidad de las personas que hacen uso de las diferentes plataformas; en las cuales estos son aspectos de vital importancia para su adecuado funcionamiento [1].

El uso de estos métodos presenta diferentes ventajas en comparación con las antiguas técnicas de identificación; entre ellas se encuentra eliminar el factor humano en el proceso de reconocimiento, el cual puede ser fácilmente sesgado y por tanto introducir errores en el proceso. Los métodos de verificación automática son fácilmente escalables, puesto que no requiere de elementos físicos especializados ni grandes capacidades de procesamiento; además que el mismo software puede ser empleado en diferentes aplicaciones. En todo desarrollo de tecnologías el primer paso en sus niveles de seguridad es prevenir accesos sin autorización, realizando sistemas de reconocimiento para prevenir ataques que afecten el sistema.

Los tipos de sistemas de verificación de identidad pueden ser basados en conocimientos, objetos, o marcadores biométricos. Los sistemas biométricos pueden dividirse en dos categorías: (1) sistemas basados en características fisiológicas de las personas como sus huellas digitales, el iris de sus ojos, entre otras, y (2) sistemas basados en características basadas en la conducta de la persona como la firma, la voz, o patrones de uso de dispositivos como teclados [2, 3, 4, 5, 6].

El reconocimiento de rostros es utilizado debido a la dificultad que presenta la falsificación de las características físicas del rostro humano, además de ser un método discreto que puede ser usado fácilmente en sistemas de alta vigilancia, en los cuales los usuarios no se percatan del uso de estos sistemas de reconocimiento. Por otro lado, el análisis de patrones de tecleo es utilizado debido a que una persona usa el teclado de la misma forma en todo posible escenario, así el usuario este usando el teclado para identificarse en una página web, escribiendo un correo, publicando en un foro, escribiendo un libro o escribiendo un mensaje corto para sus conocidos [3, 7, 8, 9].

1.2. Estado del arte.

El reconocimiento de rostros es un concepto relativamente nuevo. Desarrollado en los años 60, el primer sistema para reconocimiento facial era semiautomatizado y requería un administrador para localizar los rasgos en las fotografías para luego calcular distancias entre los puntos de referencia comunes. No fue hasta finales de los 80 que, con la aplicación de técnicas como el Análisis de componentes principales (PCA) [10], se crearon los primeros sistemas automatizados de reconocimiento de rostros en tiempo real. A partir de ese punto, surgieron dos enfoques para el reconocimiento: (1) el enfoque fotométrico basado en el análisis de imágenes planas [10, 11, 12]. (2) El geométrico que analiza características como los ojos, nariz, etc [13, 14, 15].

Con la aparición del Aprendizaje Profundo (Deep Learning), se han comenzado a utilizar redes neuronales convolucionales para la detección de rostros, con unos muy buenos resultados [16]. Actualmente existen muchas compañías compitiendo para obtener algoritmos con altos desempeños. Con el objetivo de obtener un nivel de precisión prácticamente igual al de los humanos. Dos grandes empresas tecnológicas, disponen de los algoritmos más avanzados de hoy en día. Ellas son Facebook y Google.

En 2014, Facebook anunció que había conseguido obtener un 97,3% de porcentaje de acierto con su algoritmo DeepFace [17] probado en la base de datos de imágenes LFW (Labeled Face in the Wild)[1]. Ese resultado mejoraba en un 27% el mejor resultado hasta el momento. Google, por su parte, con su algoritmo FaceNet [18] obtenía un porcentaje superior con un 99,6% de aciertos.

[1]Database: http://vis-www.cs.umass.edu/lfw/

El reconocimiento de rostros, entendido como un método de identificación de identidad de personas, evoluciona hacia nuevos niveles con el pasar de los días. No obstante, entre los objetivos de este trabajo no se encuentra la creación de nuevos algoritmos para el reconocimiento de rostros sino el estudio e implementación de técnicas ya existentes

Por otra parte, el análisis de patrones de tecleo tiene sus primeros pasos en la segunda guerra mundial, donde se construían perfiles para determinar que personas enviaban los mensajes en las líneas enemigas, analizando los tiempos entre pulsos de los mensajes en clave morse. Este basamento se extendió lográndose disímiles estudios que ayudaron a determinar que cada persona contiene un patrón único de tecleo basado en su habilidad, que puede ser utilizado para su identificación o verificación [19].

En la pasada década se han desarrollado modelos que responden a la obtención y comparación de las características de los usuarios, entre ellas aproximaciones estadísticas y de inteligencia artificial basadas en algoritmos genéticos y redes neuronales [20].

En 2014, Antal, Szabó y László probaron la posibilidad de utilizar el análisis de patrones de tecleo en plataformas móviles [21]. Ellos usaron la información de 42 usuarios. Usando diferentes metodologías de clasificación como K–vecinos más cercanos, SVM, redes neuronales artificiales, entre otros. Ellos obtienen una tasa de aciertos hasta del 88.33 %.

En este trabajo se diseñó e implemento un sistema biométrico usando dos diferentes fuentes de información: los rostros y el tecleo de forma independiente, basándose en el uso del reconocimiento de rostros y patrones de tecleo de personas. Además, se evaluará el efecto de uno de los problemas habituales en el reconocimiento de rostros como los cambios de iluminación en un ambiente determinado, y el cambio de ángulos de una persona frente a la cámara. Se utilizarán técnicas clásicas de reconocimiento de rostros como, el análisis de componentes principales (PCA), clasificadores como máquinas de soporte vectorial (SVM) [5, 8, 14, 25], redes neuronales artificiales (RNA) y redes neuronales convolucionales (CNN). Para el reconocimiento de patrones de tecleo, se abordarán diferentes

escenarios de captura de textos, simulando entornos de identificación de escritura, en el cual se usarán métodos de clasificación basados en redes neuronales artificiales [7, 8] y en SVM.

1.3. Objetivos.

- **Objetivo general:**

  Desarrollar un sistema de reconocimiento biométrico el cual identifique a las personas en dos etapas, a través de la identificación de su rostro y la comprobación de su patrón de tecleo.

- **Objetivos específicos:**
  - Recolectar una base de datos de rostros y patrones de tecleo de personas para la implementación del sistema biométrico.
  - Implementar algoritmos que permitan extraer la información más relevante de rostros de personas para su posterior reconocimiento.
  - Implementar algoritmos que permitan extraer medidas de tiempos entre diferentes teclas presionadas por el usuario para su posterior reconocimiento.
  - Identificar e implementar los métodos de clasificación más eficientes para el reconocimiento en cada modalidad.

2. Marco Teórico.

Los sistemas de reconocimiento biométrico actualmente en uso, pueden ser diseñados con diferentes tipos de reconocimiento, y pueden estar basados en biometría estática (huellas digitales, iris, retina, rostros, etc) o en biometría dinámica (voz, patrones de tecleo, conducta gestual, etc). En cualquier sistema de reconocimiento biométrico es necesario una base de datos robusta, para llevar a cabo el entrenamiento y la prueba de los métodos de clasificación usados en cada sistema [1, 9].

Se pueden utilizar diferentes métodos para el reconocimiento biométrico de personas, dependiendo la fuente de información. Por ejemplo, en el reconocimiento de rostros es común el uso de PCA, para la extracción de las características más importantes de los rostros de las personas. Las características extraídas luego son clasificadas usando SVM [11, 12, 13, 22]. En sistemas actuales se hace uso de las capacidades de las redes neuronales convolucionales (CNN), las cuales se han convertido en el estado del arte en problemas de clasificación de imágenes [13].

Otro ejemplo es el reconocimiento de patrones de tecleo, el cual busca autentificar personas basadas en la forma de su mecanografía, sin importar el tipo de texto que el usuario teclea. Para estos sistemas se extraen características básicas de tecleo, basadas en el tiempo entre teclas presionadas que habitualmente suelen ser tiempos bajos del orden de micro segundos [7]. Las características deben ser tomadas en segmentos pequeños de texto. Las medidas más comunes que pueden ser calculadas del tecleo de una persona incluyen características latencia de pulsaciones, tiempo de presión, tiempo de vuelo, tiempo entre tecla y tecla presionada, etc. [3, 7, 25]. El método de clasificación más usado para los sistemas de reconocimiento basados en patrones de tecleo son las redes neuronales artificiales [9, 10, 12, 24, 25].

2.1. Análisis de patrones de tecleo.

La dinámica de tecleo provee una detallada información de tiempos que describen de forma precisa los tiempos en los que se usa un teclado. Estos eventos generan 3 eventos comunes: (1) la presión de una tecla a usar, (2) la liberación de una tecla a usada y (3) el tiempo

de duración de una tecla usada, todos estos eventos suceden en el ingreso de cualquier palabra o texto en un computador y todos pueden ser monitoreados [8].

Este monitoreo es el proceso para encontrar medidas temporales referentes al estilo de teclear del usuario. Los cálculos de estas medidas son basadas en la combinación de los tiempos de presión y liberación de las teclas y son extraídos cuando el usuario en sesiones simples se le da la tarea de teclear textos o frases de diferentes longitudes. Como se observa en la Figura 1. Se pueden observar las diferentes medidas que pueden ser extraídas analizando la dinámica de tecleo de un usuario por medio de la agrupación di-grafica (di-graph), determinando a través de la serie de tiempo generada al teclear una palabra. A la serie de tiempo de presión y liberación se le pueden extraer las siguientes características: intervalo, tiempo de vuelo, latencia, tiempo de presión y final a final. Las características son nuevamente series de tiempo que se pueden modelar de forma estática calculando medidas estadísticas sobre la serie, o de forma dinámica analizando la serie de tiempo. El intervalo (superior izquierda) es la medida de tiempo determinada entre la liberación de una tecla y la presión de la tecla siguiente. El tiempo de presión (superior derecha) es la media de tiempo que nos da a conocer el tiempo total en el que estuvo presionada una sola tecla. La latencia, tiempo de vuelo y la medida final a final son medidas alternativas usadas en estudios previos [5, 15].

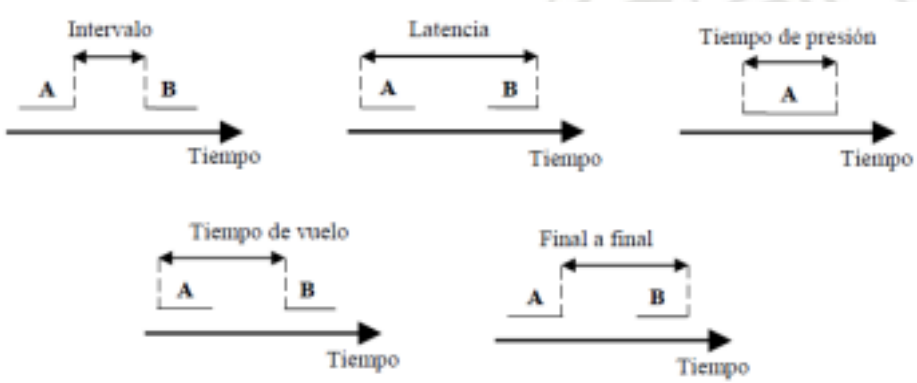


Figura 1. Medidas temporales. Intervalo, Tiempo de presión, Latencia, Tiempo de vuelo, y Final a Final.

En la Figura 2 se puede observar la serie de tiempo al escribir la palabra "GITA", a la cual se le puede determinar las medidas temporales anteriormente mencionadas.

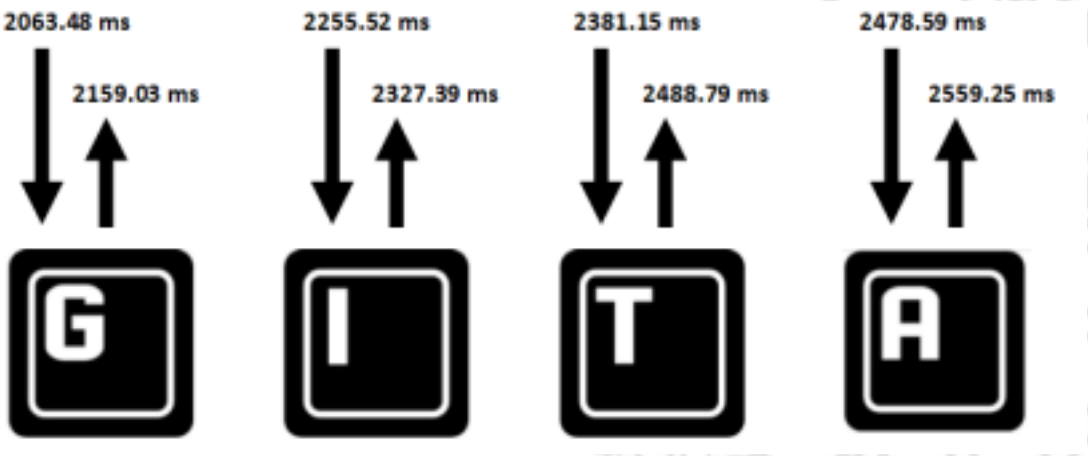


Figura 2. Tiempos de presión y liberación al escribir la palabra GITA.

2.2. Detección de rostros.

La alta variabilidad del rostro humano lo hace un objeto demasiado dinámico para la detección y uso en la visión artificial. En sus inicios la detección de rostros no fue un problema mayor para los desarrolladores ya que se partía de una imagen ya detectada. No fue hasta después de la década de los 90 que desarrollaron algoritmos de detección de rostros [27], proponiendo diversas técnicas como algoritmos de detección de bordes, como las cascadas Haar o algoritmos de detección de alto nivel que utilizan métodos avanzados de reconocimiento de patrones.

2.2.1. Cascadas Haar.

Los clasificadores Haar consisten en una serie de filtros ya diseñados con los cuales se busca encontrar un objeto en una imagen en concreto. En estos casos se busca encontrar un rostro en una imagen ya que estos pueden ser selectivos respecto a la orientación espacial y permiten ser modificados en escala y orientación. En la Figura 3 se pueden observar algunos de los filtros más comunes en la detección de rostros.

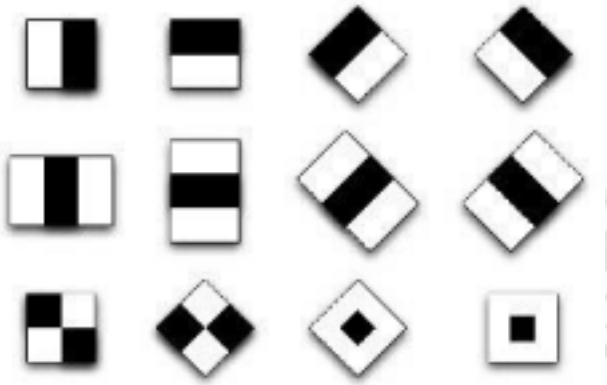


Figura 3. Filtros Haar rotados, trasladados y con cambios en su tamaño.

Los filtros de cascadas Haar, realizan codificaciones dado las diferencias de los valores de pixeles de la imagen, generando características de contornos, puntos y líneas, mediante la captura de contraste entre regiones en la imagen. En la Figura 4 se pueden apreciar los efectos de los filtros Haar a el rostro de una persona, mostrando los bordes y contornos de una parte de la imagen.

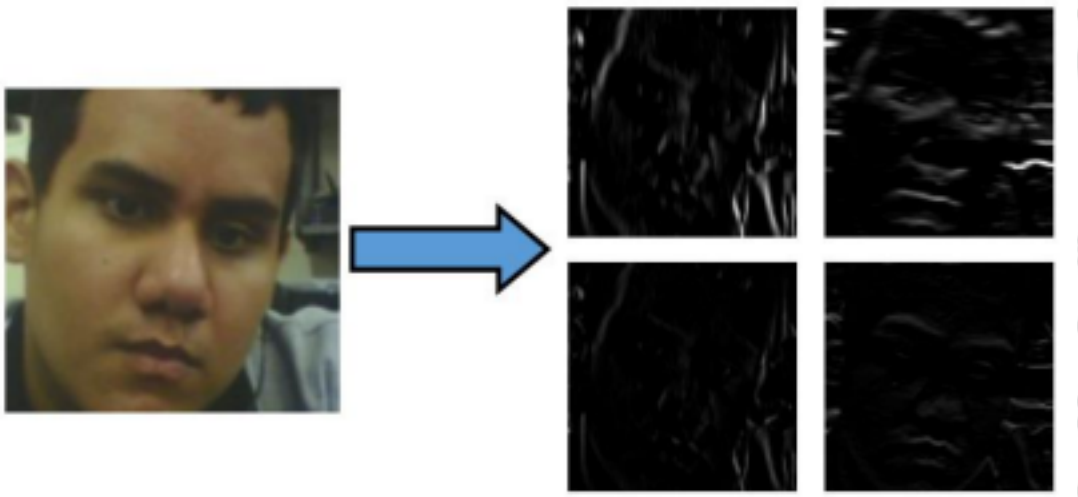


Figura 4. Efecto de los filtros Haar a la imagen de un rostro.

2.2.2. Análisis de componentes principales (PCA).

El análisis de componentes principales es un método de extracción de características el cual consiste en reducir la dimensión de un conjunto de n -características, a un nuevo conjunto de m -características con $m < n$. Este nuevo conjunto de características se

forma mediante una proyección de los datos originales en una base ortogonal, con la cual se pueda representar el conjunto de datos originales con cierta precisión [25]. Estos nuevos datos son obtenidos a través de los eigenvectores de la matriz de covarianza de los datos originales. Estas nuevas características son conocidas como eigenfaces o caras principales en tecnologías de reconocimiento de rostros. La dimensionalidad original viene dada por la cantidad de imágenes de rostros que existen en la base de datos. El número de componentes principales se determina en el porcentaje de información que se quiere conservar de las imágenes originales. Este método de extracción es usado en diferentes proyectos de reconocimiento biométrico de rostros [2, 10] debido que al reducir la dimensionalidad de una base de datos los costos computacionales usualmente disminuyen.

Este proceso de proyección lineal toma el espacio de imágenes original y lo transforma a un espacio vectorial de características de menor dimensionalidad, este cambio da un nuevo espacio vectorial que maximiza la dispersión de todas las imágenes proyectadas.

Inicialmente se considera un conjunto de N imágenes en el espacio de imágenes n-dimensionales, tal como se aprecia en la ecuación (1).

$$\{x_i\} \qquad i = 1, 2, 3, \ldots, N \qquad (1)$$

Asimismo, se considera una transformación lineal que lleva el espacio de imágenes original de n -dimensiones al nuevo espacio de características de m-dimensiones donde $m < n$. los nuevos vectores de características $y_k \in R^m$ son definidos por la siguiente transformación.

$$Y_k = W^T . x_k \qquad k = 1, 2, \ldots, N \qquad (2)$$

Donde $W \in R^{nxm}$ es una matriz con columnas ortonormales. Además, la matriz de distribución total S_T también es definida como:

$$S_T = \sum_{k=1}^{N} (x_k - \mu).(x_k - \mu)^T \qquad (3)$$

Donde μ es la media de todas las imágenes, definidas en la ecuación (1). Luego de aplicar la transformación lineal W^T, la distribución de los vectores de características $\{y_1, y_2, y_3, \ldots\ldots, y_N\}$ es $W^T S_T W$. Se toma aquella proyección W_{op} que maximiza el determinante de la distribución total de la matriz de las imágenes proyectadas, esto es.

$$W_{opt} = \arg\max_{W} [W^T S_T W] \qquad (4)$$

$$= [\omega_1, \omega_2, \omega_3, \ldots\ldots, \omega_m]$$

Donde el conjunto $\{w_i \mid i = 1,2,3, \ldots\ldots, m\}$ es el conjunto de vectores propios de n dimensiones. Estos vectores propios tienen las mismas dimensiones que una imagen del conjunto de datos original y se les denomina eigenfaces. En la Figura 5 se muestra las primeras eigenfaces obtenidas con un conjunto de imágenes de la base de datos de *Extend Yale Face Database B*[2] [8].

Figura 5. 20 caras principales de la base de datos *Extend Yale Face Database B.*

2.3. Aprendizaje automático.

2.3.1. Máquinas de soporte vectorial (SVM).

Las máquinas de soporte vectorial constituyen un método de clasificación lineal muy utilizado para reconocimiento biométrico, en el cual su objetivo principal es el de minimizar una función objetivo,

[2]Database: http://cvc.yale.edu/projects/yalefacesB/yalefacesB.html

encontrando los parámetros principales de un modelo de regresión lineal. Los parámetros que se minimizan en la función objetivo son la pendiente w y el termino independiente b. Este método al minimizar la función objetivo, se encarga de encontrar un hiper-plano óptimo de separación, ya que, en un conjunto de datos, pueden existir múltiples planos que separen el conjunto de datos, pero solo uno de estos planos es el que provee que el margen de separación sea máximo.

Para un conjunto de N datos de entrenamiento $X = \{x_1, x_2, x_3, \dots, x_N\}$ donde $x_i \in R^n$, cada conjunto de características tiene asociado una etiqueta y_i, que corresponderá a una de las clases que se pueden identificar. A el vector formado por $y_1, y_2, y_3, \dots, y_N$ le es conocido como vector de etiquetas o referencias.

En un problema linealmente separable se pueden determinar diferentes planos de separación, pero las SVM hallan el hiper-plano que maximiza la distancia entre el hiper-plano y el conjunto de datos. El hiper-plano que separa estos conjuntos esta dado de manera general por la ecuación (5).

$$(w.x) + b = 0 \quad donde \quad w, x \in R^n, \quad b \in R \qquad (5)$$

El proceso de entrenamiento de una SVM consiste en hallar el vector w de pesos que contiene la ponderación de las diferentes características, indicando qué tanto aportan en el proceso de clasificación, en tanto que b define el umbral de decisión. La función discriminante (distancia) d será la ecuación (6).

$$d(x, w, b) = \frac{|(w.x) + b|}{||w||} \quad con \quad ||w|| = \sqrt{w.w} \qquad (6)$$

donde $||w||$ es la norma asociada al producto escalar en R^n. Como se trata de patrones separables se puede reescalar w y b, de tal manera, como se muestra en la ecuación (7)

$$d(x, w, b) = \frac{1}{||w||} \Rightarrow |w \cdot xi + b| = 1 \qquad (7)$$

Así se obtiene el umbral optimo en el cual los patrones de entrenamiento más cercanos al plano tienen distancia normalizada

$d(x,w,b) = 1$, con $d(x,w,b) > 1$ para los demás patrones [28]. Dados los vectores de entrenamiento $x_i \in R^n$, $i = 1,2,\dots,N$, de dos clases, y un vector de etiquetas $y \in R^n$, de manera que $y_i \in \{-1,1\}$, suponemos que tenemos un hiper-plano que nos separa las muestras positivas y negativas. Los puntos de x que yacen en el plano serán los que satisfagan la ecuación, donde es ortonormal al hiper-plano, $\frac{b}{||w||}$ es la distancia perpendicular desde el hiper-plano al origen, y $||w||$ es la norma de w. Si consideramos $d_+ \, o \, d_-$ como la distancia más corta desde el hiper-plano a la muestra positiva (o negativa) más cercana y definimos el margen como $(d_+ + d_-)$, para el caso lineal separable, el algoritmo SVM buscará el hiper-plano que nos ofrezca un margen mayor como se aprecia en la Figura 6.

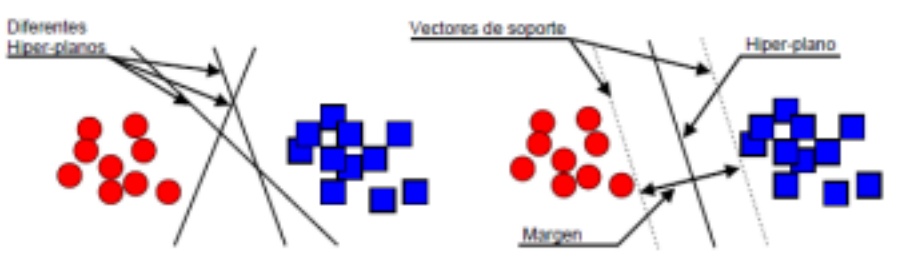


Figura 6. Diferentes planos que pueden separar el conjunto de datos (Izquierda), Plano óptimo de separación del conjunto de datos (Derecha) [6].

Para problemas donde no existe una separación lineal entre las clases, el conjunto de entrenamiento del sistema puede ser mapeado usando funciones kernel, las cuales crean un nuevo espacio de características.

En la definición del hiper-plano en SVM, los datos de entrenamiento aparecen en forma de producto escalar, $x_i \cdot x_j$, así que si transformamos los datos de entrada a un espacio Euclideo S de mayor dimensión usando la función $\phi(\cdot)$, tenemos que : $\phi : R^d \rightarrow S$. En consecuencia, a lo dicho anteriormente, tendremos que el problema del entrenamiento dependerá de $\phi^T(x_i) \cdot \phi(x_j)$ en el espacio S. Si encontramos una función Kernel que realice la operación $K(x_i, x_j) = \phi^T(x_i) \cdot \phi(x_j)$, podríamos usarla directamente sin necesidad de conocer ϕ a priori [29]. Así pues, una función kernel es una función que asigna a cada par de elementos del espacio de entrada, un valor real correspondiente al producto escalar de las

imágenes de dichos elementos en un nuevo espacio (espacio de características) [30]. Las funciones kernel comúnmente utilizadas son polinómicas, exponenciales y sigmoidales [8, 25]. En la Figura 7 se ilustra la idea del uso de un kernel, el cual, aplicado a un conjunto de datos linealmente no separables, al aplicar la función de kernel produce un conjunto de datos separables que se puede resolver como se ha mencionado en la sección anterior.

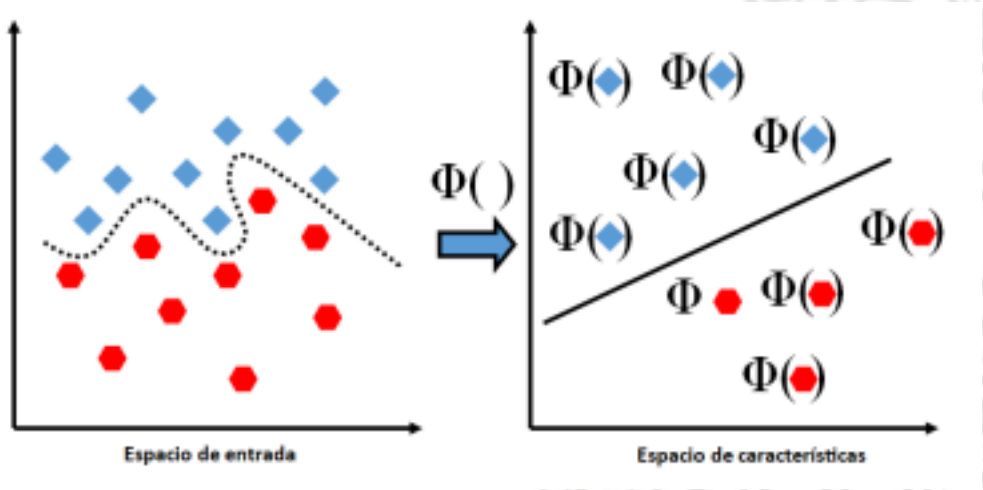


Figura 7. Uso de un kernel para la transformación de los datos

2.3.2. Redes neuronales artificiales (RNA).

Las redes neuronales artificiales son un conjunto de modelos matemáticos simples donde se pretende simular el comportamiento de una neurona biológica. Para entender el funcionamiento de una RNA es necesario conocer una simple neurona biológica, esta sin entrar en gran detalle se compone de 4 partes como se ve en la Figura 8.

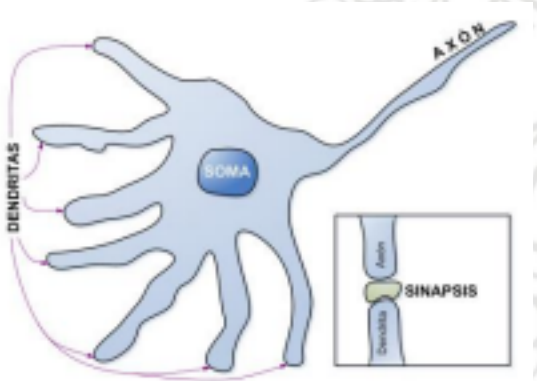


Figura 8. Composición de una neurona biológica

Dendritas.

Son los capilares que permiten el ingreso de impulsos eléctricos hacia el núcleo de la neurona. Es decir, las dendritas constituyen las "entradas" de una neurona biológica.

Sinapsis.

Es una membrana que se encuentra al extremo de cada dendrita. Esta sinapsis modifica el impulso eléctrico antes de la dendrita, atenuándolo o amplificándolo. Es conocido como el "peso sináptico" de una dendrita.

Soma.

Es el núcleo de la neurona, y es aquí donde se procesa todas las señales procedentes de las dendritas para generar una señal total. Este proceso consiste en dos etapas básicas. (1) La suma de las señales eléctricas provenientes de las dendritas. (2) Con la señal total obtenida de todas las dendritas, esta pasa por una función de transferencia propia de cada tipo de neurona y genera un resultado final.

Axón.

Es la salida de la neurona, es decir, el resultado de procesar todas las señales eléctricas provenientes de las dendritas, es transportada por el axón que puede extenderse no solo hasta otra neurona, sino que puede extenderse y dar su señal resultante hasta a 10000 neuronas [31], y así crear conexiones con otras neuronas y dar forma a lo que se conoce como una red neuronal.

Mediante los datos de entrada $X = x_1,\ x_2,\ x_3, \ldots\ldots, x_n.$, se modifican los pesos sinápticos $W = w_1,\ w_2,\ w_3\ , \ldots\ldots, w_n$ de una neurona, los cuales conforman una neurona junto con una función de activación $f(\ )$ que puede ser escalones, sigmoides, entre otras funciones. Estas ayudan a determinar una salida adecuada para el sistema como se ve en la Figura 9.

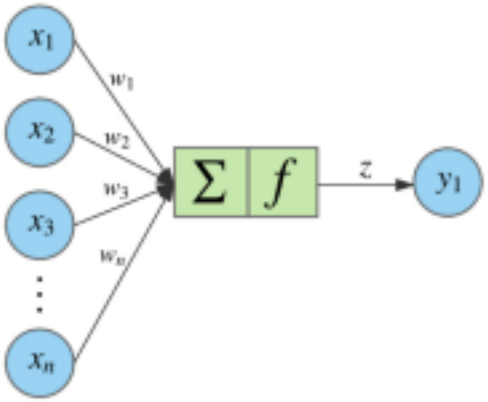


Figura 9. Composición de una neurona artificial [19].

La salida y de una sola neurona viene dada por la ecuación (8).

$$Z = f(x.w) = f\left(\sum_i xi.wi\right) \quad con \;\; x \in d_{1xn}, w \in d_{nx1}, z \in d_{1x1} \qquad (8)$$

Si la red neuronal contiene un bias la cual es una entrada adicional a la neurona, cuyo propósito es ayudar a la red si todos los valores de características en la entrada son 0. Determinando el siguiente cambio en la ecuación de la neurona como se expresa en la siguiente ecuación (9).

$$Z = f\left(b + \sum_i xi.wi\right) \quad donde \quad x \in d_{1xn}, w \in d_{nx1}, b \in d_{1x1}, z \in d_{1x1} \qquad (9)$$

El entrenamiento de una red neuronal se realiza principalmente usando el algoritmo de gradiente descendente, el cual busca determinar la variación de los pesos (ΔW) de cada neurona. El algoritmo busca minimizar una función de perdidas L, que se calcula entre la salida predicha Yo, y la salida deseada Yd. Existen varias funciones de perdidas, las más comunes incluyen el error cuadrático medio, o la entropía cruzada.

Luego de calcular la función de pérdidas, ésta se multiplica por la entrada, con en el fin de calcular la variación de los pesos, de acuerdo con la ecuación (10).

$$\Delta W = x_i * L \qquad (10)$$

Los pasos para el entrenamiento de una red neuronal son determinados en un algoritmo iterativo que emplea los siguientes pasos. Determinar la salida Y_o de la red neuronal ante una entrada X. Este proceso es llamado Propagación hacia adelante. Luego de estimar el valor predicho de la red se debe calcular el gradiente o la variación de los pesos semánticos determinado por la función de perdidas, con el fin de actualizar los pesos internos de la red neuronal. La actualización de los pesos semánticos de la red se determina con la ecuación (11), siendo μ la tasa de aprendizaje de la red. Repitiendo este proceso en diferentes iteraciones lleva a tener una red neuronal en la cual se espera encontrar el mínimo de la función de pérdidas.

$$W_{i+1} = W_i - \mu . \Delta W \qquad (11)$$

Las redes neuronales artificiales que se trabajan usualmente para problemas de reconocimiento de patrones, son redes de capas ocultas con múltiples neuronas, como se muestra en la Figura 10. Las cuales implementan un algoritmo de backpropagation para la reducción del error de la red, llevando el error a cada capa oculta de la red neuronal. [4]. Los parámetros de la red se modifican por medio de un algoritmo que permite minimizar una función de costo determinada a través de la salida de la red neuronal y la salida esperada del sistema.

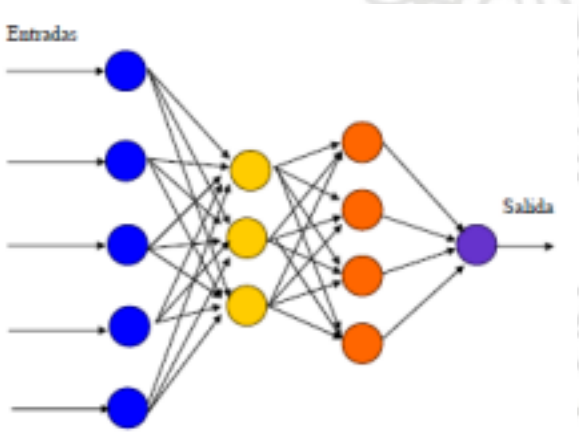


Figura 10. Red neuronal artificial multicapa.

2.3.3. Redes Neuronales Convolucionales.

Las redes neuronales convolucionales actualmente son el estado del arte en sistemas de clasificación de imágenes. En la mayoría de métodos aprendizajes se emplean arquitecturas de redes neuronales tradicionales, las cuales contienen comúnmente dos o tres capas ocultas, mientras que las redes neuronales profundas pueden tener hasta 150 capas. Uno de los tipos más comunes de redes neuronales profundas son las redes neuronales convolucionales. Una CNN convoluciona las características aprendidas con los datos de entradas y emplea capas convolucionales de 2D, la cual la hace una arquitectura más adecuada para el procesamiento de datos 2D, tales como imágenes.

Las CNN eliminan la necesidad de una extracción de características manual, por lo que no es necesario identificar las características utilizadas para la clasificar las imágenes. Las características relevantes se aprenden mientras la red se entrena con una colección de imágenes. Esta extracción de características automatizada hace que los modelos de aprendizaje profundo sean muy precisos para tareas de visión artificial.

Las CNN aprenden a diferenciar características de una imagen en cada una de sus capas ocultas. Cada capa oculta aumenta la complejidad de las características de las imágenes aprendidas. Por ejemplo, la primera capa oculta podría aprender como detectar bordes, mientras la segunda aprende a cómo detectar formas complejas propias del objeto que se intenta reconocer. Una CNN calcula la salida de la red por medio de un flujo de datos dividido en tres etapas:

(1) La imagen de entrada pasa por una serie de n filtros entrenados para extraer las características más importantes de la imagen para su reconocimiento, generando un nuevo conjunto de n imágenes filtradas. Un ejemplo sencillo es el que se ilustra en la Figura 11, el cual es el proceso de detección de bordes de una imagen cualquiera. Procesos similares son hecho en esta etapa, pero con diferentes tipos de filtros.

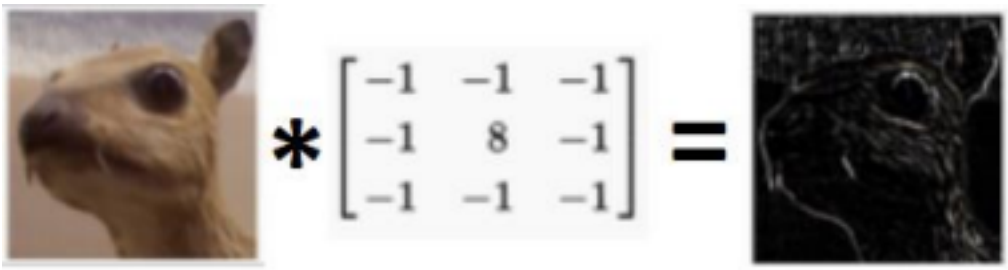


Figura 11. Proceso simple de filtrado para la detección de bordes.

(2) En este nuevo conjunto de n-imágenes se aplica una función de activación que comúnmente es una función lineal rectificada (ReLU) que se observa en la ecuación (12), aunque existen otras funciones de activación como la sigmoidal y la tanh, la ReLU se ha considerado que provee un mejor rendimiento en la mayoría de situaciones [4, 6, 34, 35].

$$\varphi(x) = \begin{cases} 0 & para\ x < 0 \\ x & para\ x \geq 0 \end{cases} \qquad (12)$$

(3) A el nuevo conjunto de n-imágenes encontradas se le somete a un proceso de sub-muestreo con el propósito de minimizar la alta dimensionalidad del conjunto de n -imágenes, este proceso de diezmado consiste en dejar la información más importante de la imagen filtrada. El diezmado de la imagen se puede utilizar diferentes tipos: Max, Promedio, Suma, etc. En la Figura 12 se puede observar el proceso de diezmado de un conjunto de pixeles de 4x4 a un nuevo conjunto de pixeles de 2x2 usando un diezmado Max.

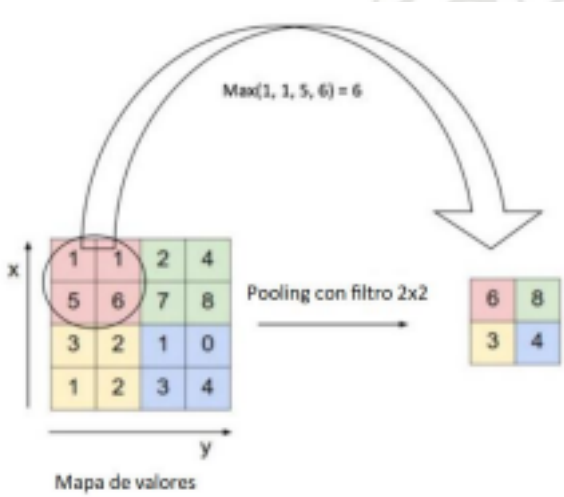


Figura 12. Proceso de diezmado utilizando un filtro de diezmado de 2x2 a un conjunto de pixeles.

En la Figura 13 se puede observar la arquitectura de filtrado y diezmado en una capa para imagen de un rostro común. El proceso de filtrado y diezmado de las imágenes en conjunto de entrenamiento se suele repetir varias veces con base en el número de capas de la red. (4) Finalmente, se adiciona una capa totalmente conectada y una capa de salida para tomar la decisión final de clasificación [14].

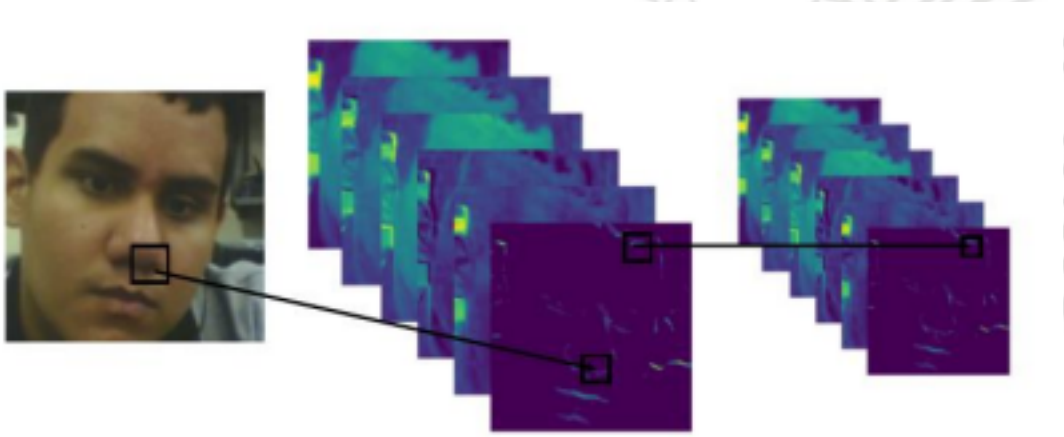


Figura 13. Procesos de filtrado y diezmado en las redes neuronales convolucionales.

3. Metodología

3.1. Base de datos.

Para el desarrollo del sistema de reconocimiento, se recolectó con la ayuda de diferentes estudiantes de la Universidad de Antioquia, una base de datos compuesta por las fotografías de los rostros de 52 diferentes estudiantes, 34 hombres y 18 mujeres entre los 19 y 26 años y además contiene diferentes archivos en los cuales se encuentra la información de la mecanografía de los estudiantes, el cual consta de diferentes tiempos de presión y liberación entre las teclas que se usaron en el sistema de reconocimiento. En esta base de datos participaron 52 estudiantes de la Universidad de Antioquia. El proceso de recolección se dio inicialmente con la recolección de las fotografías de los estudiantes en diferentes zonas del campus universitario, utilizando la cámara incorporada de un computador portátil Toshiba Satellite L845, aprovechando los diferentes sitios e iluminaciones que la universidad ofrece. En estos lugares se dieron 5 sesiones en las cuales se citaron a los estudiantes para la toma de 5 fotografías de sus rostros, en el cual se variaba la posición de los rostros de las personas frente a la cámara del computador portátil, tomando posiciones como totalmente de frente a la cámara, con el rostro inclinado hacia abajo y el rostro girado mirando hacia otro lugar, Los rostros de las imágenes fueron recortadas de forma automática usando la librería *opencv* de Python2 dando como resultado lo que se aprecia en la Figura 14.

Figura 14. Diferentes posiciones a los que fueron sometidos los estudiantes.

En la implementación de la captura de texto mecanográfico se utilizó un programa desarrollado en la plataforma Jupyter Notebook escrito en Python2, el cual consta 5 etapas de captura de datos, los cuales consistieron en el ingreso del nombre completo del estudiante, un documento de identidad y tres frases referentes a la universidad, estas frases fueron "universidad gita", "laboratorio led" e "investigacion udea" las cuales fueron escritas por los estudiantes que participaron 5 veces por sesión en un total de 4 sesiones, el programa desarrollado tomaba el tiempo en el cual se ejecutaba la presión de una tecla y el tiempo en el cual la tecla fuera liberada de la presión sometida, esta información se almacenaba en un archivo de datos .csv. En conjunto con la recolección de datos mecanográficos se realizó una toma extra de fotografías de los rostros de los participantes, cada 1,5 segundos mientras el sistema de captura de texto estaba activo el sistema obtenía la fotografía, esto se realizó para complementar la base de datos de rostros anteriormente descrita.

3.2. Sistema de verificación de rostros.

Con la base de datos recolectada se procedió a utilizar los diferentes métodos de extracción ya mencionados para el desarrollo del sistema de reconocimiento de rostros, primero en cada una de las fotos de la base de datos se les implemento los filtros de cascada Haar para la detección de los rostros en la imagen. Luego se utilizó el método de análisis de componentes principales (PCA) utilizando un desde un 70% hasta un 90% de la información de los rostros originales de la base de datos, con esto se obtuvieron diferentes caras principales o también conocidas como eigenfaces, como se muestra en la Figura 15.

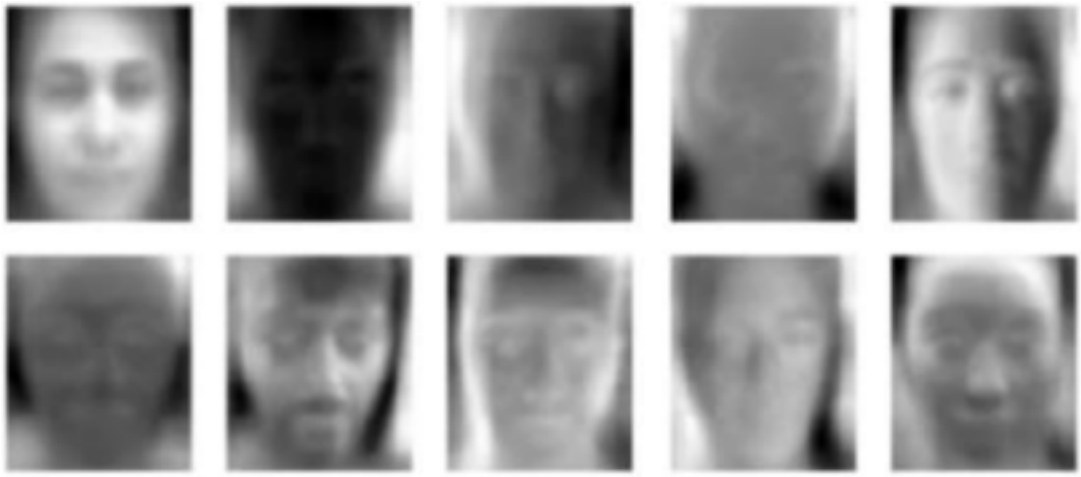


Figura 15. Primeras 10 cara principales (Eigenfaces) de la base de datos construida.

Con base en las eigenfaces encontradas se procedió a aplicar el método de reconstrucción de rostros partiendo de la nueva base creada con PCA, En la Figura 16 se puede apreciar el cambio de un conjunto imágenes de la base de datos cuando son reconstruidas con base en el 90% de la información original.

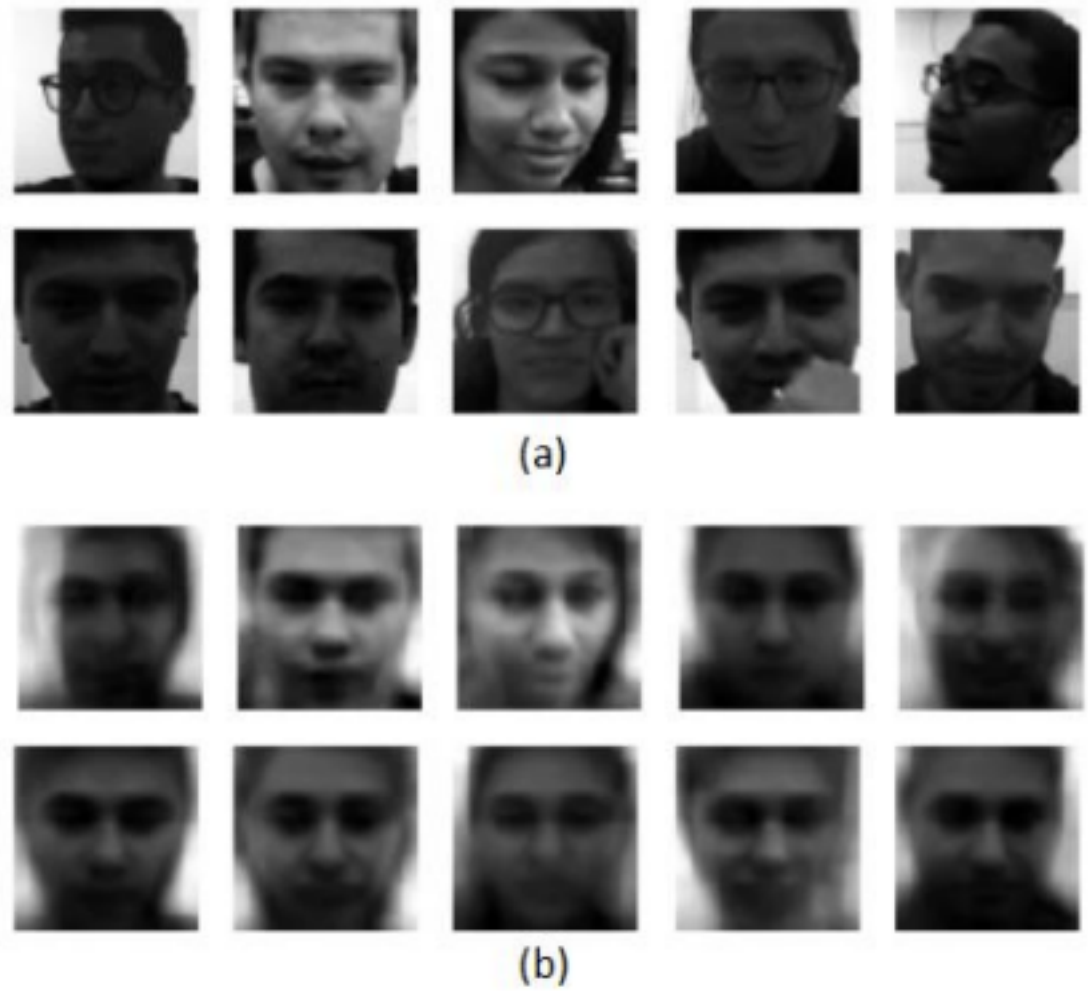


Figura 16. (a) Conjunto original de fotos. (b) Conjunto reconstruido con el 90% de la información original.

Luego estas características extraídas fueron clasificadas usando el método de máquinas de soporte vectorial (SVM) con una función lineal. Además de variar el parámetro C de entrenamiento de la SVM, con el fin de encontrar el valor de C que me entregue la SVM con menor error de validación, esto fue realizado entrenando y validando cada SVM con diferente parámetro C.

Los diferentes valores que tomará C para el entrenamiento de la SVM se muestran en la siguiente Tabla 1.

Tabla 1. Diferentes valores de C para encontrar el parámetro adecuado.

C	10^{-3}	10^{-2}	10^{-1}	10^{0}	10^{1}	10^{2}	10^{3}	10^{4}

Adicionalmente fue puesto a prueba un sistema que tenía la posibilidad de utilizar el de detección de rostros a través de redes neuronales convolucionales. Un sistema ya entrenado por miles de personas provenientes de la *Labeled Faces in the Wild Database* para la detección concreta de rostro de personas. Los rostros de la base de datos fueron normalizados respecto a una posición central como se muestra en la Figura 17.

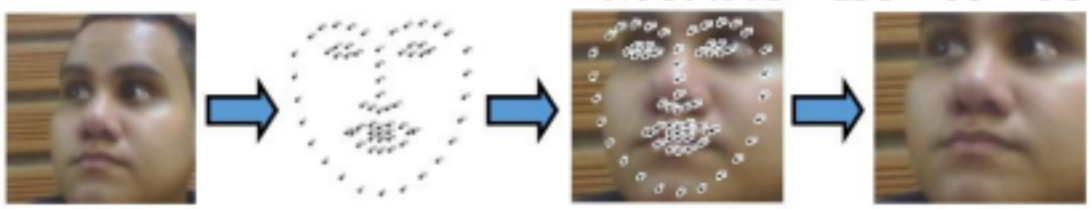


Figura 17. Normalización de los rostros para forma general de los rostros.

Posteriormente estos fueron procesados por una red neuronal convolucional de 128 salidas basada en la red neuronal convolucional Facenet, la cual contiene capas adicionales de módulos conocidos como Inception como la que se muestra en la Tabla 2 [32], estas salidas fueron usadas como características finales del rostro inicial. Luego de haber determinado las características de los rostros en la base de datos estos fueron guardados y debidamente etiquetados para su posterior uso en el sistema de verificación.

Tabla 2. Topología de la red neuronal convolucional de FaceNet.

Tipo de capa	Tamaño de salida
conv1 (7x7x3,2)	112x112x64
max pool + norm	56x56x64
inception (2)	56x56x192
norm + max pool	28x28x192
inception (3a)	28x28x256
inception (3b)	28x28x320
inception (3c)	14x14x640
inception (4a)	14x14x640
inception (4b)	14x14x640
inception (4c)	14x14x640
inception (4d)	14x14x640
inception (4e)	7x7x1024
inception (5a)	7x7x1024
inception (5b)	7x7x1024
avg pool	1x1x1024
fully conn	1x1x128
L2 normalization	1x1x128

El módulo inception es un módulo que es utilizado para desplegar múltiples filtros convolucionales y múltiples capas de diezmado en forma paralela dentro de una sola capa. En la Figura 18, se puede observar un diagrama del módulo inception usado en FaceNet, el cual utiliza filtros convolucionales 1x1, 3x3 y 5x5, además, utiliza filtros convolucionales de 1x1 para lograr la reducción de dimensionalidad de los datos.

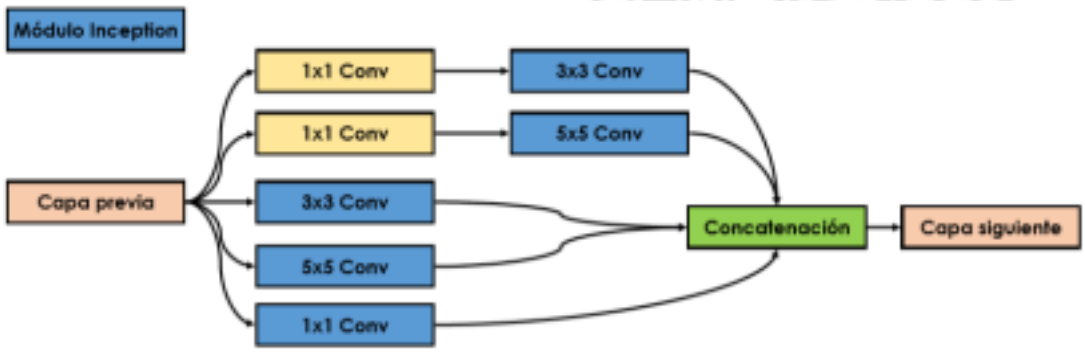


Figura 18. Módulo inception con reducción de dimensionalidad y múltiples filtros convolucionales.

El sistema de verificación final fue desarrollado como un sistema de tiempo real que utilizó la debida digitación de un número de identificación del usuario para determinar el conjunto de características el cual sería la base de verificación del usuario. Cuando un rostro sea identificado en pantalla por medio de las cascadas Haar o la red neuronal convolucional, a este se le extraerán las características pertinentes del método a trabajar. Luego, se comparan el rostro de entrada con el conjunto de datos seleccionado previamente, determinando la similitud que tiene el rostro por medio de la distancia coseno y una función de activación sigmoidal, esto permitiendo hallar el porcentaje de similitud entre el nuevo rostro y el rostro de la base de datos a comparar. La interfaz de usuario con la que se hicieron pruebas del sistema de tiempo real se muestra en la Figura 19, dicha interfaz se encarga de tomar un frame de video cada 0.1 segundos, al cual se le procede a detectar un rostro y si este es hallado el rostro es comparado con los guardados en la base de datos, determinando el valor promedio de similitud que tiene con el usuario registrado.

Figura 19. Interfaz de detección de rostros y verificación de usuario

3.3. Dinámica de tecleo.

En el sistema de identificación mecanográfica se procedió a la realización de un programa el cual tomará los archivos .csv y dividirá los datos de tiempos en el en dos columnas, la primera una columna con las teclas presionadas en la escritura de las frases y en la

segunda con los tiempos de liberación de las teclas usadas en la escritura de la frase. Luego estos vectores de tiempos fueron usados para encontrar las 5 diferentes métricas de medidas de tiempo que fueron descritas anteriormente, duración, latencia, intervalo, tiempo de vuelo y final a final. Las diferentes métricas fueron concatenadas en un vector de características de tiempos como se aprecia en la Figura 20.

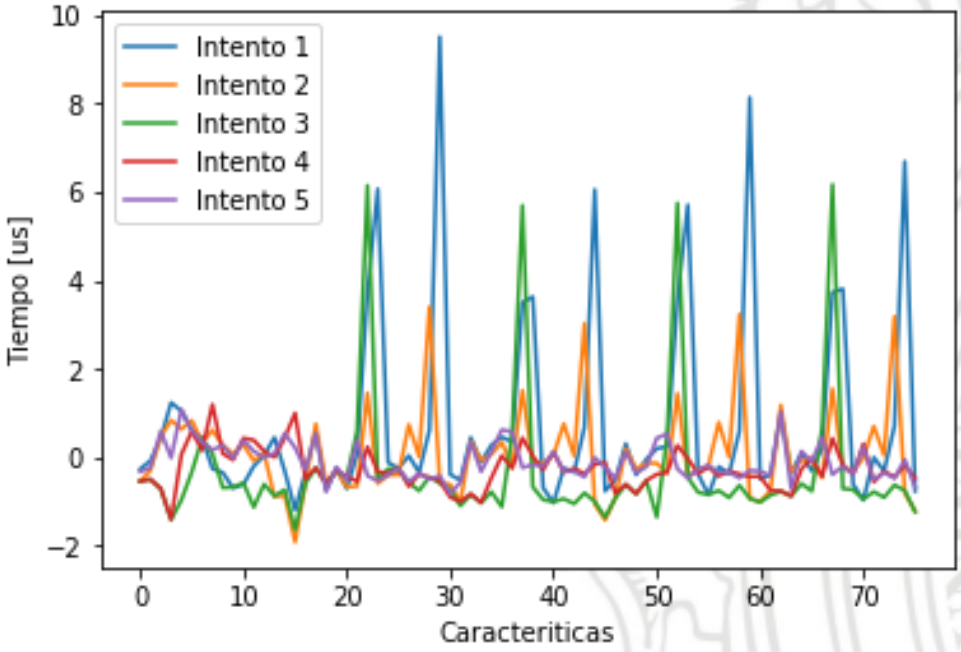


Figura 20. Múltiples vectores de características de un solo usuario para diferentes intentos de ingreso de la frase "universidad gita".

Además, en la Figura 21, se muestra la comparación entre los vectores de características de dos usuarios al teclear "universidad gita".

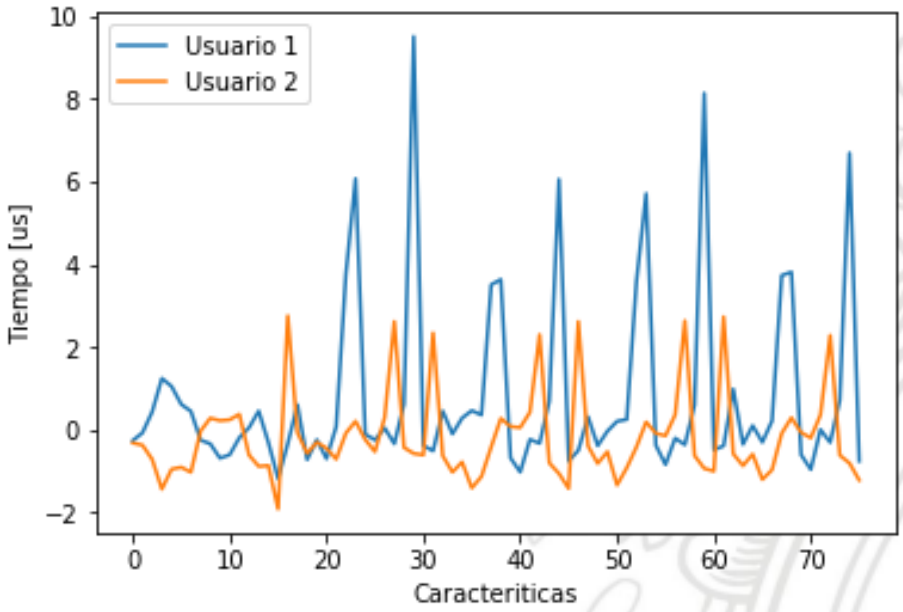


Figura 21. Comparación de los vectores de características de dos usuarios de la base de datos.

Con este vector de características determinado por la serie de tiempo de la base de datos, se entrenaron 5 sistemas de reconocimiento para usar:

El primero un sistema de reconocimiento basado en encontrar la mínima distancia euclidiana entre un nuevo conjunto de datos de entrada y los diferentes datos ya almacenados en la base de datos, encontrando el conjunto de datos más cercano y asociándolo a la persona con características más similares de tecleo.

El segundo un sistema de reconocimiento basado en encontrar el valor de correlación entre un nuevo conjunto de datos de entrada y los diferentes datos ya almacenados en la base de datos encontrando el conjunto de datos con un coeficiente de correlación más cercano a 1 y asociándolo a la persona con estas características de tecleo.

El tercero un sistema de reconocimiento basado en encontrar la distancia coseno entre un nuevo conjunto de datos de entrada y los diferentes datos ya almacenados en la base de datos encontrando

el conjunto de datos más cercano y asociándolo a la persona con características más similares de tecleo.

El cuarto un sistema basado en uso SVM con diferentes valores de C para la clasificación de los datos.

El quinto un sistema de reconocimiento con el uso de RNA, el cual se le varió su número de capas y su número de neuronas por capa.

4. Resultados y análisis.

4.1. Reconocimiento de rostros.

4.1.1. PCA-SVM

En la selección de mejores características usando PCA, se procedió a entrenar una máquina de soporte vectorial SVM dividendo los datos iniciales en un porcentaje de 70%-30% para entrenamiento y prueba, Esto se realizó para encontrar qué porcentaje de reconstrucción por medio de PCA nos brinda más aciertos en el proceso de entrenamiento y prueba de la SVM. En la Tabla 3 se encuentran los valores de porcentaje de reconstrucción del método PCA y el valor de C que brindo una mejor cantidad de aciertos en la prueba.

Tabla 3. Valores de reconstrucción de PCA, Valores de C en entrenamiento, Porcentaje de aciertos en entrenamiento y prueba.

PCA	Aciertos entrenamiento	Aciertos prueba	Valor C de la SVM
70 %	78 %	64 %	10
75 %	84 %	76 %	100
80 %	98 %	82 %	10
85 %	99 %	84 %	1000
90 %	99 %	87%	100

La implementación del primer sistema basado en PCA+SVM, brindo buenos resultados en su entrenamiento y prueba solo si se trabaja en un ambiente controlado como una sala de investigación. Cuando el sistema es entrenado en un ambiente controlado y puesto a prueba en un ambiente no controlado, este sistema se ve expuesto a cambios de iluminación y cambios de fondos en la imagen esto llevando a perdidas en la eficiencia del sistema, llevando al sistema a reducir bruscamente su porcentaje de aciertos como se puede observar en la Tabla 4.

Tabla 4. Porcentaje de acierto del sistema PCA entenado en condiciones controladas y puestos a prueba en dos diferentes escenarios.

Ambiente de prueba	Porcentaje de aciertos
Controlado	70 %
Reales	33 %
Cambios en la inclinación del rostro	46 %

Estos cambios en el porcentaje de aciertos es producto de la sensibilidad del método PCA frente los cambios de iluminación, y además de la posición de los rostros de las personas, por el cual, si una persona giraba su rostro el sistema volvería a fallar.

4.1.2. CNN

Para tener las mejores métricas la base de datos fue divida en un porcentaje 70%-30% para entrenamiento y prueba, respectivamente. Para cada conjunto de datos de prueba se le calculo su distancia coseno a cada conjunto de entrenamiento para así encontrar los valores de correcta predicción o de incorrecta predicción y con esto determinar un promedio de tasa de falsa aceptación o false true rate (FTR) y un promedio de falso rechazo o false negative rate (FNR). Los valores promedios se exponen en la Tabla 5.

Tabla 5. Valores promedios de tasa de falsa aceptación y de tasa de falso rechazo.

Métricas	Porcentaje
FPR	7.95 % ± 5.09 %
FNR	8.41 % ± 4.31%

Los resultados de entrenamiento y prueba del sistema son mostrados a través de la Figura 22, la cual suministra la información de la ROC promedio de entrenamiento, la cual se construye con base a la razón de verdaderos positivos y la razón de falsos positivos para diferentes valores de umbral para la distancia coseno en conjunto con la función activación sigmoidal.

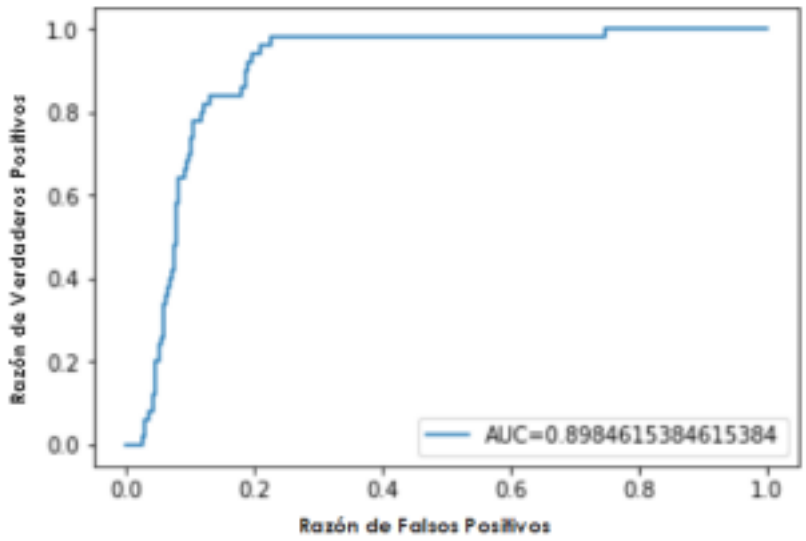


Figura 22. Razón de Verdaderos Positivos V.S. Razón de Falsos Positivos

Para el sistema de verificación actual, el área de la ROC fue de 0.89 aproximadamente, en los sistemas de verificación se busca que el área de la ROC sea mayor al 0.5 y lo más aproximado a 1.0. Esto ayuda a determinar qué tan eficiente es el sistema a la hora de verificar la identidad de un usuario que se encuentre en frente a la cámara.

Para este sistema, la CNN se comporta de forma muy estable en los cambios de ambiente en el proceso de prueba, en el laboratorio y por fuera de este el porcentaje de aciertos estuvo encima del 80%. Los diferentes cambios de iluminación y los cambios de posición del rostro disminuyeron el porcentaje de aciertos del sistema de una forma mínima, en la Tabla 6 se muestran los diferentes porcentajes de aciertos en las diferentes pruebas.

Tabla 6. Porcentaje de acierto del sistema CNN entenado en condiciones controladas y puestos a prueba en dos diferentes escenarios.

Ambiente de prueba	Porcentaje de aciertos
Controlado	92.4 %
Reales	85.3 %
Cambios en la inclinación del rostro	88.7 %

4.2. Dinámica de tecleo.

El sistema de dinámica de tecleo se procedió a entrenar 5 diferentes sistemas, todos utilizando las 3 frases con la que se creó la base de datos, cada sistema con un método diferente a utilizar para encontrar el que mejor se adapte a este problema. Para los sistemas de SVM y RNA se procedió a utilizar una partición de 70%-30% para entrenamiento y pruebas, además de utilizar sesiones nuevas con los estudiantes para la realización de pruebas adicionales, que arrojaron los resultados que se aprecian en la Tabla 7.

Tabla 7. Aciertos en las pruebas extra que se trabajaron para cada sistema.

Método	**Aciertos en pruebas**
Distancia Euclidiana	23 %
Coeficiente de correlación	55 %
Distancia coseno	53 %
SVM	62 %
RNA	71 %

Los resultados muestran una muy baja probabilidad de aciertos, debido a que la base de datos de información era demasiado pequeña de unos 60 archivos de datos de tecleo por persona, y en la literatura se encuentran desde 2000 archivos de datos de escritura por persona [9, 26, 33]. Dado esta información el sistema no es capaz de parametrizar los datos para encontrar el modelo que se adapte a todos los casos en los que una persona puede teclear, ya sea por su costumbre al teclado o por su estado de ánimo al escribir.

5. Conclusiones

Al poner a prueba ambos sistemas de PCA-SVM se ven muchas fallas en este método, ya que para poder tener una medida convincente para el trabajo se tuvo que realizar muchas pruebas para el sistema en un ambiente muy controlado, ya que cambios en la iluminación, en el fondo e incluso cambios en el rostro como su posición o si la persona no usaba lentes o se afeitaba la tasa de aciertos disminuía de una manera muy crítica, pasando de un 80% de aciertos a un 35% de aciertos.

Al contrario de los sistemas de aprendizaje profundo como la CNN trabajada, cuyo diseño se basa en la red FaceNet [18], sistema el cual permite parametrizar con una manera muy distinguible a los usuarios totalmente diferentes, y de una forma similar a las diferentes imágenes de un mismo usuario, esto permitiendo que una persona así su rostro estuviera girado, con gafas o si su ambiente fuese afectado por opciones como iluminación y fondos, no se viera afectado en el ingreso al sistema, tal cual como se aprecia en los resultados brindando hasta una tasa de falso rechazo 15 %.

En el sistema de tecleo partió de 5 medidas básicas basadas en el análisis di-graph, estas características fueron mezcladas y probadas para hallar la mejor combinación características, en las pruebas hechas se vio que utilizar las 5 medidas de tiempo, brindaba el desempeño más alto en el sistema, de un 76 % en pruebas. Aun así, utilizando ese número de características el sistema tuvo un desempeño bajo en las pruebas extras, debido a que la base de datos no era lo suficientemente extensa para encontrar un modelo adecuado para la verificación.

En futuros trabajos, para mejorar el desempeño del sistema de verificación de identidad, se buscará implementar sistemas multimodales, los cuales unirán los datos determinados por los rostros de los usuarios y los datos de patrones de tecleo, buscando mejorar los resultados vistos. Adicionalmente, se plantea la posibilidad de implementar métodos como las redes neuronales recurrentes (RNN) con unidades de larga memoria de corto plazo (LSTM) [36].

Referencias Bibliográficas

[1] GÁMEZ, F. D. G. (2016). *Una técnica para mejorar la implantación de la autenticación facial en entornos virtuales de aprendizaje en la educación superior* (Doctoral dissertation, Universidad a Distancia de Madrid).

[2] BELHUMER, P.N., HESPANHA, J.P, KRIEGMAN, D.J. (1997). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. En: IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7). p- 711-720.

[3] MONROSE y A. D. RUBIN. Keystroke dynamics as a biometric for authentication. En: Future Generation Computer Systems., Vol. 16, 2000, No. 4, p. 351–359.

[4] H. MACILRAITH, A.; CARD. Birdsong recognition using backpropagation and multivariate statistics. *IEEE Transactions on Signal Processing*, *45*(11), 2740-2748.

[5] NISENSON, M., et al. Towards behaviometric security systems: Learning to identify a typist. En *European Conference on Principles of Data Mining and Knowledge Discovery*. Springer, Berlin, Heidelberg, 2003. p. 363-374.

[6] HUANG, G. B., RAMESH, M., BERG, T., y LEARNED-MILLER, E. (2007). *Labeled faces in the wild: A database for studying face recognition in unconstrained environments* (Vol. 1, No. 2, p. 3). Technical Report 07-49, University of Massachusetts, Amherst.

[7] DHOLI, P.R. CHAUDHARI, K.P. Typing Pattern Recognition Using Keystroke Dynamics. En: Communications in Computer and Information Science, 296 (2013). p. 275-280.

[8] SHIMSHON, T. MOSKOVITCH, R., ROKACH, L., & ELOVICI, Y. Clustering di-graphs for continuously verifying users according to their typing patterns. En: Proceedings of the IEEE 26th convention of electrical and electronics engineers in Israel (2010). p. 445–449.

[9] ALI, M. L., MONACO, J. V., TAPPERT, C. C., and Qiu, M. (2017). Keystroke biometric systems for user authentication. *Journal of Signal Processing Systems*, *86*(2-3), 175-190.

[10] TURK M. and PENTLAND A.. Eigenfaces for Recognition. En: Journal of Cognitive Neuroscience, vol. 3, no. 1, 1991.

[11] GUMUS, E. KILIC, N. SERTBAS, Ahmet. UCAN, Osman. Evaluation of face recognition techniques using PCA, wavelets and SVM. En: Expert Systems with Applications. 37 (2010). p. 6404-6408.

[12] DÉNIZ, O. CARTRILLÓN, M. HERNÁNDEZ, M. Face recognition using independent component analysis and support vector machines. En: Pattern Recognition Letter. 24 (2003). p. 2153-2157.

[13] WANG, C., LAN, L., ZHANG, Y., y GU, M. (2011, May). Face recognition based on principle component analysis and support vector machine. In *Intelligent Systems and Applications (ISA), 2011 3rd International Workshop on* (pp. 1-4). IEEE.

[14] LECUN, Y., BENGIO, Y. & HINTON, G. Deep learning. Nature 521, 2015. p. 436–444.

[15] GUNETTI, D.; PICARDI, C. Keystroke analysis of free text. *ACM Transactions on Information and System Security (TISSEC)*, 2005, vol. 8, no 3, p. 312-347.

[16] QIN-QIN T, SHU Z, XIAO-Hong i, and TORU K. Robust face detection using local cnn and svm based on kernel combination. Neurocomputing , 211:98 – 105, 2016. ISSN 0925-2312. SI: Recent Advances in SVM.

[17] TAIGMAN, Y., YANG, M. y RANZATO, M. (2014) DeepFace: Closing the Gap to Human-Level Performance in Face Verification. California: Facebook AI Research.

[18] Applied Deep Learning. Web. En: https://towardsdatascience.com/applied-deep-learning-part-1-artificial-neural-networks-d7834f67a4f6. Arden Dertat, 8 Agosto, 2017. 2 Mayo, 2018.

[19] DVORAK, A., MERRICK, N. L., DEALEY, W. L., & Ford, G. C. (1936). Typewriting behavior. *New York: American Book Company*.

[20] CHO, S., HAN, C., HAN, D. H., & KIM, H. I. (2000). Web-based keystroke dynamics identity verification using neural network. *Journal of organizational computing and electronic commerce*, *10*(4), 295-307.

[21] ANTAL, M., SZABÓ, L. Z., & LÁSZLÓ, I. (2015). Keystroke dynamics on android platform. *Procedia Technology*, *19*, 820-826.

[22] LUO, Yuan. WU, Cai-ming. ZHANG, Yi. Facial expression recognition based on fusion feature of PCA and LBP with SVM. En: Optik. 124 (2013). p. 2767-2770.

[23] STAN, Z. Li. ANIL, K. Jain. Handbook of Face Recognition. Segunda Edición. London. Springer, 2011.

[24] ANIL, K. Jain. PATRICK, Flynn. ARUN, A. Ross. Handbook of biometrics. London. Springer, 2008.

[25] HEMPSTALK, K. Continuous typist verification using machine learning. Hamilton, 2009, 178. Doctor of Philosophy. The University of Waikato. Department of Computer Science.

[26] COAKLEY, M. J., MONACO, J. V., and TAPPERT, C. C. (2016, September). Keystroke biometric studies with short numeric input on smartphones. In *Biometrics Theory, Applications and Systems (BTAS), 2016 IEEE 8th International Conference on* (pp. 1-6). IEEE.

[27] GUEVARA, M. L., ECHEVERRY, J. D., & URUEÑA, W. A. (2008). Detección de rostros en imágenes digitales usando clasificadores en cascada. *Scientia et technica*, *1*(38).

[28] MOSCOVITZ, L. J., and RENGIFO, P. R. (2010). Al interior de una máquina de soporte vectorial. *Revista de Ciencias*, *14*, 73-85.

[29] AGUILAR GONZÁLEZ S. R. Igualación de canal no lineal mediante algoritmos kernelizados, septiembre 2009. URL http://hdl.handle.net/10016/5943.

[30] CARMONA SUAREZ E. J., Tutorial sobre máquinas de vectores soporte (SVM). Dpto. de Inteligencia Artificial, ETS de Ingeniería Informática, Universidad Nacional de Educación Distancia (UNED), 1(1):1–25, 2014.

[31] LOPEZ LOAIZA D. Diseño y construcción de una red neuronal artificial general. Universidad Politécnica Salesiana. Octubre, 2007.

[32] SCHROFF, F., KALENICHENKO, D., & PHILBIN, J. (2015). Facenet: A unified embedding for face recognition and clustering. En: *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 815-823).

[33] MONACO, J. V., BAKELMAN, N., CHA, S. H., and TAPPERT, C. C. (2012, August). Developing a keystroke biometric system for continual authentication of computer users. En: *Intelligence and Security Informatics Conference (EISIC), 2012 European* (pp. 210-216). IEEE.

[34] GEORGHIADES, A.S. and BELHUMEUR, P.N. and KRIEGMAN, D.J. From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose. IEEE Trans. Pattern Anal. Mach. Intelligence 23(6):643-660 (2001).

[35] HUANG, G., MATTAR, M., LEE, H., and Learned-Miller, E. G. (2012). Learning to align from scratch. In *Advances in neural information processing systems* (pp. 764-772).

[36] KOBOJEK, P., y SAEED, K. (2016). Application of recurrent neural networks for user verification based on keystroke dynamics. En: *Journal of Telecommunications and Information Technology*, (3), 80.