SISTEMA DE CONTROL DE ACCESO USANDO RECONOCIMIENTO FACIAL CON UNA RASPBERRY PI 4 Y OPENCV

ACCESS CONTROL SYSTEM USING FACIAL RECOGNITION WITH A RASPBERRY PI 4 AND OPENCV

José Ignacio Vega Luna
Universidad Autónoma Metropolitana, Unidad Azcapotzalco, México
vlji@azc.uam.mx

Gerardo Salgado Guzmán
Universidad Autónoma Metropolitana, Unidad Azcapotzalco, México
vlji@azc.uam.mx

Francisco Javier Sánchez Rangel
Universidad Autónoma Metropolitana, Unidad Azcapotzalco, México
vlji@azc.uam.mx

José Francisco Cosme Aceves
Universidad Autónoma Metropolitana, Unidad Azcapotzalco, México
vlji@azc.uam.mx

Víctor Noé Tapia Vargas
Universidad Autónoma Metropolitana, Unidad Azcapotzalco, México
vlji@azc.uam.mx

Mario Alberto Lagos Acosta
Universidad Autónoma Metropolitana, Unidad Azcapotzalco, México
vlji@azc.uam.mx

Recepción: 30/octubre/2020 **Aceptación:** 4/diciembre/2020

Resumen

El objetivo de este trabajo fue realizar un sistema de control de acceso usando reconocimiento facial para acceso a un centro de datos. Se desarrolló usando una tarjeta Raspberry Pi 4, una cámara de video y una pantalla táctil. La programación del sistema implanta el algoritmo de Viola-Jones para la detección del rostro y el reconocimiento del mismo usando funciones de OpenCV. La interfaz de usuario se muestra en la pantalla táctil. Cuando un usuario no autorizado intenta acceder al centro de datos, se transmite un mensaje de alerta de WhatsApp a un teléfono móvil.

Las pruebas realizadas mostraron que la exactitud del sistema es 99.6 % y el tiempo de respuesta 400 ns. A partir de los resultados logrados el sistema puede usarse en otro tipo de instalaciones o aplicaciones de tiempo real.

Palabras Clave: OpenCV, Raspberry Pi 4, reconocimiento facial, Viola-Jones, WhatsApp.

Abstract

The objective of this work was to make an access control system using facial recognition for access to a data center. It was developed using a Raspberry Pi 4 card, a video camera and a touch screen. System programming implements the Viola-Jones algorithm for face detection and face recognition using OpenCV functions. The user interface is displayed on the touch screen. When an unauthorized user tries to access the data center, a WhatsApp alert message is transmitted to a mobile phone. The tests carried out showed that the accuracy of the system is 99.6 % and the response time 400 ns. Based on the results achieved, the system can be used in other types of installations or real-time applications.

Keywords**:** *Face recognition, OpenCV, Raspberry Pi 4, Viola-Jones, WhatsApp.*

1. Introducción

En los últimos años, el acelerado desarrollo de la tecnología digital ha impactado positivamente en el avance de la humanidad. La aparición de microcontroladores incrustados en tarjetas de desarrollo y herramientas de software ha traído como consecuencia que actualmente sea más sencilla y rápida la creación de aplicaciones especializadas. Uno de los campos en constante avance es la detección y reconocimiento facial o de rostros. La detección facial es una tarea en la que los investigadores han trabajado desde la aparición de la computadora. Con frecuencia se desarrollan nuevos algoritmos y aplicaciones en esta área cuyo tiempo de respuesta es de milisegundos y que hasta hace unos pocos años tardaban segundos [Huang, 2015]. En el campo de visión por computadora, la detección facial se lleva cabo por medio de algoritmos y procedimientos cuyo objetivo es identificar la ubicación y tamaño de rostros humanos dentro de una escena o imagen digital.

Se detectan las características faciales y se ignora el resto de la imagen. La detección de rostros es la primera etapa para el proceso de reconocimiento.

Durante las últimas décadas se ha propuesto una cantidad considerable de técnicas y métodos de detección facial que pueden clasificarse en cuatro tipos principales, basados en: el conocimiento, los caracteres invariantes, las plantillas y la apariencia o aspecto. Este último es el que se utiliza más en la actualidad, ya que su eficiencia se basa en no ser necesario conocer las características del rostro a detectar. Aplica los conceptos de entrenamiento y de aprendizaje y se obtienen mejores resultados en función de la cantidad de imágenes que usa para la detección.

El objetivo de este trabajo fue realizar un sistema de control de acceso a la sala de equipos de un centro de datos usando reconocimiento facial. El sistema es de tamaño compacto, confiable, de respuesta rápida y fácil de instalar y operar. Recibe las imágenes de los rostros de usuarios autorizados a entrar desde la aplicación que se ejecuta en una computadora instalada en la oficina de control. Cuando el sistema reconoce el rostro del usuario autorizado debe activarse el actuador de la puerta. En caso de que un usuario no autorizado alcance la cantidad máxima de intentos configurada, se transmite un mensaje de alerta de WhatsApp al teléfono móvil del responsable de la seguridad del centro de datos.

Una de las partes más importantes del desarrollo en este trabajo fue elegir el algoritmo de detección de rostros más apropiado. Se usó el algoritmo de Viola-Jones, el cual es uno de los más comunes de su tipo por su eficiencia para implantar el método basado en aspecto para detectar rostros. Es conocido también como plataforma para detección de objetos Viola-Jones o clasificador Haar. Fue propuesto en 2001 por Paul Viola y Michael Jones. Usa la función matemática, o Wavelet Haar, propuesta por Alfred Haar en 1909 [Viola, 2001].

Se determinó usar este algoritmo por las siguientes razones:

- Demanda menos recursos computacionales y tiempo de procesamiento que otros.
- Las diferentes aplicaciones realizadas han mostrado que presenta una probabilidad de verdaderos positivos de 99.9% y una probabilidad de falsos positivos de 3.33% [Liu, 2016].

- Se basa en el uso de un entrenador AdaBoost y un clasificador en cascada que utilizan la imagen en escala de grises para discriminar de manera rápida las regiones que no contienen el rostro. El clasificador utiliza las características Haar.

El algoritmo de Viola-Jones emplea la extracción de características a partir de imágenes integrales por medio de las máscaras de Haar y consiste en tres fases: la generación de la imagen integral a través de un filtro, la extracción de características y la clasificación en cascada, como se indica en el diagrama de bloques de figura 1.

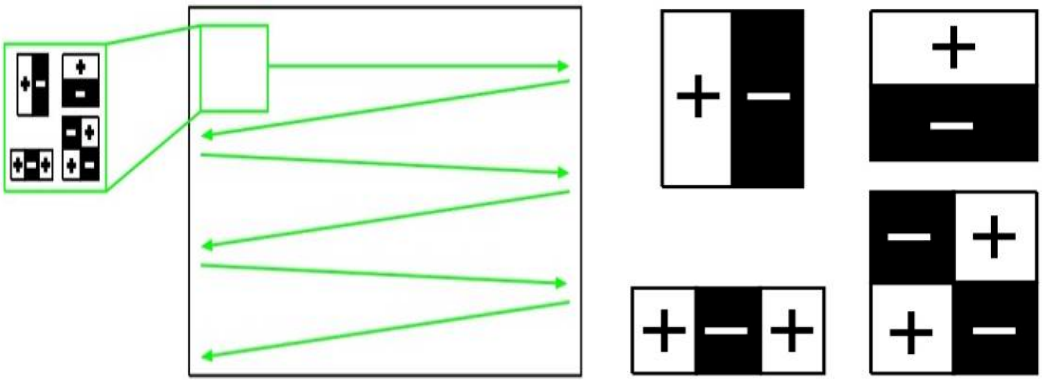
Figura 1 Fases del algoritmo de Viola-Jones.

En la primera etapa se transforma la imagen original a una imagen en escala de grises para obtener la imagen integral. Para realizar esto, se divide la imagen en regiones rectangulares, o ventanas de búsqueda o de detección, de diferente tamaño y las ventanas se aplican a la imagen de forma deslizante [Murphy, 2017].

En la segunda fase, la plataforma busca y extrae características que implican el cálculo de sumas de pixeles en áreas rectangulares de la imagen, similar a las funciones básicas de Haar. Esto se fundamenta en que los rostros humanos comparten propiedades similares que pueden obtenerse usando características de Haar, similares a los wavelet de Haar, las cuales consisten de la diferencia de intensidades luminosas entre regiones rectangulares adyacentes de la imagen. Cada ventana deslizante aplicada a la imagen está formada por una cantidad de rectángulos cuyo valor escalar consiste de la suma de pixeles de cada rectángulo [Ranftl, 2017]. En la figura 2 se muestran cuatro tipos de características usadas en la plataforma y la manera de aplicar las ventanas deslizantes.

A partir de lo anterior, se pueden obtener las características faciales expresadas como gradientes orientados de intensidades de pixeles que indican la ubicación y tamaño de los ojos, del puente de la nariz y la boca, entre otras. En la figura 3 se

muestran las características de Haar, similares al puente de la nariz y a la región de los ojos, aplicadas a un rostro.



a) Características para el rostro *b) Aplicación de la ventana deslizante*

Figura 2 Cuatro tipos de características usadas en la plataforma.

Figura 3 Características similares a la nariz y región de los ojos aplicadas a un rostro.

El valor de una característica compuesta por dos rectángulos es la suma de la intensidad de los pixeles en la parte clara menos la suma de la intensidad de los pixeles en la parte oscura. Los rectángulos tienen la misma área, forma y son adyacentes.

A partir del valor de las características calculadas previamente se usa el principio de que siempre, al menos un área rectangular es adyacente a otra. La característica de dos áreas rectangulares puede calcularse con seis puntos o referencias de la imagen. La característica de tres áreas rectangulares puede calcularse usando ocho puntos y la de cuatro rectángulos se calcula con nueve puntos, con lo cual se obtiene la imagen integral del rostro.

Sin embargo, en una ventana de tamaño común, de 24x24 pixeles existen 162,336 características posibles, de manera que evaluar una imagen puede requerir demasiado tiempo y recursos de una computadora: Por esta razón, en la tercera etapa de la plataforma se utiliza una variante del algoritmo de aprendizaje AdaBoost para entrenar los clasificadores que seleccionan las mejores características [Omidyeganeh, 2016].

Aplicar en cascada todas las características es suficiente para determinar si en la imagen, o porción de ésta, se encuentra el rostro. Con la aplicación de estas características se pueden clasificar las diferentes secciones de la imagen, logrando un alto rendimiento en la detección con una discriminación rápida en las primeras etapas [Kamarol, 2016].

Aunque los dos algoritmos más usados en reconocimiento facial son el de Viola-Jones y los basados en Haar, existe una cantidad importante de trabajos y algoritmos. Las investigaciones recientes se han concentrado en dos líneas principales: 1) Algoritmos de reconocimiento heterogéneos (HFR-Heterogeneous Face Recognition) y 2) Métodos de aprendizaje de características faciales.

La primera línea aborda la restricción de trabajar exclusivamente con imágenes con el rostro alineado hacia el frente. Realiza reconocimiento de rostros en video considerando el difuminado, expresiones, variaciones de posturas, obstrucciones y escasa iluminación [Ding, 2018]. Esto se ha logrado con técnicas que usan redes neuronales convolucionales (CNN-Convolutional Neural Networks) entrenadas y matrices de bibliotecas de características [Moeini, 2015] [An, 2019] [Punnappurath, 2015]. Otras técnicas utilizan el concepto de ángulo de fuerza gravitacional local (LGFA-Local Gravitational Force Angle), el cual considera parte de la reflectancia producida por el efecto de textura de los pixeles de la imagen [Roy, 2016]. Métodos más sofisticados se basan en imágenes en 3D para llevar a cabo rotaciones del rostro enfocadas a la región periocular donde se conservan las características más estables y discriminativas del rostro humano, excluyendo otras regiones susceptibles a expresiones [Juefei-Xu, 2015]. Las imágenes en 3D se emplean también en la reconstrucción y alineación del rostro a través de la regresión en cascada [Liu, 2020] o para corregir problemas geométricos de distorsión [Liu, 2018].

En la segunda línea, los trabajos más recientes proponen algoritmos basados en: métodos de aprendizaje de características y descriptores binarios de rostros usando patrones binarios locales (LBP-Local Binary Patterns) para la extracción de vectores de diferencia de pixeles [Lu, 2015]; redes neuronales convolucionales para aprendizaje de características invariantes del rostro a partir de imágenes capturadas con luz visible y cercana al infrarrojo (NIR-Near InfraRed) [He, 2019]; métodos basados en clasificadores usando regresión ortogonal para manejar las variaciones de posiciones del rostro [Tai, 2016] y reconocimiento a partir de imágenes parciales de rostros ocultos por objetos y dificultad para obtener la imagen completa [Weng, 2016].

Otras investigaciones han trabajado con: reconocimiento de imágenes de iris y huella digital para evitar intentos de acceso fraudulentos [Galbally, 2014]; técnicas basadas en cámaras de campo de luz (LFC-Light Field Camera) que registran tanto la dirección de cada rayo de luz que incide en ella como la intensidad de la luz para capturar múltiples profundidades de la imagen [Raghavendra, 2015] y métodos para detección de falsificación de imágenes de rostros [Chen, 2019].

Para implantar el algoritmo de Viola-Jones en este trabajo, se determinó emplear OpenCV (Open Source Computer Vision Library). OpenCV es una biblioteca de funciones de visión artificial en tiempo real y aprendizaje de máquinas de código abierto y acceso libre bajo licencia BSD. Es una herramienta multiplataforma con más de 2,500 funciones que incluyen el estado del arte de algoritmos de visión por computadora.

Para lograr un sistema compacto, se optó por utilizar en este trabajo la tarjeta de desarrollo Raspberry Pi 4, fundamentalmente por las razones siguientes:

1. Es una tarjeta de reciente creación.
2. Usa el sistema operativo Raspbian en el cual puede usarse OpenCV.
3. Es la tarjeta de su tipo que cuenta con la mayor cantidad de recursos.
4. Ejecuta programas en Python para el cual existe una cantidad importante de bibliotecas de código abierto para diversas aplicaciones. El sistema operativo Raspbian, que se ejecuta en el hardware de las tarjetas Raspberry, está basado en Debian e incluye más de 35,000 paquetes de software.

Las aportaciones de esta aplicación son las siguientes:

- Resuelve una necesidad real usando tecnología de reciente creación y software de uso libre, lo cual redujo el costo del sistema y el tiempo de desarrollo.
- Se implantó el algoritmo de reconocimiento usando Python, OpenCV y Raspbian en un sistema embebido, logrando minimizar el tamaño, costo y tiempo de respuesta de 400 ns para una aplicación de tiempo real.

2. Métodos

En la oficina de control y monitoreo del centro de datos se encuentra instalada una computadora con su propia aplicación que registra las altas, bajas y cambios y captura de fotografías de usuarios autorizados a entrar. Esta aplicación no fue realizada ni se modificó en este trabajo. En la figura 4 se muestra el diagrama de flujo se los procesos que describen la operación del sistema.

El desarrollo del sistema se dividió en dos partes: el sistema digital y la programación del módulo de reconocimiento.

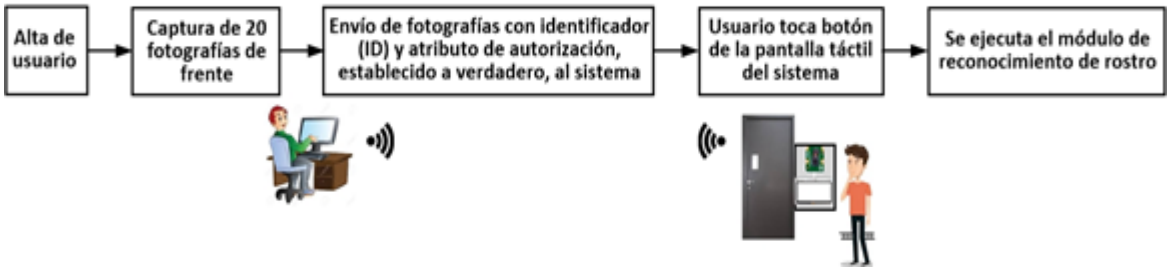


Figura 4 Diagrama de procesos del funcionamiento del sistema.

El sistema digital del módulo de reconocimiento

La arquitectura del módulo de reconocimiento está formada por los elementos siguientes: la tarjeta Raspberry Pi 4, la cámara de video, la pantalla táctil y la interfaz eléctrica, como se indica en el diagrama de bloques de la figura 5.

Los principales recursos con los que cuenta la tarjeta Raspberry Pi 4 son los siguientes: procesador Cortex-A72 de cuatro núcleos de 64 bits a 1.5 GHz; memoria RAM de 4 GB, interfaces Gigabit Ethernet, WiFi 2.4 GHz y 5.0 GHz IEEE 802.11ac

y Bluetooth Low Energy; 5 puertos USB; 40 terminales de propósito general (GPIO-General Purpose Input Output), interfaz serie para cámara (CSI-Camera Serial Interface), 2 puertos micro-HDMI y ranura para tarjeta de memoria micro SD para el sistema operativo y datos.

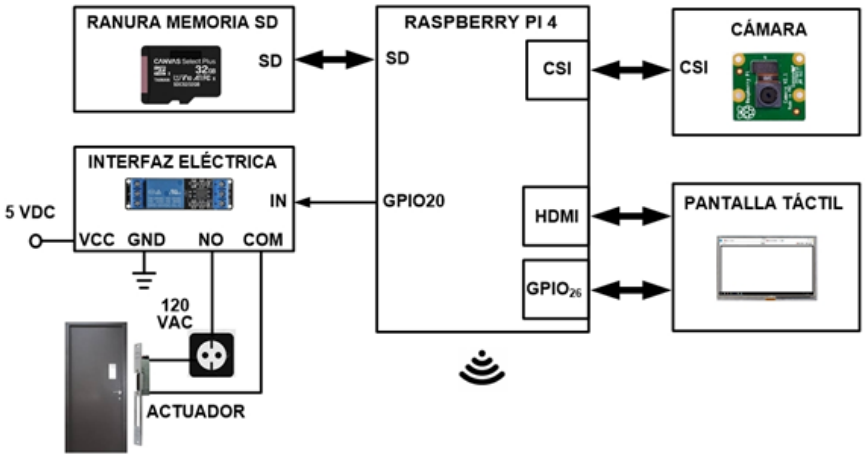


Figura 5 Arquitectura del módulo de reconocimiento de rostros.

Para capturar la imagen del rostro del usuario que intenta acceder al centro de datos se utilizó el módulo de cámara V2 para Raspberry Pi. La resolución de esta cámara es de 8 Megapixeles usando un sensor Sony IMX219. Proporciona las imágenes en archivos de diferente formato, entre ellos JPEG. La cámara se conectó al puerto CSI de la tarjeta Raspberry Pi 4. Para la captura de las imágenes desde el programa en Python se utilizó la biblioteca PiCamera.

Para implantar la interfaz de usuario en el módulo de reconocimiento se usó una pantalla táctil de 5 pulgadas y resolución de 800x480 pixeles. Esta pantalla se conectó a un puerto HDMI y a 26 terminales GPIO de la tarjeta Raspberry Pi 4.

En la ranura de memoria micro SD de la Raspberry Pi 4 se utilizó una memoria de 32 GB. En ella se almacenan la imagen del sistema operativo y las fotografías transmitidas por la computadora de la oficina de control. A esta memoria se le aplicó formato FAT32.

La interfaz eléctrica entre la tarjeta Raspberry Pi 4 y el actuador de apertura de la puerta del centro de datos se implantó usando el módulo de relevador de un canal de 5 V/125 VAC-250 VAC. Este módulo integra un relevador SRD-05VDC-SL-C de

un polo dos tiros y un opto-acoplador para aislar el circuito digital, en este caso la tarjeta Raspberry Pi 4, del sector de potencia. El relevador del módulo se alimenta con 5 V. Para abrir la puerta del centro de datos, la entrada IN del módulo de relevador se activa por medio de una terminal GPIO de la tarjeta Raspberry Pi 4, configurada como salida. El actuador de la puerta está conectado a la salida normalmente abierta (NO) del relevador.

La programación del módulo de reconocimiento

La programación de este módulo se realizó usando Python y OpenCV 4.2.0. Se instaló el paquete *libtbb-dev* para usar en la programación el soporte multi-núcleo de OpenCV. Inicia configurando e inicializando la terminal GPIO conectada a la interfaz eléctrica, la interfaz WiFi, la cámara de video y la pantalla táctil.

A continuación, abre un socket para recibir en el puerto 5000 la información de la computadora de la oficina de control. Esta información se almacena en la memoria SD organizada en directorios, uno para cada ID de usuario. En cada directorio se almacenan los archivos de las fotografías capturadas y un archivo que contiene el valor del atributo de autorización del usuario. Existe también un directorio de accesos, o bitácora, el cual contiene un archivo para cada día en el cual se registra la hora y el ID de usuarios que han intentado acceder de manera exitosa y no exitosa.

Posteriormente, el programa muestra en la pantalla táctil la interfaz de usuario. Ésta despliega al principio el mensaje que indica que se debe presionar el botón de inicio del reconocimiento.

Una vez que el usuario presiona el botón, la interfaz le indica que debe colocarse de manera frontal a la cámara para capturar la fotografía. Acto seguido, el programa arranca el proceso de reconocimiento, el cual captura la imagen del rostro en un archivo JPEG usando las funciones *camera.start_preview()* y *camera.capture()* integradas a Raspbian.

Los clasificadores de OpenCV que implantan el algoritmo Viola-Jones trabajan con imágenes en escala de grises, razón por la cual la imagen es filtrada a escala de grises y dimensionada para obtener el cuadro del rostro del usuario y la imagen

integral a través del uso de las funciones de OpenCV cv2.cvtColor() y cv2.resize(), respectivamente. A continuación, el programa carga el clasificador en cascada para detectar y ubicar el rostro por medio de las funciones:

- *cv2.CascadeClassifier(haarcascade_frontalface_alt2.xml)*
- *haar_cascade.detectMultiScale(img,scaleFactor,MinNeighbors,MinSize)*

En esta parte del programa, OpenCV integra clasificadores en cascada entrenados para detectar rostros, ojos, nariz, boca, cuerpos completos, sonrisas y placas de autos, entre otras cosas.

La primera función carga el clasificador entrenado de OpenCV *haarcascade_frontalface_alt2* y la segunda obtiene las coordenadas de la ubicación del rostro del usuario que intenta acceder usando el clasificador anterior en la imagen capturada y un archivo tipo YML (YAML-Ain't Markup Language). Este último archivo es resultado del proceso de entrenamiento de fotografías de usuarios registrados cuya operación se explicará más adelante.

El siguiente paso en el programa es intentar reconocer el rostro utilizando la función *confidence=predict(x,y)*. Esta función usa el clasificador cargado previamente, el archivo YML y las coordenadas del rostro detectado. Retorna el ID del usuario reconocido en las fotografías entrenadas y el porcentaje de confianza o coincidencia del reconocimiento, siendo 0% la confianza perfecta.

En la implantación de este trabajo se usó un valor de confianza menor igual a 60% para considerar que el rostro del usuario ha sido reconocido.

Si el usuario no es reconocido, se muestra el mensaje correspondiente en la interfaz de usuario y se registra el intento en el archivo de la bitácora del día. Si el rostro es reconocido, el programa consulta en el archivo del usuario si el acceso está autorizado. En caso afirmativo, se muestra el mensaje de entrada en la interfaz de usuario, se actualiza el archivo de la bitácora del día y se activa el actuador de la puerta. En caso negativo, si la cantidad de intentos durante el día es igual a 3, se muestra un mensaje de alerta en la interfaz de usuario. Si la cantidad de intentos es igual a 5 se muestra un mensaje en la interfaz, se transmite el mensaje de alerta de WhatsApp al teléfono móvil del responsable de acceso del centro de datos y se

registra el evento en la bitácora. En la figura 6 se muestra el diagrama de flujo usado para realizar la programación del módulo de reconocimiento.

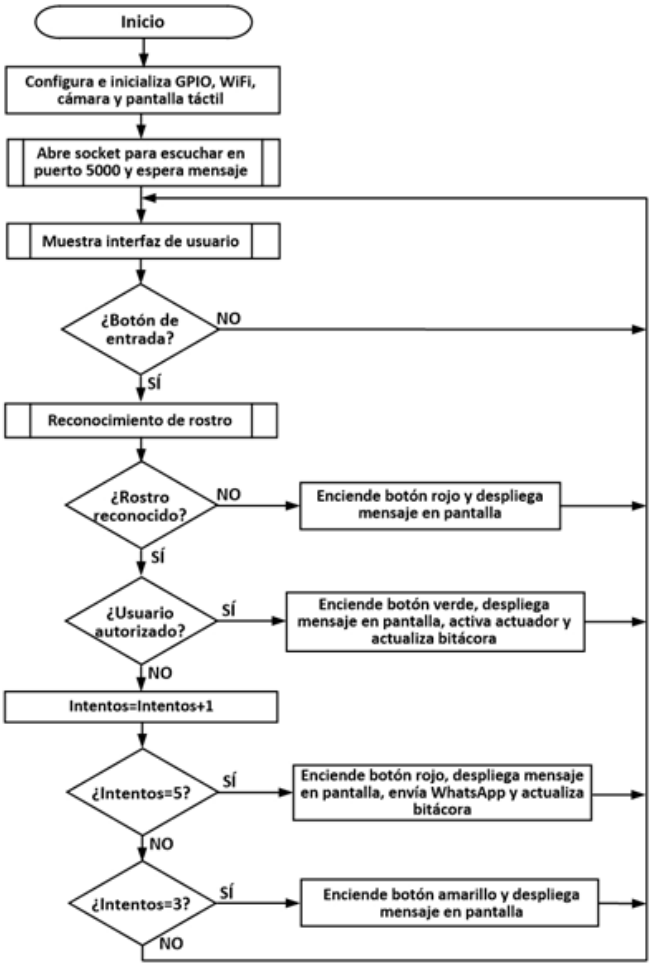


Figura 6 Diagrama de flujo de la programación del módulo de reconocimiento.

Es importante indicar que la función:

haar_cascade.detectMultiScale(img,scaleFactor,MinNeighbors,minSize)

Tiene cuatro parámetros importantes. El primer parámetro, *img*, indica el archivo que contiene la imagen de grises. El segundo parámetro, *scaleFactor*, permite compensar la percepción falsa de tamaño cuando un rostro parece ser más grande que otro al estar más cerca de la cámara. En esta implantación se estableció el parámetro *scaleFactor* con un valor de 1.1, ya que el usuario no se coloca lejos de la cámara. El tercer parámetro, *MinNeighbors*, indica al detector la cantidad de

vecinos que se encuentran cerca del rostro a detectar antes de poder declararlo como encontrado. Se usa para especificar la cantidad de veces que el detector debe recorrer la imagen para reducir los falsos positivos. El último parámetro, *minSize*, especifíca el tamaño mínimo posible del rostro, de forma tal que los más pequeños a este valor son ignorados. En este módulo parámetro *minSize* se estableció a 5.

La rutina que recibe por el puerto 5000 la información de la computadora de la oficina de control realiza las siguientes tareas:

- Recibe por la interfaz WiFi las 20 fotografías capturadas y el ID de los usuarios que causan alta, almacenándolos en el directorio correspondiente.
- Realiza el proceso de entrenamiento de estas fotografías. Esta última tarea se llevó a cabo usando la función de OpenCV *cv2.createFisherFaceRecognizer(trainer.yml)*, la cual genera el archivo del tipo YML. Este archivo contiene un arreglo *numpy* de imágenes e ID de usuarios. Los arreglos *numpy* son multidimensionales y de acceso rápido con los cuales se pueden realizar diversos cálculos usando inclusive el arreglo completo.

La interfaz de usuario se implantó a través del módulo *Tkinter*. Este módulo es la interfaz estándar de Python con el toolkit Tk GUI. Está orientado a objetos y permite la creación de GUI de manera fácil y rápida para el desplegado de ventanas, botones y cuadros combinados, entre otras funciones.

Por último, para acceder las bitácoras de intentos de acceso, almacenadas en la memoria micro SD, se puede realizar una conexión, vía WiFi, al módulo de reconocimiento y descargar a través de ftp los archivos necesarios.

3. Resultados

Se realizaron tres grupos de pruebas cuyo objetivo fue determinar el tiempo que toma al sistema entrenar las imágenes de usuarios registrados, el tiempo de reconocimiento o respuesta y la exactitud del sistema, respectivamente.

En el primer grupo de pruebas se registraron inicialmente 50 usuarios en la oficina de control del centro de datos y las fotografías de ellos se transmitieron al módulo

de reconocimiento. A continuación, se registraron cuatro conjuntos de usuarios adicionales, cada uno de 50, hasta tener 250 usuarios. Se realizó un programa que registró la hora de inicio y terminación de la rutina de reconocimiento en cada conjunto de usuarios. El tiempo del entrenamiento en cada prueba fue: 1.12, 1.75, 2.25, 2.74 y 3.11 minutos, respectivamente, como se indica en la figura 7.

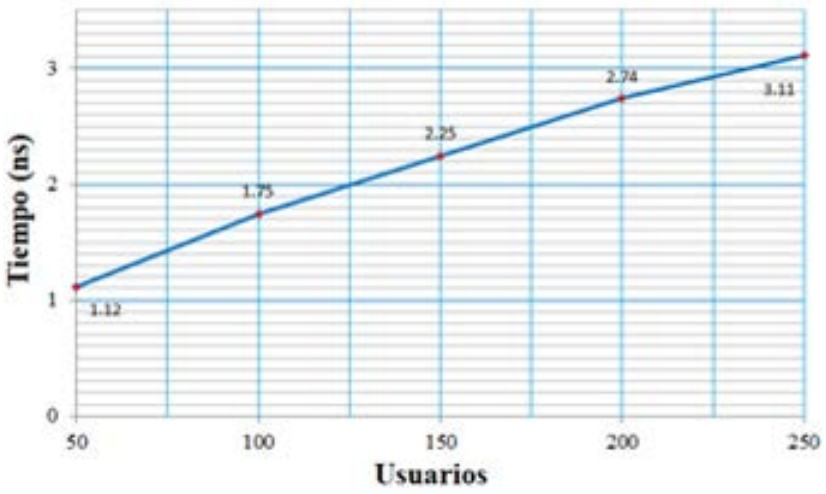


Figura 7 Tiempo de entrenamiento del clasificador.

Tomando en cuenta estos resultados, se observó que el tiempo de entrenamiento de imágenes aumenta proporcionalmente con la cantidad de usuarios registrados, de manera tal que al entrar por primera vez el usuario debe esperar unos minutos para que el sistema pueda reconocerlo. El segundo grupo de pruebas consistió en realizar el reconocimiento de los 50 usuarios de cada uno de los cinco grupos indicados anteriormente. El tiempo que tomó al sistema fue en promedio 225 ns usando 1,000 fotografías entrenadas, 20 de cada usuario, el cual aumentó hasta llegar a 400 ns cuando el sistema tuvo 5,000 fotografías entrenadas, como se indica en la gráfica de la figura 8. En el tercer grupo de pruebas se llevó a cabo a partir de los resultados obtenidos en el grupo anterior: en la primera prueba se reconocieron exitosamente 47 usuarios de 50 registrados, en la segunda 98 de 100, en la tercera 149 de 150, en la cuarta 199 de 200 y en la última 249 de 250 usuarios registrados. En base a estos resultados, se logró una exactitud del 94 %, 98 %, 99.33 %, 99.5 % y 99.6 % en cada prueba.

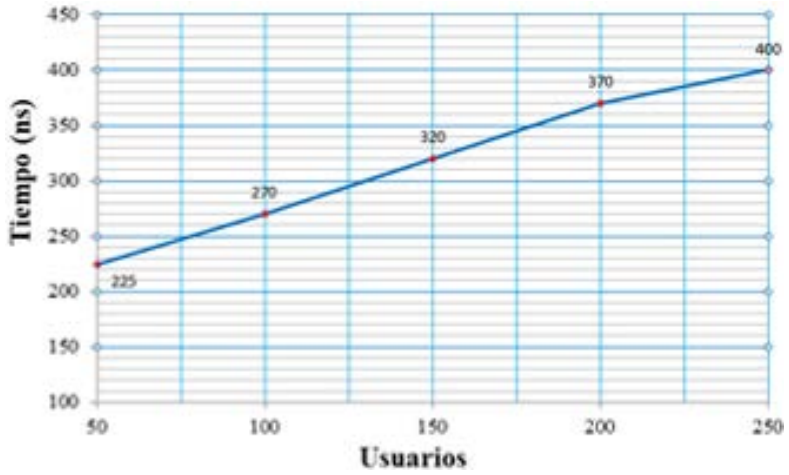


Figura 8 Tiempo de reconocimiento promedio.

Se realizaron adicionalmente pruebas de usuarios registrados que no fueron reconocidos. Esto fue porque se capturaron las imágenes en diferentes condiciones. Por ejemplo, en algunos casos la persona estaba en posición inclinada, tenía anteojos o con escasa iluminación en la oficina donde se le tomó la fotografía.

4. Discusión

Los resultados obtenidos usando el algoritmo Viola-Jones son aceptables para la aplicación realizada por las razones siguientes:

- Cuando se registran usuarios nuevos, deben esperar máximo 3.11 minutos para que el sistema entrene las fotografías y pueda reconocerlos. Esto no es un problema porque el centro de datos no considera autorizar el acceso a más de 100 personas a la vez.
- En caso de necesitar registrar usuarios con escasa iluminación, puede entrenarse el clasificador del sistema con imágenes bajo las condiciones necesarias para generar y usar el archivo XML del clasificador en lugar del que proporciona OpenCV.
- No se incluyó en este trabajo llevar a cabo la aplicación que captura la imagen de los usuarios para su registro, lo cual puede representar una ventaja debido a que puede usarse cualquier aplicación de registro y enviar las fotografías al sistema aquí presentado.

5. Conclusiones

Actualmente, el sistema se usa en el centro de datos y la retroalimentación recibida en cuanto a su desempeño ha indicado que proporciona un control de acceso eficiente con las características siguientes:

- La exactitud y tiempo de respuesta del algoritmo implantado son adecuados para la aplicación de tiempo real realizada.
- El sistema puede usarse en otro tipo de instalaciones e instalarse de manera rápida y sencilla. Únicamente requiere recibir las imágenes de usuarios autorizados a entrar al centro de datos desde la aplicación externa usada para tal fin.
- Para cantidades mayores a 250 usuarios, se estima que el tiempo de entrenamiento del clasificador no será mayor a 5 minutos.
- Finalmente, se tiene planeado a continuación optimizar el acceso a la información de usuarios almacenada en la memoria micro SD usando un manejador de base de datos y adicionar un servidor web en la tarjeta Raspberry Pi 4 para administrar remotamente el registro de usuarios y acceder a las bitácoras del sistema.

6. Bibliografía y Referencias

[1] An, Z., Deng, W., Hu, J., Zhong, Y. & Zhao, Y. APA: Adaptive Pose Alignment for Pose-Invariant Face Recognition. IEEE Access, Vol. 7, 14653-14670, 2019.

[2] Chen, H., Chen, Y., Tian, X. & Jiang, R. A Cascade Face Spoofing Detector Based on Face Anti-Spoofing R-CNN and Improved Retinex LBP. IEEE Access, Vol. 7, 170116-170133, 2019.

[3] Ding, C. & Tao, D. Trunk-Branch Ensemble Convolutional Neural Networks for Video-Based Face Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 40, No. 4, 1002-1014, 2018.

[4] Galbally, J., Marcel, S. & Fierrez, J. Image Quality Assessment for Fake Biometric Detection: Application to Iris, Fingerprint, and Face Recognition. IEEE Transactions on Image Processing, Vol. 23, No. 2, 710-724, 2014.

[5] He, R., Wu, X., Sun, Z. & Tan, T. Wasserstein CNN: Learning Invariant Features for NIR-VIS Face Recognition.IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 41, No. 7, 1761-1773, 2019.

[6] Huang, Z., Shan, S., Wang, R., Zhang, H. & Lao, S. A Benchmark and Comparative Study of Video-Based Face Recognition on COX Face Database. EEE Transactions on Image Processing, Vol. 24, No. 12, 5967-5981, 2015.

[7] Juefei-Xu, F., Luu, K. & Savvides, M. Spartans: Single-Sample Periocular-Based Alignment-Robust Recognition Technique Applied to Non-Frontal Scenarios. IEEE Transactions on Image Processing, Vl. 24, No. 12, pp. 4780-4795, 2015.

[8] Kamarol, S. K., Jaward, M. H., Parkkinen, J. & Parthiban, R. Spatiotemporal feature extraction for facial expression recognition. IET Image Processing, Vol. 10, No. 7, 534-541, 2016.

[9] Liu, F., Zhao, Q., Liu, X. & Zeng, D. Joint Face Alignment and 3D Face Reconstruction with Application to Face Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 42, No. 3, 664-678, 2020.

[10] Liu, M., Jiang, H., Chen, J. & Huang, M. H. Tidal Volume Estimation Using Portable Ultrasound Imaging System. IEEE Sensors Journal, Vol. 16, No. 24, 9014-9020, 2016.

[11] Liu, Y. F., Guo, J. M., Liu, P. H. & Lee, J. D. Panoramic Face Recognition. IEEE Transactions on Circuits and Systems for Video Technology, Vol. 28, No. 8, 1864-1874, 2018.

[12] Lu, J., Liong, V. E., Zhou, X. & Zhou, J. Learning Compact Binary Face Descriptor for Face Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 37, No. 10, 2041-2056, 2015.

[13] Moeini, A. & Moeini, H. Real-World and Rapid Face Recognition Toward Pose and Expression Variations via Feature Library Matrix. IEEE Transactions on Information Forensics and Security. Vol. 10, No. 5, 969-984, 2015.

[14] Viola, P. & Jones, M. H-Rapid object detection using a boosted cascade of simple features. Proceedings of the 2001 IEEE Computer Society

Conference on Computer Vision and Pattern Recognition, 1-9, ISBN: 0-7695-1272-0, HI, USA, 2001.

[15] Murphy, T. M., Broussard, R., Schultz, R. & Rakvic, R. Face detection with a Viola–Jones based hybrid network. IET Biometrics, Vol. 6, No. 3, 200-210, 2017.

[16] Omidyeganeh, M., Shirmohammadi, S. & Abtahi S. Yawning Detection Using Embedded Smart Cameras. IEEE Transactions on Instrumentation and Measurement, Vol. 65, No. 3, 570-582, 2016.

[17] Punnappurath, A., Rajagopalan, A. N. & Taheri, S. Face Recognition Across Non-Uniform Motion Blur, Illumination, and Pose. IEEE Transactions on Image Processing, Vol. 24, No. 7, 2067-2082, 2015.

[18] Raghavendra, R., Raja, K. B. & Busch, C. Presentation Attack Detection for Face Recognition Using Light Field Camera. IEEE Transactions on Image Processing, Vol. 24, No. 3, 1060-1075, 2015.

[19] Ranftl, A., Alonso-Fernandez, F. & Karlsson, S. Real-time AdaBoost cascade face tracker based on likelihood map and optical flow. IET Biometrics, Vol. 6, No. 6, 468-477, 2017.

[20] Roy, H. & Bhattacharjee, D. Local-Gravity-Face (LG-face) for Illumination-Invariant and Heterogeneous Face Recognition. IEEE Transactions on Information Forensics and Security, Vol. 11, No. 7, 1412-1124, 2016.

[21] Tai, Y., Yang, J., Zhang, Y., Luo, L. & Qian, J. Face Recognition With Pose Variations and Misalignment via Orthogonal Procrustes Regression. IEEE Transactions on Image Processing, Vol. 25, No. 6, 2673-2683, 2016.

[22] Weng, R., Lu, J. & Tan, Y. P. Robust Point Set Matching for Partial Face Recognition, IEEE Transactions on Image Processing, Vol. 25, No. 3, 1163-1176, 2016.

[23] Xu, Y., Fang, X., Li, X., Yang, J. & You, Y. Data Uncertainty in Face Recognition. IEEE Transactions on Cybernetics, Vol. 44, No. 10, 1950-1961, 2015.