DETECCIÓN DE PERSONAL NO AUTORIZADO EN EL DEPARTAMENTO DE TI UTILIZANDO REDES NEURONALES CONVOLUCIONALES EN TIEMPO REAL CON RASPBERRY Pi 3 B+

DETECTION OF UNAUTHORIZED PERSONNEL IN THE IT DEPARTMENT USING REAL-TIME CONVOLUTIONAL NEURAL NETWORKS WITH RASPBERRY Pi 3 B +

https://doi.org/10.5281/zenodo.3926937

AUTORES: Miguel Quiroz Martínez[1]*

Galo Valverde Landivar[2]

Jonathan Prieto Villamar[3]

Luis Apupalo Del Rosario[4]

Dirección para correspondencia: mquiroz@ups.edu.ec

Fecha de recepción: 10 / 02 / 2019

Fecha de aceptación: 25 / 06 / 2020

RESUMEN.

El presente proyecto detalla la elaboración de un sistema de reconocimiento facial y envío de alertas mediante correo electrónico email. Para llevarlo a cabo, se utilizó la librería TensorFlow, Numpy y OpenCV. Además, una Pi cámara V2 y un Raspberry Pi 3 B+. Las librerías encargadas para alinear los rostros de las personas a detectar y luego usar estos rostros "alineados" para entrenar la red neuronal que se usó en la aplicación son

[1*] Systems Engineering Faculty Universidad Politécnica Salesiana - Guayaquil, Ecuador
[2] Systems Engineering Faculty Universidad Politécnica Salesiana - Guayaquil, Ecuador
[3] Systems Engineering Faculty Universidad Politécnica Salesiana - Guayaquil, Ecuador
[4] Systems Engineering Faculty Universidad Politécnica Salesiana - Guayaquil, Ecuador

TensorFlow y Numpy. OpenCV para iniciar la captura de video y detección de objetos en tiempo real sobre el ambiente de un Raspberry Pi 3, encargado de procesar y almacenar la base de datos con las fotos de los rostros. Las alertas se envían mediante un correo electrónico (email) por cada rostro no detectado y encendiendo un LED color rojo. Se implementó este sistema de reconocimiento facial con el objetivo de detectar rostros que no se encuentran dentro de la base de datos. El sistema de autenticación se realizó en la Raspberry Pi 3, con la ayuda de las librerías TensorFlow, Numpy y OpenCV, los resultados obtenidos en este estudio fueron favorables luego de una evaluación sobre la base de datos de 10 personas. Este sistema trabaja perfectamente con la base de datos, esta estudia, aprende los rostros captados y los compara con los almacenados en la base de datos; y alertará cuando detecte un rostro no encontrado.

Palabras clave: MML; algoritmos; aprendizaje profundo.

ABSTRACT.

This project details the development of a facial recognition system and the sending of alerts by email email. To carry it out, the TensorFlow, Numpy and OpenCV library was used. Also, a Pi V2 camera and a Raspberry Pi 3 B +. the libraries in charge of aligning the faces of the people to be detected and then using these “aligned” faces to train the neural network that was used in the application are TensorFlow and Numpy. OpenCV to initiate the capture of video and detection of objects in real time on the environment of a Raspberry Pi 3, in charge of processing and storing the database with the photos of the faces. Alerts are sent by email (email) for each undetected face and lighting a red LED. This facial recognition system was implemented with the aim of detecting faces that are not found in the database. The authentication system was carried out on the Raspberry Pi 3, with the help of the TensorFlow, Numpy and OpenCV libraries, the results obtained in this study were favorable after an evaluation based on the data of 10 people. This system works perfectly with the database, it studies, learns the captured faces and compares them with those stored in the database; and will alert when it detects a face not found.

Keywords: MML; algorithms; deep learning.

INTRODUCCIÓN

Cada vez es más difícil ignorar el avance de la tecnología en el campo de la inteligencia artificial, esta va evolucionando y optimizando tareas cotidianas en la vida de las personas, ahora sabemos que los sistemas de seguridad de vigilancia también tienen que formar parte de este avance. Estos sistemas hasta la fecha se han limitado, en su mayoría, a utilizar el mencionado sistema biométrico, usando técnicas de reconocimiento de rostro, lectura de huella digital, retina, palma de la mano, reconocimiento de firma o de la voz(Aguilar, 2006). Los sistemas biométricos pueden presentar problemas al momento de autenticar a los usuarios, logrando dar acceso a personas no autorizadas a áreas restringidas.

Pero ¿por qué realizar un sistema de detección usando aprendizaje profundo en tiempo real dentro del departamento de TI de una empresa? El presente estudio ha ido de alguna manera a mejorar la seguridad e integridad de las personas, dentro de un lugar público o privado, informando de manera automática, utilizando herramientas de reconocimiento facial en tiempo real, algoritmos de aprendizaje profundo, una base de datos con rostros de personas que tienen el acceso a las áreas restringidas, y librerías de Machine Learning, que brinden seguridad al detectar aquellas personas que no corresponden a dichas áreas. Mostrando un porcentaje de acierto eficiente de aquellas personas que tienen permitido acceso, con el fin de evitar que equipos e información delicada sea comprometida con personas no autorizadas, y problemas desde daños a equipos hasta robo de información (Rodríguez, n.d.).

Los departamentos de TI de la empresa no cuentan con un sistema de seguridad y vigilancia fiable dentro de la sala de servidores, utilizan una cerradura común en las puertas o un sistema biométrico que permite el ingreso a los usuarios, estas llaves pueden ser suplantadas por terceros no autorizados permitiendo acceso a las salas, o en caso de los sistemas biométricos, acceso a la persona registrada y no verificar cuantas personas ingresan junto al personal autorizado. Una solución permanente para este inconveniente de acceso a la sala de servidores es implementar un sistema que además de controlar el ingreso, monitoree constantemente que personal se encuentra dentro de él, y evitar depender de llaves para apertura de estas salas.

Por medio del método inductivo para observar, analizar y clasificar las pruebas realizadas (Torrey & Shavlik, 2010), se detecta a las personas que no tienen acceso a las áreas restringidas, usando librerías de aprendizaje automático (MLL) que aseguran rapidez y fiabilidad en el reconocimiento minucioso de las facciones del rostro (Papernot, Goodfellow, Sheatsley, Feinman, & McDaniel, 2016). El lenguaje de programación Python con las librerías OpenCV y TensorFlow, brindan la rapidez y precisión que necesita un sistema de reconocimiento facial en tiempo real (Marzal & Luengo, 2002).

Esto Permite alertar al personal de vigilancia que se encuentre en esa área, por medio de un indicador luminoso, el cual reacciona al momento que se detecte a personas que no cuenta con un permiso a dichas áreas restringidas, evidenciando un proceso ejecutado satisfactoriamente luego de hacer una comparación de los rostros enfocados y los que se encuentran en la base de datos.

Se utilizaron 2 librerías fundamentales, donde cada una de estas hace posible el reconocimiento facial, la librería TensorFlow; está basada en redes neuronales de aprendizaje profundo permitiendo aprender y reconocer objetos en tiempo real (Géron, 2017), y la librería OpenCV que posee algoritmos para implementar un sistema de reconocimiento facial (Bradski & Kaehler, 2008), el cual al detectar un rostro con la cámara establece un patrón en tiempo real; y si el rosto captado se encuentra dentro de la base de datos, dará un porcentaje de acierto.

Para la implementación se utiliza un Raspberry Pi 3 que cuenta con una tarjeta de memoria donde se almacenan los datos con la Red Neuronal ya entrenada de las personas autorizadas (Buhus, Timis, & Apatean, 2016). Busca las fotos dentro de la base de datos y los compara con los rostros captados, si los rostros captados no se encuentran en la base; envía una señal al GPIO 17 alertando mediante un LED rojo que se encuentra conectado al Raspberry Pi 3. En cambio, si los rostros captados se encuentran en la base de datos ya entrenada, este enviará una señal al GPIO 13 conectado al Raspberry Pi 3, alertando mediante un LED verde.

Con estas precauciones, se tiene mayor confianza al momento de manejar la información dentro del área de trabajo, sabremos todo el tiempo quienes son los encargados de ingresar a las áreas restringidas; con este control no existe la preocupación que un tercero pueda dar

algún tipo de inconveniente con la información o con los equipos dentro de estas áreas. Esto dará gran ventaja porque las características de los rostros captados son únicas y no depender de ingresar tarjetas, códigos de acceso y llaves; cosas que pueden ser sustraídas.

MATERIALES Y MÉTODOS

Para el funcionamiento del sistema embebido se precargo una base de datos con las imágenes de 10 usuarios del departamento de TI que tuvieron acceso, con esto se alinearon las fotos recortando solo los rostros y se guardaron en la base de datos procesada dentro del directorio photo_new. Con las fotos procesadas se entrenó la red CNN, estas fotos pasaron por el proceso de las capas de Convoluciones y max pooling que posee la red NCC, logrando identificar elementos del rostro más elaborados de cada imagen, en el transcurso de cada capa.

Durante la captura del video se realizaron los siguientes pasos:

- Segmentación de pixeles: este proceso cambió el video de color a blanco y negro para un mejor reconocimiento respecto al color de piel de las personas detectadas.
- Captación de rostros: en este proceso se captaron los rostros y se usó la CNN ya entrenada para proceder a identificar si es autorizado o no el acceso.

Cuando una persona fue reconocida se encendió el led verde, y en el video apareció el nombre del trabajador.

Si la persona no fue reconocida se encendió el led rojo, y en el video apareció la palabra "Desconocido" y se envió un correo a los supervisores de esa área. Ver figura 1

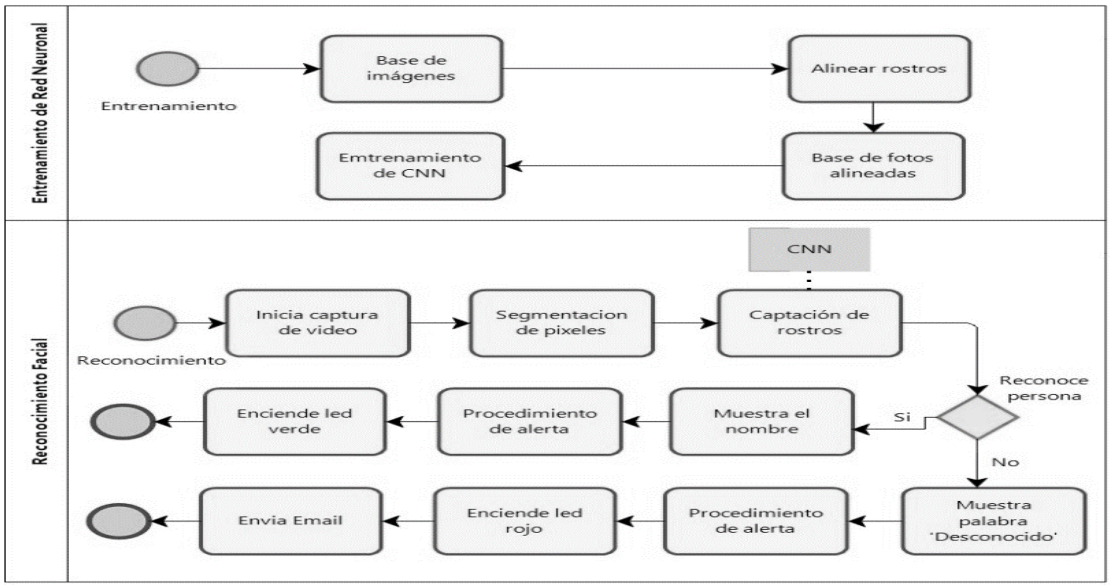

Fig. 1: Procedimiento para alertar al captar un rostro no conocido.

A. Red Neuronal Convolucional

Se utilizó la Red Neuronal Convolucional para el procesamiento de imágenes, se tuvo una mayor eficacia al momento de captar los rostros. Esto se obtuvo gracias a la capa de convoluciones y max pooling que se encuentran entre las capas de entrada y salida (Loncomilla, 2016). Como se muestra en la Figura 2.

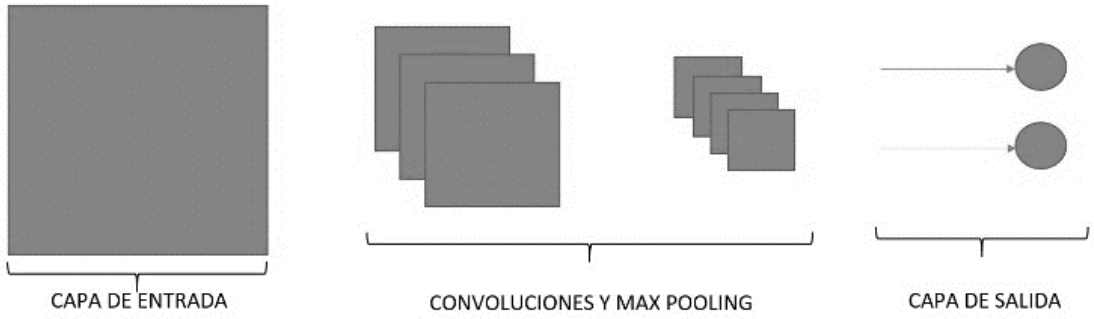

Fig. 2: Capas que contiene la red neuronal convolucional, estas capas aprenden, comparan y predicen.

Cuando se entregó una imagen al algoritmo, entre las convoluciones y max pooling lo que estaba realizando esta estructura fue ir reduciendo el tamaño de la imagen, conforme va avanzando en toda su estructura; cada capa iba reduciendo el tamaño de la imagen (Barrio Algarabel, 2019), va identificando los elementos más importantes y, cada capa tuvo un nivel de abstracción más elaborado conforme más se iba acercando a la capa de salida(Loncomilla, 2016). Como se aprecia en la figura 3.

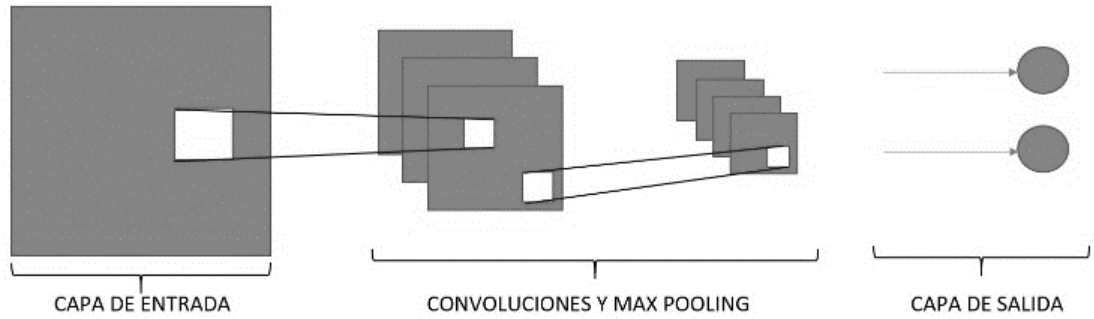

Fig. 3: Procedimiento Convoluciones y Max Pooling que se realiza al otorgar una imagen al algoritmo.

Para este escenario se ha divido en tres capas: la primera capa que se tuvo con convolución y max pooling fue identificar cicatrices, marcas y arrugas, en la segunda capa identificó formas del rostro como las cejas y boca; en la tercera capa los elementos fueron más elaborados, identificando los ojos y nariz (Loncomilla, 2016). Detallado en la Figura 4.

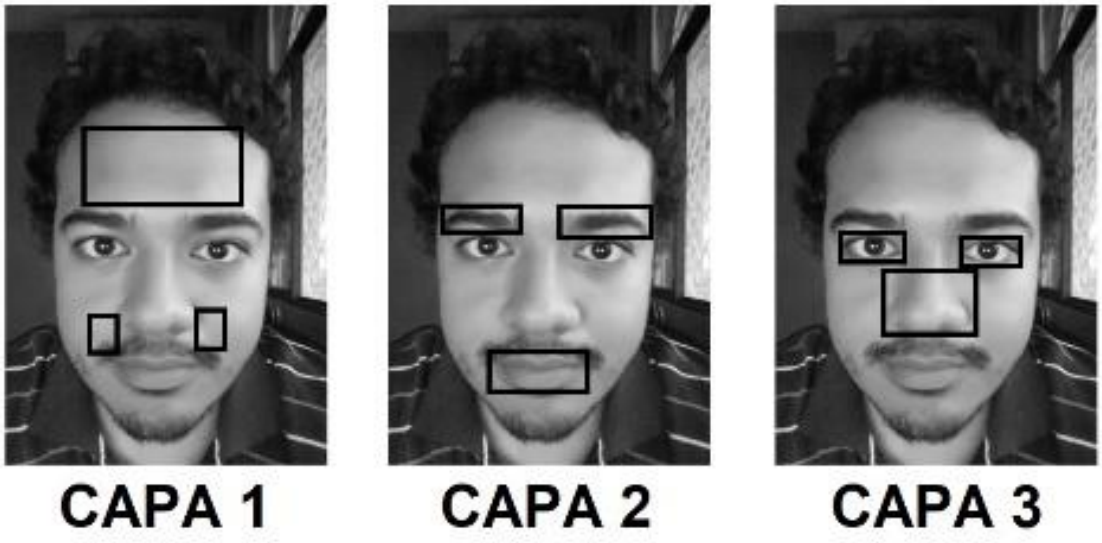

Fig. 4: Proceso de identificación de rostros. En esta imagen se aprecian las partes que va identificando en el transcurso de cada capa.

Cada vez que un usuario se presentó ante la cámara, capturo la imagen y los comparó con los rostros que se encontraban en la red neuronal identificando a la persona que corresponde (Cicero, 2018).

B. Segmentación de pixels

Su función principal fue encender la cámara, con esto se inició la captura del video; cambiando mediante una operación de umbral binario el video de color a blanco y negro evitando cualquier tipo de confusión como el color de piel o con alguna clase de reflejo(Robles, Jiménez, & Pizo, 2013).

Esto sucedió cada vez que una persona se presentaba ante una cámara, identificaba su rostro y pasaba por el proceso de TensorFlow; reconociendo a la persona en tiempo real.

C. Entrenamiento de la Red Neuronal

Para el entrenamiento de la red neuronal se lo realizó en dos pasos:

1. Se procesaron las fotos, para esto fue necesario alinear los rostros. Este proceso consistió en usar las fotos que se tomaron del personal autorizado y por medio del procesamiento del sistema recortando las mismas, dejando solo la cara para una mejor interpretación; guardando el resultado en un directorio distinto para su posterior uso(Salas, 2004). Como se muestra en la Figura 5.

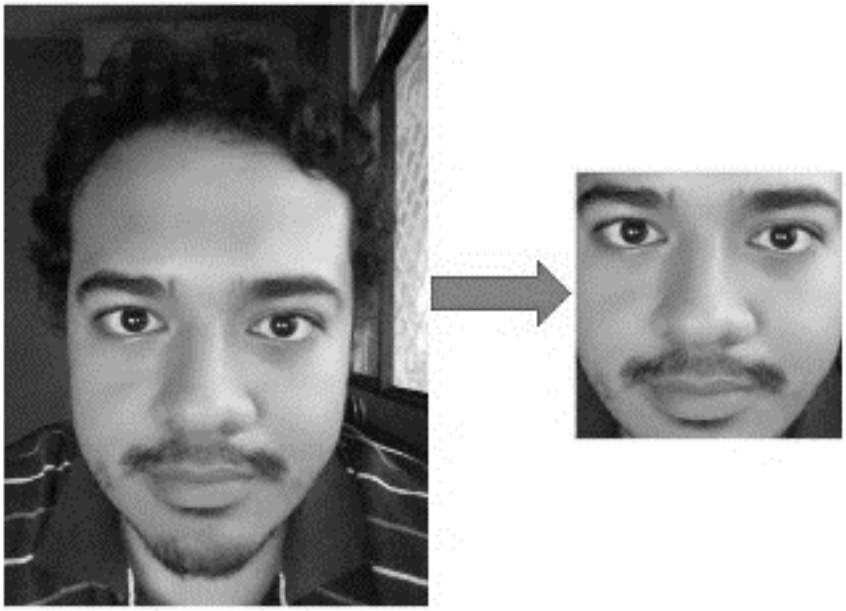

Fig. 5: Proceso que tienen las fotos almacenadas por medio del sistema.

2. Se entrenó un clasificador específico con las imágenes de los rostros procesados y se creó un arreglo con el nombre de cada persona en el orden en que estas fueron procesadas(Salas, 2004). Este clasificador fué el encargado de interactuar con OpenCV para el reconocimiento del personal autorizado. De esta manera se cargó el clasificador y se empezó con el reconocimiento en tiempo real.

D. Procedimiento captación de rostro

Para el procedimiento de captura de video se utilizó el comando cv2.VideoCapture(0) el cual inició la captura de video con OpenCV, para su posterior captación y análisis de los rostros(Olivier et al., n.d.).

Una vez iniciada la captura de video, se extrajo los frames fabricando así los conjuntos de los datos; lo cual se realizó con el comando frame = cap.read()(Mordvintsev & Abid, 2014).

Se transformó la imagen a escala de grises con el objetivo de manejar una matriz de pixeles por imagen, evitando problemas como ruidos y reflejos de luz(Luis & Suárez, 2019).

Se cargó el detector de rostro, este tuvo un modo pre-entrenado y se lo utilizo para extraer solo el rostro. Los cuales almacenaron cada frame en escala de grises dentro de una ruta específica.

E. Procedimiento de alerta

Raspberry pi 3 B+ cuenta con varios puertos que ayudaron a la hora de interactuar con los rostros captados de las personas. Cuando se detectó a una o varias personas que no tienen el debido acceso, alerto por medio del diodo led que está conectado al GPIO 17 del Raspberry Pi 3 B+; enviando un email por cada persona no reconocida. Como se puede apreciar en la Figura 6.

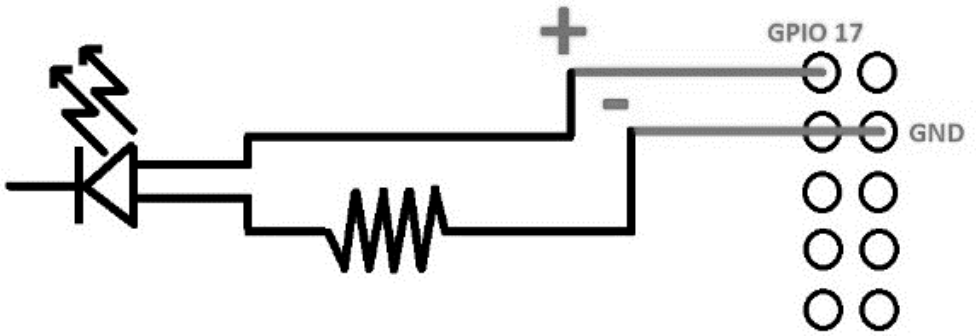

Fig. 6: Mensaje de alerta. En esta figura se aprecia la implementación del diodo led conectado al RaspBerry Pi 3 B+, alertando al momento de detectar a una persona sin el correspondiente acceso.

RESULTADOS

Al momento de capturar un rostro y aprenderlo, se compararán dichos rostros y los almacenados en la base de datos; estos se encuentran en un clasificador específico creado con anticipación en el mismo Tensorflow. El clasificador ayudará a comprender el rostro y así dar la predicción de la persona a la que corresponde con mayor precisión.

Cada vez que se capte un rostro, y si éste no corresponde a una persona dentro de la base, alertara a través de un indicador led rojo, y mostrando en el monitor que la persona no está

identificada, avisando que se acabó de detectar a una persona sin autorización de ingreso a un área restringida.

En base a las experimentaciones realizadas a lo largo del artículo se puede decir lo eficaz que resulta trabajar con las librerías utilizadas, y la ayuda que puede otorgar para futuras investigaciones.

Las librerías OpenCV y TensorFlow que se usaron en este artículo aportaron con la suficiente integración y buena comunicación al momento de trabajar con la base de datos; lo que proporcionó un gran porcentaje de eficacia al momento de dar una predicción.

Haar Cascade librería utilizada anteriormente, necesitaba alrededor de un mínimo de 26 fotos para darnos una identificación de rostro más acertada, mientras que Tensorflow puede trabajar con pocas fotos dando un porcentaje bajo de acierto en el reconocimiento del rostro; incluso el tiempo de reconocimiento trabajando con Tensorflow es más rápido, esto facilita al momento de captar un rostro en movimiento, y no encontrándose solo fijamente. Ver tabla 1.

TABLE I. ALGORITMOS DE RECONOCIMIENTO FACIAL

% de acierto al reconocer a una persona	**Algoritmos**	
	Haar Cascade	***Tensorflow***
4 – 8 fotos	0%	20%
9 – 15 fotos	0%	50%
16 – 30 fotos	76%	87%
Tiempo de detección de rostro	40,24 ms	10 ms

Se trabajó con Pi camera v2 de 8Mp, las cámaras de menor resolución no permiten un adecuado enfoque; recordando que a mayor resolución de cámara mayor será el punto de partida de la captación de los rostros. Ver tabla 2.

TABLE II. COMPARATIVO DE CÁMARAS PARA RECONOCIMIENTO FACIAL

Datos técnicos de la cámara	**Cámaras**		
	Laptop	***Webcam Genius***	***Picamera v2***
Pixeles		5 MP	8 MP
Resolución	720p	640x480	3280x2464
Sensor	FaceTime HD	CMOS	CMOS Sony IMX219PQ
Frecuencia de imagen	30 fps	30fps	30 fps

CONCLUSIONES

El resultado obtenido al trabajar con TensorFlow fue una notable mejoría en comparación de Haar Cascade. Uno de los motivos más importantes de trabajar con Tensorflow es su gran porcentaje de acierto y la rápida respuesta, con un 87% al momento de reconocer, comparar y dar reconocimiento con un tiempo de 10 ms; a diferencia del Haar Cascade con 40,24 ms. Es cierto que el 76% que ofrece Haar Cascade es bueno, pero Tensorflow se encuentra en un porcentaje de más alto de eficiencia y rapidez en cuanto al reconocimiento y predicción, lo cual genera una ventaja bastante evidente contra Haar Cascade.

Al inicio de este artículo las pruebas se realizaron con una cámara tradicional de pc “Apex” y el clasificador Haar Cascade; estos resultados fueron poco satisfactorios debido que existía latencia en los frames captados por la cámara; además el porcentaje de predicción del clasificador era bastante bajo cuando se lo exigía al máximo, es decir, cuando se usaban pocas fotos en la base de datos.

AGRADECIMIENTO

Finalmente, nos gustaría agradecer a la Universidad Politécnica Salesiana, consejo de investigación de sede y al Grupo de Investigación GIIAR por el apoyo brindado para el desarrollo de este proyecto.

REFERENCIAS BIBLIOGRÁFICAS

Aguilar, J. F. (2006). *Adapted fusion schemes for multimodal biometric authentication (esquemas adaptados de fusión para autenticación biométrica multimodal)*. Universidad Politecnica de Madrid.

Barrio Algarabel, A. (2019). *Desarrollo de un sistema de análisis de imágenes médicas basado en técnicas de Deep Learning*.

Bradski, G., & Kaehler, A. (2008). *Learning OpenCV: Computer vision with the OpenCV library*. " O'Reilly Media, Inc."

Buhus, E. R., Timis, D., & Apatean, A. (2016). Automatic parking access using openalpr on raspberry pi3. *Acta Technica Napocensis*, *57*(3), 10.

Cicero, I. E. (2018). *Utilización de redes neuronales convoluciones para la detección de tipos de imágenes*.

Géron, A. (2017). *Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems*. " O'Reilly Media, Inc."

Loncomilla, P. (2016). *Deep learning: Redes convolucionales*. Recuperado de https://ccc. inaoep. mx/~ pgomez/deep/presentations ….

Luis, F., & Suárez, B. (2019). *PATRONES DE CONDUCTA FACIAL, PARA IDENTIFICAR ACCESOS INFORMÁTICOS NO AUTORIZADOS*.

Marzal, A., & Luengo, I. G. (2002). *Introducción a la Programación con Python y C*. Publicacions de la Universitat Jaume I.

Mordvintsev, A., & Abid, K. (2014). Opencv-python tutorials documentation. *Obtenido de Https://Media. Readthedocs. Org/Pdf/Opencv-Python-Tutroals/Latest/Opencv-Python-Tutroals. Pdf*.

Olivier, T., Jiménez-Del Real, M., Escobedo, M., Estrada-Medrano, R., Ochoa, A., Castro, B., … Noriega, S. (n.d.). *Implementación de un modelo de reconocimiento visual para mejorar el modelo adaptativo en niños con daltonismo usando un robot Nao para niños vulnerables en una ciudad inteligente*.

Papernot, N., Goodfellow, I., Sheatsley, R., Feinman, R., & McDaniel, P. (2016). cleverhans v2. 0.0: an adversarial machine learning library. *ArXiv Preprint ArXiv:1610.00768*, *10*.

Robles, L. V. M., Jiménez, E. Q., & Pizo, H. D. V. (2013). Reconstrucción 3D De Objetos Sumergidos En Agua. *Ingeniería*, *18*(2), 5.

Rodríguez, R. H. F. (n.d.). *Escuela de Ingeniería Eléctrica Facultad de Ingeniería*.

Salas, R. (2004). Redes neuronales artificiales. *Universidad de Valparaıso. Departamento de Computación*, *1*.

Torrey, L., & Shavlik, J. (2010). Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques* (pp. 242–264). IGI Global.