



Visión Computarizada para Reconocimiento Facial aplicado a IoT

(Computer Vision for Facial Recognition applied to IoT)

Marcos Almarza

SM PHARMA C.A.

marcosalmarza@hotmail.com

Simon Bravo

SM PHARMA C.A.

simon130598@gmail.com

RESUMEN

La creciente ola de avances tecnológicos en la industria de las telecomunicaciones, interconexiones con múltiples dispositivos, redes de datos masivas e interacciones a distancia han creado nuevas modalidades de trabajo a distancia de forma inteligente y eficiente que nos permiten controlar una gran variedad de dispositivos de forma remota y organizada. Estas tecnologías pueden conocerse como el internet de las cosas o IOT (Internet Of Things), y gracias al constante avance en el campo de aplicaciones disponible para esto, hoy en día disfrutamos de múltiples beneficios como pueden ser los sistemas de monitoreo en casa con la simplicidad de una aplicación telefónica para interconectar cada dispositivo, o las tablas de datos referentes al proceso industrial en plantas de producción, conteniendo información enviada de manera constante por cada dispositivo inteligente dentro del complejo que esté conectado a la red para ser capaz de enviar sus datos recopilados. El objetivo de esta investigación se centró en la realización de un modelo de aplicación versátil para reconocimiento facial computarizado. Esta investigación está fundamentada por los autores L. Bourtole et al. (2021), OpenCV Tutorials y Hernández, R. Fernández, C. Baptista, P. (2014). Esta consta de cuatro fases que explican la descripción, definición de parámetros, diseño y validación por medio de pruebas de la aplicación. Se clasificó como descriptiva proyectiva con diseño experimental. Como resultados se obtuvo la comunicación mediante la plataforma de IOT ThingSpeak, permitiendo a través de un algoritmo de reconocimiento facial, instalado en un Raspberry pi, su posible implementación para tareas como sistemas de vigilancia interconectados que contengan bases de datos para identificación del personal permitido, cumpliendo los objetivos planteados anteriormente.

Palabras clave: IOT; Visión Computarizada



ABSTRAC

The growing wave of technological advances in the telecommunications industry, interconnections with multiple devices, massive data networks and remote interactions have created new ways of working remotely in an intelligent and efficient way that allow us to control a wide variety of devices in an intelligent and efficient way. remote and organized. These technologies can be known as the Internet of Things or IOT (Internet Of Things), and thanks to the constant progress in the field of applications available for this, today we enjoy multiple benefits such as home monitoring systems with the simplicity of a telephone application to interconnect each device, or the data tables referring to the industrial process in production plants, containing information sent constantly by each intelligent device within the complex that is connected to the network to be able to send its data collected. The objective of this research focused on the realization of a versatile application model for computerized facial recognition. This research is supported by the authors L. Bourtoule et al. (2021), OpenCV Tutorials and Hern ndez, R. Fern ndez, C. Baptista, P. (2014). It consists of four phases that explain the description, definition of parameters, design and validation through application tests. It was classified as projective descriptive with experimental design. As a result, communication was obtained through the IOT ThingSpeak platform, allowing through a facial recognition algorithm, installed on a Raspberry pi, its possible implementation for tasks such as interconnected surveillance systems that contain databases for identification of permitted personnel, fulfilling the previously stated objectives.

Keywords: IOT; computer visi n

Introducci n

De acuerdo a los avances tecnol gicos que permiten tener un impacto econ mico y social desarrollado hoy en d a se encuentra un  mbito en com n cuando se habla de las caracter sticas que dichos avances requieren para ser relevantes en cuanto a su operatividad, la cual se refiere a la conectividad. Dicho aspecto ha ganado relevancia en la  ltima d cada de acuerdo a las innovaciones observadas y realizadas para el avance de la Industria 4.0 y su correcta implementaci n de parte de la poblaci n en general para comunicar cada uno de los elementos que se encuentren en nuestro entorno de forma sutil y practica para el d a a d a.

De la misma forma la visi n computarizada forma parte de un conjunto de estrategias de supervisi n la cual se adapta a estos nuevos avances en el  mbito del IoT, ya que este  mbito se encarga de capturar y almacenar un



grupo de imágenes las cuales se transforman en datos que pueden ser analizados por un dispositivo para generar etiquetas o acciones con respecto a las imágenes que se graben en tiempo real. Un ejemplo de ello se puede observar en el ámbito de la vigilancia, el cual adopta dichas técnicas de recolección de imágenes que se relacionan a la visión computarizada y se encarga de encontrar patrones de comportamiento de acuerdo a las horas de video almacenadas en el dispositivo, esto con el fin de identificar patrones que necesiten ser parametrizados por los usuarios.

Una de las herramientas que son utilizadas actualmente en el ámbito de la identificación de parámetros y el procesamiento de imagen son las Redes Neuronales Artificiales, debido a que existen varios tipos de dichas redes (Convolucionales, DNN, entre otros) las cuales requieren de un gran número de muestras registradas de imágenes para su descomposición a datos que serán ingresado a dichos modelos de control para identificar las acciones que se realicen en las imágenes analizadas. Debido a ello, se observa una gran demanda en el ámbito del Big Data y de la recopilación de imágenes mediante Visión Computarizada para el análisis de dichas imágenes y su clasificación, datos que son utilizados para establecer dichas estrategias de control mediante Redes Neuronales.

De acuerdo a lo anteriormente planteado, una de las aplicaciones que se observa en la rama de la Visión Computarizada se plantea en el reconocimiento de rostros y demás rasgos de una persona de acuerdo a una imagen que se analice en tiempo real. Dicha herramienta permite identificar, de acuerdo a un entrenamiento que se haga de acuerdo a las necesidades de los usuarios, patrones en los rasgos faciales identificados por la herramienta, para ser clasificados y contabilizados por un servidor el cual controle dicha información. De acuerdo a ello, se puede contabilizar las veces que se detecta a un hombre pasar por dicha cámara, por ejemplo, así como determinar la cantidad de personas con rasgos físicos similares que se encuentren en un área, o identificar a una persona entre un grupo diverso.

Actualmente en diferentes países a nivel mundial se han desarrollado las tecnologías que utilizan visión computarizada para ser implementadas en diferentes áreas tanto a nivel empresarial como en el hogar, entre ellas se destaca el ámbito de la seguridad en las señalizaciones de tráfico para estudio del comportamiento, flujo de automóviles y supervisión de los peatones en las diferentes áreas, de la misma forma como control de acceso e identificación de personal en diferentes empresas gracias a la supervisión del personal y las áreas por las que transita mediante las imágenes recopiladas por dichos equipos, entre muchas otras aplicaciones.

Dicha tecnología no ha sido implementada a fondo en el país, lo cual representa un atraso tecnológico en cuanto a los avances observados en otras regiones, razones por la cuales se debe fomentar dichas aplicaciones para ser



utilizadas en diferentes sectores en el  mbito local, con la meta de avanzar en cuanto a las aplicaciones de la visi n computarizada en lo cotidiano, utilizando los avances del IoT para intercomunicar dichas im genes captadas en un momento dado con diferentes elementos que se encuentren conectados en malla, para as  obtener una base de datos previamente actualizada las cuales sean recopiladas y analizadas por el usuario.

En la presente investigaci n se plantea como objetivo realizar una herramienta de Visi n Computarizada mediante la aplicaci n de Python y la librer a de OpenCV2 como enlace de entrenamiento para el reconocimiento de patrones e identificaci n de personas en tiempo real mediante una c mara la cual recopile informaci n en todo momento, con la finalidad de elaborar una base de datos en la cual se observe las variables clasificadas para fines de obtener datos ya clasificados que puedan ser ingresados a diferentes controladores modernos. De la misma forma, se plantea la comunicaci n entre dicho modelo y una red IoT desde el punto de vista de protocolos y conectividad.

Fundamentos Te ricos

1. OpenCV2: De acuerdo con lo documentado por el personal de OpenCV Tutorials (2016), OpenCV inicio como parte de Intel en 1999 siendo fundada por Gary Bradsky, y su primer lanzamiento fue en el a o 2000. M s adelante, Vadim Pisarevsky se unio al proyecto para manejar el equipo ruso de programaci n OpenCV. El programa fue utilizado en el desarrollo de Stanley, el veh culo que gan  el Gran Desaf o DARPA 2005. M s adelante, el desarrollo continu  con el apoyo de Willow Garage con Gary Bradsky y Vadim Pisarevsky encabezando el proyecto. Actualmente, OpenCV soporta una multitud de algoritmos relacionados a la visi n por computadora y Machine Learning, expandi ndose diariamente.

OpenCV soporta una variedad de lenguajes de programaci n como lo son C++, Python, Java, etc. Y est  disponible en distintas plataformas incluyendo Windows, Linux, OS X, Android y iOS. Las interfaces para procesamiento gr fico de alta velocidad usando CUDA y OpenCL tambi n se encuentran en desarrollo. Este art culo se desarroll  utilizando la API de OpenCV-Python.

2. OpenCV-Python: Conforme con lo expuesto por el personal de OpenCV Tutorials (2016), esta es una librer a de anclajes de Python, dise ada para la resoluci n de problemas de visi n computarizada. Python es un lenguaje de programaci n de generaci n de prop sito general creado por Guido van Rossum que fue popularizado de forma r pida, debido a la simplicidad que ofrece para lectura de c digo. Permite al programador expresar sus ideas con una menor cantidad de l neas de c digo sin reducir su f cil lectura.

Comparado con lenguajes como C/C++, Python trabaja de forma más lenta, pero puede ser extendido con dichos lenguajes para permitir la escritura de computación intensa en forma de módulos a ejecutar por Python y mejorar el procesamiento. Esto nos permite ventajas como realizar código cuya ejecución sea tan rápida como C/C++ que será ejecutado en el trasfondo y también que es más sencillo crear código en Python, comparado con C/C++. OpenCV-Python hace uso de Numpy, una librería número altamente optimizada para operaciones numéricas con un estilo de escritura como MATLAB. Todas las formaciones estructuradas por OpenCV son convertidas por formaciones de Numpy.

3. Entendimiento de figuras: Según lo explicado por el personal de OpenCV Tutorials (2016), durante la niñez, la mayoría juega o jugo alguna vez con rompecabezas. El juego consiste en armar una imagen de mayor tamaño con piezas pequeñas. Esto nos lleva a una pregunta. ¿Cómo podríamos hacer que esta idea sea ejecutada por una computadora? De cumplir la premisa anterior, ¿Podemos aumentar la escala e imágenes usadas para crear una diferente y mayor en tamaño? Al completar la tarea anterior, podemos pasar al siguiente paso, el cual podría ser, como ejemplo, armar un modelo 3D a partir de la imagen resultante. No obstante, la pregunta principal prevalece. ¿Cómo podemos hacer que la computadora tome una variedad de imágenes y las organice, como identificar cada una de tal manera que pueda realizar la organización?

La respuesta es la búsqueda de patrones y características únicas que sean sencillas de comparar con el resto. Para armar un rompecabezas se busca la continuidad de la imagen y características especiales en cada uno de los trozos de este para completarlo. Debido a esto, resulta tan sencillo como para que menores de edad estén en plena capacidad de poder hacerlo y así entrenar su memoria y poder de reconocimiento de rasgos específicos. Con lo cual extendemos nuestra pregunta en cuanto al número de imágenes, pero se convierte en una más específica, que es señalar dichos rasgos de tal forma que una computadora pueda reconocerlos. En la siguiente imagen encontramos ejemplos de reconocimiento visual.

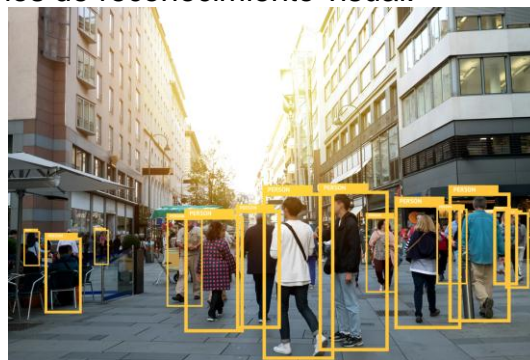


Figura 1. Reconocimiento por visión computarizada.



Fuente: Martínez, S. (2018)

En la imagen anterior se encuentra el reconocimiento de humanos y vehículos de forma individual, por medio de programas de reconocimiento que utilizan los principios propuestos anteriormente. Al encontrar rasgos específicos en imágenes varias y describirlas para el programa, se obtiene una base de datos para comparar con otras imágenes en las cuales existan los componentes encontrados en lo descrito. Así como se puede segmentar visualmente el cuerpo en partes como cabeza, torso y extremidades, es posible hacerse de la misma manera con todo tipo de elementos.

Al tener cada elemento segmentado y organizar la información, podemos describir el conjunto de elementos como un cuerpo completo que representa algo, en este caso humanos y vehículos. Bajo la premisa anteriormente explicada, se puede aplicar una base de datos de reconocimiento a un programa que permita interpretar la información visual obtenida por cámaras para localizar ciertos elementos como colores y formas. Una vez localizados, estos pueden ser utilizados para cualquier fin. Los algoritmos de visión artificial permiten adquirir, procesar y analizar imágenes del mundo real mediante el uso de computadores, como explica (Chandrasekaran et al., 2021).

La Visión Artificial puede ser definida como los procesos de obtención, caracterización e interpretación de información de imágenes tomadas desde un mundo tridimensional a partir de imágenes bidimensionales (Fu, González & Lee, 1988). Por esto se entiende que la visión artificial es la interpretación posible programada en computadores con la finalidad del desarrollo de un modelo digital a partir de las imágenes procesadas como muestra.

Por su parte, los autores García, I. Caranqui, V. (2015) explican que el reconocimiento es el proceso que clasifica en categorías los objetos presentes en la imagen utilizando los descriptores del proceso anterior. Los objetos detectados que presenten descriptores semejantes se agrupan automáticamente en una misma clase o categoría o con una mínima intervención humana. Mediante una organización suficiente de datos recopilados y descripciones correctas para cada muestra, se pueden realizar las tareas de identificación propuestas para un sistema de reconocimiento a través de la visión artificial.

4. IoT: En cuanto a lo que es conocido como internet de las cosas, los autores Castaño, M. Lopez, A. y Villa, P. (2021) el Internet de las Cosas permite unir los mundos virtual y físico mediante diversos conceptos técnicos y sociales, además, es un paradigma que conecta cosas, entidades u objetos en la infraestructura de red global, intercambiando datos de interés para completar varias tareas. En numerosos campos de aplicación pueden ser utilizadas las tecnologías de reconocimiento facial, permitiendo el crecimiento de esta. El continuo desarrollo de nuevos algoritmos y aplicaciones hacen de esta disciplina una tecnología en constante evolución y ha experimentado un

rápido avance en las últimas décadas, así lo demuestran las numerosas investigaciones y publicaciones existentes en la comunidad científica.

Metodología

Para Hernández, R. Fernández, C. Baptista, P. (2014) la metodología representa el estudio de los modos o maneras de llevar a cabo un objetivo, con esto se refiere al estudio de métodos para la realización de un objetivo. El artículo se clasificó como una investigación de tipo descriptiva proyectiva, con un diseño experimental, por tratarse de un programa versátil para ser implementado en aplicaciones IoT de sistemas de reconocimiento facial.

La investigación consta de cuatro (04) fases, las cuales se enfocan en describir el funcionamiento de los componentes electrónicos para la implementación del modelo, definir los parámetros operativos para el reconocimiento facial del modelo, diseñar la visión computarizada para el reconocimiento facial y para finalizar la validación mediante pruebas operativas del comportamiento del modelo realizado.

Resultados

En cuanto al primero objetivo de la investigación, el cual consta de describir el funcionamiento de los componentes electrónicos necesarios para la implementación del modelo, se encontraron varias bibliografías las cuales describen de forma teórica y práctica los elementos requeridos para el diseño de un modelo de visión computarizada, el cual consta en gran parte de ellos en un controlador u ordenador desde el cual se rige el comportamiento del programa, teniendo en cuenta que como requerimiento principal se encuentra que es necesario el lenguaje Python para el desarrollo del programa.

De la misma forma, en cuanto a su conexión a la red y a los demás dispositivos que se encuentren en línea y asociados a la red del usuario, se requiere de un controlador Raspberry Pi el cual consta de diferentes módulos para expansión con la posibilidad de cámara, y al mismo tiempo presenta compatibilidad con diversas plataformas (ThingSpeak, Carriots, Xively, Mosquitto, entre otros) las cuales se encarga de interconectar dispositivos para la implementación de una red de IoT configurada por el usuario. En la Figura 1 se observa los elementos necesarios para la conexión de un Raspberry Pi con el módulo de cámara necesario para la investigación.

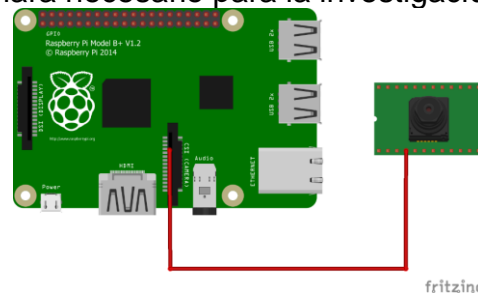


Figura 1. Conexiones Eléctricas utilizando Raspberry Pi.
Fuente: Almarza, Bravo (2021)

De acuerdo a los elementos a utilizar para el desarrollo del algoritmo de Reconocimiento Facial, es necesario establecer una serie de datos de entrenamiento sobre los cuales el controlador se va a regir para él envió de dichos datos mediante ThingSpeak, esto de acuerdo a la necesidad del usuario, razón por la cual se debe establecer una metodología de implementación adaptable en cuanto a los objetos que dicho prototipo reconoce, esto con la finalidad de que dichos datos sean ingresados por el usuario para ser analizados y establecer el entrenamiento de reconocimiento en el modelo establecido.

Por lo mismo, los datos de entrenamiento para el desarrollo del Algoritmo de Reconocimiento Facial se seleccionó un Algoritmo de Reconocimiento el cual identifica a las personas en un área y las clasifica de acuerdo a su nombre. Debido a ello, se plantea un control de acceso sobre un área determinada, por lo cual se procede a recopilar imágenes faciales de aquellos individuos que se encuentren de manera corriente en dicha área, esto para elaborar un conjunto de datos a ingresar en el algoritmo de reconocimiento facial.

De la misma forma, en las librerías de OpenCV2 se encuentran una serie de bases de datos en la cual se definen diversos rostros que serán comparados con las muestras tomadas por el usuario para determinar mediante la visión computarizada que segmento de la imagen pertenece al rostro. De acuerdo a las diferentes imágenes que se encuentran en dichas bases de datos, se procede a realizar un procesamiento de imagen, operándolas hasta obtener la imagen en una escala de grises, dicha característica con la finalidad de obtener valores numéricos de cada uno de los pixeles de acuerdo con la tonalidad de gris que se encuentre en la imagen, siendo en un rango desde $[0,255]$ en cada uno de los valores en los cuales se descomponen la imagen, dicho comportamiento se observa en la Figura 2.

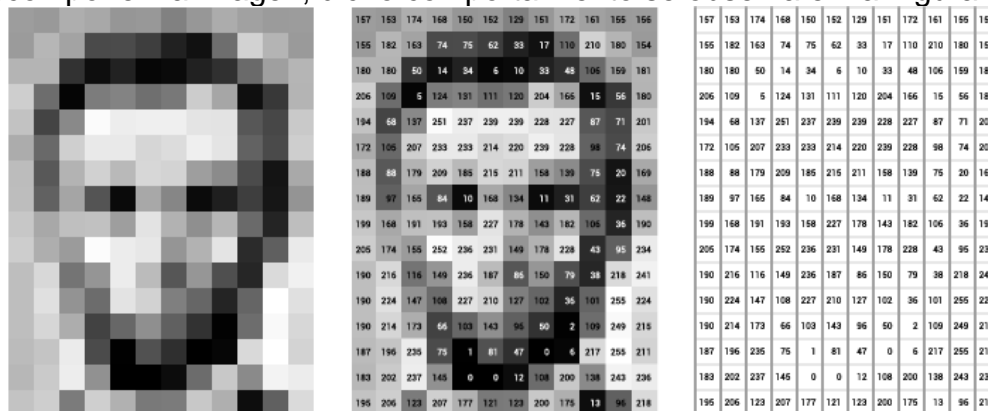


Figura 2. Descomposición de Imagen para Reconocimiento Facial.

Fuente: Almarza, Bravo (2021)

Para identificar al personal que se encuentre en un área determinada, se requiere de una serie de imágenes las cuales sean representativas y se adhieran a la necesidad del usuario, razón por la cual el siguiente paso se trata de recopilar imágenes del personal que se analizara con el fin de entrenar dicho algoritmo para reconocer el rostro de la persona cuando se encuentre enfrente del dispositivo. Por esta razón, en la Figura 3 se pueden encontrar una serie de imágenes de prueba en las cuales se encuentran los rostros de las personas a identificar en el algoritmo para ser ingresadas y analizadas por la programación realizada, de esta forma generar un entrenamiento acorde con el objetivo de la investigación.



Figura 3. Imágenes Ingresadas al Algoritmo.
Fuente: Almarza, Bravo (2021)

De esta forma dichas imágenes recopiladas serán analizadas por el algoritmo planteado en la investigación con la finalidad de obtener un reconocimiento y un control de acceso de acuerdo a los datos ingresados, se tomó una muestra de 30 imágenes las cuales serán analizadas con la finalidad de otorgar mayor precisión sobre el modelo a realizar. De la misma forma, diferentes imágenes de rostros pueden ser ingresados y separados de acuerdo al rostro que se analizara, por lo que dicho etiquetado e ingreso de imágenes este sujeto a las necesidades del usuario.

Para el diseño del algoritmo de reconocimiento facial se debe establecer previamente un entrenamiento, en el cual se analizan las imágenes previamente ingresadas a dicho modelo, el cual realizara comparaciones con las librerías de OpenCV2 para identificar patrones faciales que se encuentren en cada una de las imágenes, y así ser clasificadas para el programa principal. En la Figura 4 se observa el procesamiento de las imágenes, el acondicionamiento que debe llevar cada uno de las carpetas de imágenes a analizar y los comandos de reconocimiento y comparación para ser utilizados en el programa principal.

```

18  for root, dirs, files in os.walk(image_dir):
19      for file in files:
20          #Procesamiento de las Imagenes
21          if file.endswith("png") or file.endswith("jpg"):
22              path = os.path.join(root, file)
23              label = os.path.basename(root).replace(" ", "-").lower()
24              print(label, path)
25              if not label in label_ids:
26                  label_ids[label] = current_id
27                  current_id += 1
28              id_ = label_ids[label]
29
30          #Acondicionamiento de las Imagenes
31          pil_image = Image.open(path).convert("L") # Escala de Grises
32          size = (550, 550)
33          final_image = pil_image.resize(size, Image.ANTIALIAS)
34          image_array = np.array(final_image, "uint8")
35
36          #Reconocimiento de Rostros de Imagenes Ingresadas
37          faces = face_cascade.detectMultiScale(image_array, scaleFactor=1.5, minNeighbors=5)
38
39          for (x,y,w,h) in faces:
40              roi = image_array[y:y+h, x:x+w]
41              x_train.append(roi)
42              y_labels.append(id_)

```

Figura 4. Entrenamiento de Algoritmo de Reconocimiento Facial.
Fuente: Almarza, Bravo (2021)

Es de suma importancia resaltar que con el fin de que la precisión de dicho modelo sea mayor, establecer una serie de imágenes ingresadas que presenten el mismo tamaño (550x550 píxeles en este caso) el cual ayude a determinar con mayor precisión los rasgos faciales. Al finalizar dicho entrenamiento, el programa exporta un archivo sobre el cual se va a registrar el programa principal, siendo esta una base de datos en la cual se compara la imagen recibida de una cámara conectada al controlador con la base de datos previamente entrenada y etiquetada, esto con la finalidad de establecer un modelo que permita efectuar una acción de control cuando reconozca un rostro que se encuentre en dicha base de datos.

En cuanto a el reconocimiento del rostro en una imagen en vivo para poder elaborar dichas comparaciones faciales en el programa principal se utilizó un código programado en Python y utilizando las librerías OpenCV2, las cuales permiten establecer unos parámetros para el desarrollo de la misma, así como ingresar una base de datos previamente elaborada para generar comparaciones y salidas de dicho programa, de acuerdo con la aplicación utilizada por el usuario para la comunicación al entorno IoT (ThingSpeak siendo el más utilizado). En la Figura 5 se observa el código utilizado para el reconocimiento facial y la comparación con la base de datos previamente realizada.

```
cap = cv2.VideoCapture(0)

while(True):
    ret, frame = cap.read()
    gray = cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY)
    faces = face_cascade.detectMultiScale(gray, scaleFactor=1.5, minNeighbors=5)

    for (x,y,w,h) in faces:
        s_gray = gray[y:y+h,x:x+w]
        s_color = frame[y:y+h,x:x+w]
        #Reconocimiento de Cara
        id_, conf = recognizer.predict([s_gray])
        if conf>=40:
            print(labels[id_])
            font = cv2.FONT_HERSHEY_SIMPLEX
            name = labels[id_]
            color = (255, 255, 255)
            stroke = 2
            cv2.putText(frame, name, (x,y), font, 1, color, stroke, cv2.LINE_AA)

        color = (255,0,0)
        stroke = 2
        cord_x = x+w
        cord_y = y+h
        cv2.rectangle(frame, (x,y), (cord_x,cord_y), color,stroke)

cv2.imshow('frame',frame)
if cv2.waitKey(20) & 0xFF == ord('q'):
    break
```

Figura 5. Algoritmo de Reconocimiento Facial.
Fuente: Almarza, Bravo (2021)

Para validar el reconocimiento facial y su correcto etiquetado, se utilizó imágenes en tiempo real para observar el comportamiento del mismo y su clasificación en el programa, ya que esta salida es la que se comunicara con los demás elementos del entorno IoT. En la Figura 6 se puede observar el funcionamiento de dicho programa, el cual detecta el rostro de la persona y dibuja un marco alrededor de ella, sobre el cual se basa para el algoritmo de comparación entre dicha imagen en vivo y la base de datos, además de ello, se observa el identificador en la parte superior, información la cual será la salida del controlador al entorno IoT.



**Figura 6. Salida del Algoritmo de Reconocimiento Facial.
Fuente: Almarza, Bravo (2021)**

De esta forma se valida el comportamiento del sistema de acuerdo a una sensibilidad preestablecida, la cual actúa con el porcentaje de certeza sobre el cual el modelo se guía para determinar la clasificación de las imágenes registradas. Dicho controlador mientras más muestras e imágenes de entrenamiento posee mejora en cuanto a la precisión del reconocimiento y a la rapidez del mismo, de la misma manera se utiliza también los rostros de diferentes personas para distinguir los rastros entre unas y otras, para así tener una mayor precisión del modelo.

Conclusión

Por medio de la información mostrada en los resultados del artículo, con el uso de plataformas como ThingSpeak, un algoritmo que permite realizar el reconocimiento por medio de una base de datos de imágenes referentes y empleando los dispositivos correctos para la programación como puede ser un microcontrolador Raspberry, se concluye como cumplido el objetivo de demostrar las posibilidades de aplicar el uso de visión computarizada para componer y realizar distintas tareas apegadas al reconocimiento facial computarizado, las cuales, al ser aplicadas a los sistemas de seguridad, permiten la identificación de individuos que pueden estar administrados dentro de una base de datos, a través de la cual el mismo sistema sea capaz de reconocer quienes tienen acceso al lugar y que individuos deben ser



reportados por intrusión no autorizada en las proximidades de una forma versátil.

Este mismo reconocimiento puede ser ampliado en funcionalidad más allá de una alarma local, Ya que podría ser conectado para contactar directamente a las autoridades más cercanas, aumentando la seguridad del lugar. De la misma forma, la plataforma de ThingSpeak permite acoplar diferentes dispositivos de IoT en la misma central para mantener un monitoreo de un numero de dispositivos inalámbricos que envíen sus datos a la nube para su recolección, lo cual amplía las posibilidades de implementación de tecnologías IoT sobre un mismo sistema.

Referencias Bibliográficas

Castaño, M. López, A. Villa, P. Review of the use of IoT technologies and devices in physical security systems. (2021). Ingeniería y Competitividad, In press; e30611034. Recuperado en: <https://doi.org/10.25100/iyv.v24i1.11034>

García, I. Caranqui, V. La visión artificial y los campos de aplicación. Vol. 1 Núm. 1 (2015): Tierra Infinita. Recuperado en: <https://doi.org/10.32645/26028131.76>

Hernández, R. Fernández, C. Baptista, P. (2014). Metodología de la investigación 6ta Edición. México D.F.: McGraw-Hill. Interamericana Editores, S.A. De C.V. ISBN: 978-1-4562-2396-0.

Bourtole L. et al. (2021). "Machine Unlearning,". IEEE Symposium on Security and Privacy (SP), pp. 141-159, doi: 10.1109/SP40001.2021.00019.

Martínez, S. (2018): Reconocimiento por visión computarizada. Recuperado en: <https://www.jasminsoftware.es/blog/machine-learning/>

OpenCV-Python Tutorials (2016), recuperado en: https://opencv24-python-tutorials.readthedocs.io/en/latest/py_tutorials/py_tutorials.html