

Diseño de un prototipo para el control de acceso de seguridad mediante reconocimiento facial

Design of a prototype for security access control by facial recognition

Omar Arrieta, Andres Riascos, Jonathan Salcedo, Cristian Rodelo, G. Acosta,
Profesor investigador de grupo: J. Calabria

Universidad Simón Bolívar, Barranquilla-Colombia

Open Access

Publicado:

1 junio de 2022

Correspondencia:

omar.arrieta@unisimon.edu.co

Resumen

El siguiente proyecto solucionará de una forma directa las dificultades de inseguridad y contaminación presentadas por las comunidades, donde se realizará un sistema de detector facial, el cual manejará una base de datos que guardará la información de las personas, permitiendo les entrar desactivando el mecanismo de cerradura por medio de sus rostros. En los procedimientos se hará uso de una plataforma como Python que se comunicará con Arduino el cual manejará el algoritmo encargado de controlar y manejar la base de datos del detector facial, se hará un enlace para un microcontrolador de Arduino, el que será el encargado de manejar el mecanismo de seguridad de nuestro sistema de detector facial.

Palabras claves: Base de datos, mecanismo, comunicación, imagen.

Abstract

The following project will solve in a direct way the difficulties of insecurity and contamination presented by the communities, where a facial detector system will be made, which will manage a database that will store the information of the people, allowing them to enter by deactivating the lock mechanism through their faces. In the procedures will make use of a platform such as Python that will communicate with Arduino which will handle the algorithm responsible for controlling and managing the database of the facial detector, a link will be made for an Arduino microcontroller, which will be responsible for managing the security mechanism of our facial detector system.

Keywords: Database, mechanism, communication, image.

Como citar (IEEE): O. Arrieta, A. Riascos, J. Salcedo, C. Rodelo, G. Acosta & J. Calabria, "Diseño de un prototipo para el control de acceso de seguridad mediante reconocimiento facial", *Investigación y Desarrollo en TIC*, vol. 13, no. 1, pp. 21-35., 2022

Introducción

El presente artículo está centrado en una investigación aplicada. Esto debido, a que se tiene como objetivo resolver un determinado problema, enfocándose en la búsqueda y consolidación del conocimiento para su aplicación, y a su vez, para el enriquecimiento del desarrollo científico. Teniendo en cuenta que esta investigación busca resolver una necesidad social, la investigación que se realizará será la aplicación tecnológica, ahí se abarcará la problemática propuesta y mediante el control de acceso seguro a las organizaciones y se brindará una notable mejora a la sociedad mediante el control de las medidas de bioseguridad en general. Con esto se buscará integrar las distintas áreas de trabajo que permitirán realizar una serie de prototipos adecuados para alcanzar el buen funcionamiento. El reconocimiento facial será trabajado mediante de una sencilla base de datos con la cual se buscará mejorar tanto la seguridad como la contaminación en el entorno donde se trabajará. Esta investigación fue escogida para poder implementar los conocimientos adquiridos en nuestras prácticas de pregrado y para poder colocar un sistema de seguridad más avanzado a la disposición de las personas con menos recursos, brindándoles comodidad, seguridad en su salud y dentro de sus hogares. Debido a situaciones adversas en el mundo con altos índices de inseguridad y de contaminaciones transmitidas por el ser humano ha sido indispensable buscar alternativas las cuales puedan solucionar problemas que las personas muy poco se han dedicado a resolver. El alcance que se tiene propuesto en primera instancia es a nivel nacional, en donde se vea plasmada primordialmente en la zona norte de la ciudad de Barranquilla, en donde muchos hogares y microempresas van a tener la satisfacción de poder implementar este tipo de negocios.

Este sistema de reconocimiento facial [1]. se dividirá en deferentes fases u objetivos alcanzar, en el cual cada uno contará con una serie de pasos para que se pueda llevar a cabo de manera pertinente algunas de estas fases son: los estudios realizados, encuestas a la comunidad, el diseño de prototipos, módulos de auditoría, elaboración de algoritmos y programas de reconocimiento, análisis de materiales, costos. Este artículo basado en la domótica [2]. daremos a mostrar los conocimientos que tenemos sobre el tema, y como mejorar algunos implementos, el de porque nos parece mejor utilizar reconocimiento facial en vez de otro tipo de componentes, y ampliaremos conocimientos con respecto a las deficiencias que tengamos como lo son las comunicaciones entre los diferentes lenguajes de programación y de cómo poder solucionarlos. Las condiciones actuales permiten que las personas (familias y negocios) tengan un sistema automatizado de seguridad para implementación rápida a bajo costo en la que mejorará la vida y la salud de todos.

Estado del arte

El reconocimiento facial desde un principio no solo fue implementado para la seguridad sino para distintos datos y registros de la persona, facilitando y gestionando de mejor manera los datos de cada usuario o individual. Con el trascurso de los años se han utilizado en la seguridad, mas comúnmente en nuestros aparatos móviles en donde con simplemente el rostro tenemos acceso a este [3].

Desde hace un tiempo este sistema de reconocimiento facial se utilizaba para la asistencia a instituciones educativas y demás [4], pero por lo que no fue de mucha eficacia ya que generaba un alto costo y no lograba abarcar los requisitos mínimos para su implementación, por lo que generalizando un poco en el ámbito de las organizaciones estudiantiles, que implemente sistema de reconocimiento facial [5], por esto se incorporó la librería OpenCV para el reconocimiento facial con la función LBPH; y Python, como la fuente de lenguaje de programación, utilizando el modelo

de prototipo. El producto resultante fue un sistema de reconocimiento facial, basada en aprendizaje automático, capaz de registrar las E/S de los empleados de la universidad BICU, donde cada empleado registrado en el sistema es identificado por los rasgos faciales [6].

En otros estudios se tiene para mirar o identificar las emociones de las personas esto con el fin de hacer distintas pruebas y estrategias para modificar y para mejorar los algoritmos en este caso se utilizó como base el sistema de algoritmo básico Eigenfaces o análisis de componente principal, el cual es uno de los modelos de reconocimientos de otros más utilizados; así mismo fue utilizado el lenguaje Python y alguna de sus librerías disponibles como Numpy, OpenCV, Sklearn para su implementación [7].

Los sistemas de reconocimiento facial también ayudan a la automatización de muchos procesos en organizaciones, adicionalmente este proceso le da un plus de seguridad como lo es encendido vehicular mediante el reconocimiento, ayudar a resolver la problemática es esencial, que es el robo vehicular. Se evidenció que por medio del reconocimiento facial se puede obtener la identificación de las personas que son captadas por medio de una cámara, permitiendo el acceso de la persona registrada en el mismo. El prototipo del sistema propuesto brinda una idea de cómo se puede resolver el problema de robo vehicular a través de un programa elaborado en Python para lograr el reconocimiento facial y el encendido del vehículo [8].

Materiales y métodos

Los métodos utilizados fueron los de análisis y prueba en los que se observó cómo sería la mejor manera de realizar un prototipo de un detector de reconocimiento facial a menor precio y con una efectividad alta [11].

Para las pruebas del reconocimiento facial se realizó una puerta a escala en madera, donde se le implementaron los diferentes componentes para las pruebas.

Cámara: Se utilizó para la detección facial una cámara de referencia Lenovo ThinkSmart esta cámara cuenta con una velocidad de fotogramas de 4K con 30 fps, 1080p con 60 o 30 fps, 720p con 60 o 30 fps, 360p con 60 o 30 fps. Campo de visión horizontal 100° / vertical 68° / diagonal 125, HDR para una excelente visualización en condiciones de poca luz.



Imagen 1. Cámara.

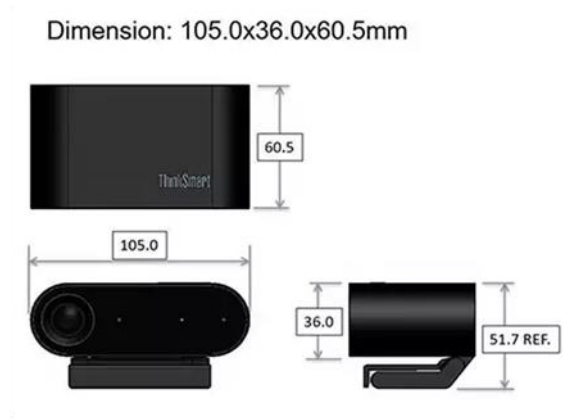


Imagen 2. Dimensiones de la cámara.

Tarjeta de control: para el control se utilizó una tarjeta Arduino uno la cual fue la encargada de controlar el actuador que activará nuestro reconocimiento facial.



Imagen 3. Tarjeta de control (Arduino).

Arduino es utilizado como un microcontrolador, cuando tiene un programa descargado desde un ordenador y funciona de forma independiente de éste, controla y alimenta determinados dispositivos y toma decisiones de acuerdo con el programa descargado e interactúa con el mundo físico gracias a sensores y actuadores.

Motor: Este componente fue utilizado como el actuador que accionaba la cerradura de la puerta para que así se pudiera abrir o cerrar, este motor funciona con un voltaje de 5 a 7 voltios, que va conectado directamente al microcontrolador Arduino.



Imagen 4. Actuador (servo motor).

Resultados

En nuestro sistema de reconocimiento facial se programó en el lenguaje de Python, donde utilizamos la interfaz de spider para plasmar nuestro código, con lo cual utilizamos La interfaz de OpenCV (Visión artificial abierta) que es una biblioteca libre de visión artificial originalmente desarrollada por Intel. Para la detección de movimiento, reconocimiento de objetos, reconstrucción 3D a partir de imágenes, son sólo algunos ejemplos de aplicaciones de OpenCV [12].

```
import cv2
```

```
import os
```

```
import imutils
```

```
personName = 'Andres'
```

```
dataPath = "C:\\Users\\equipo1\\Desktop\\Codigo final\\Data"#Cambia a la ruta donde hayas  
almacenado Data
```

```
personPath = dataPath + '/' + personName
```

```
if not os.path.exists(personPath):
```

```
    print('Carpeta creada: ',personPath)
```

```
    os.makedirs(personPath)
```

```
cap = cv2.VideoCapture(1,cv2.CAP_DSHOW)
```

```
#cap = cv2.VideoCapture('Omar.mp4')
```

```
faceClassif = cv2.CascadeClassifier(cv2.data.harcascades+'haarcascade_frontalface_default.xml')
```

```
count = 0
```

while True:

```
ret, frame = cap.read()
if ret == False: break
frame = imutils.resize(frame, width=640)
gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
auxFrame = frame.copy()

faces = faceClassif.detectMultiScale(gray,1.3,5)

for (x,y,w,h) in faces:
    cv2.rectangle(frame, (x,y),(x+w,y+h),(0,255,0),2)
    rostro = auxFrame[y:y+h,x:x+w]
    rostro = cv2.resize(rostro,(150,150),interpolation=cv2.INTER_CUBIC)
    cv2.imwrite(personPath + '/rotro_{}.jpg'.format(count),rostro)
    count = count + 1
cv2.imshow('frame',frame)

k = cv2.waitKey(1)
if k == 27 or count >= 300:
    break
```

cap.release()

cv2.destroyAllWindows()

1. Segmento 1 del código Python (Rostros.py)

El anterior código corresponde a la pestaña “rostros”, donde se da el tamaño a tomar del rostro, en nuestro caso lo asumimos de 150 x 150. Le decimos que tome 300 capturas del video para la verificación de los gestos al momento de ir a entrenamiento. [13].

Nosotros creamos una carpeta llamada código final, en donde se encuentran las carpetas de reconocimiento facial, rostros, y entrenamiento; también hay una llamada “Data” en esta se guardará las imágenes que posterior mente utilizaremos en nuestro reconocimiento y para el entrenamiento del código.

```
import cv2

import os

import numpy as np

dataPath = "C:\\Users\\equipo1\\Desktop\\Codigo final\\Data"#Cambia a la ruta donde hayas
almacenado Data

peopleList = os.listdir(dataPath)

print('Lista de personas: ', peopleList)

labels = []

facesData = []

label = 0

for nameDir in peopleList:

    personPath = dataPath + '/' + nameDir

    print('Leyendo las imágenes')

    for fileName in os.listdir(personPath):

        print('Rostros: ', nameDir + '/' + fileName)

        labels.append(label)

        facesData.append(cv2.imread(personPath+'/'+fileName,0))

        #image = cv2.imread(personPath+'/'+fileName,0)

        #cv2.imshow('image',image)

        #cv2.waitKey(10)

    label = label + 1

#print('labels= ',labels)

#print('Número de etiquetas 0: ',np.count_nonzero(np.array(labels)==0))

#print('Número de etiquetas 1: ',np.count_nonzero(np.array(labels)==1))

# Métodos para entrenar el reconocedor

face_recognizer = cv2.face.EigenFaceRecognizer_create()
```

```
cv2.face.FisherFaceRecognizer_create()
cv2.face.LBPHFaceRecognizer_create()

print("Entrenando...")
face_recognizer.train(facesData, np.array(labels))

face_recognizer.write('modeloEigenFace.xml')
print("Modelo almacenado...")
```

2. Segmento 2 del código Python (Entrenamiento.py)

En la segunda ventana la tenemos nombrada como “entrenamiento.py” en este segmento del código, entrenamos al código y verificamos todas las imágenes recopiladas en la base de datos, para que el código la pueda reconocer y comparar al mismo tiempo así poder ejecutar el reconocimiento de una manera satisfactoria. [14].

En la tercera pestaña.

```
import cv2
import os
import serial
import time

ser = serial.Serial('COM8', 9600, timeout=1)
time.sleep(2)

dataPath = "C:\\Users\\equipo1\\Desktop\\Codigo final\\Data" #Cambia a la ruta donde hayas
almacenado Data

imagePaths = os.listdir(dataPath)
print('imagePaths=',imagePaths)

face_recognizer = cv2.face.EigenFaceRecognizer_create()
#face_recognizer = cv2.face.FisherFaceRecognizer_create()
#face_recognizer = cv2.face.LBPHFaceRecognizer_create()

# Leyendo el modelo
face_recognizer.read('modeloEigenFace.xml')
```



```
cap = cv2.VideoCapture(1,cv2.CAP_DSHOW)
```

```
#cap = cv2.VideoCapture('Video.mp4')
```

```
faceClassif = cv2.CascadeClassifier(cv2.data.harcascades+'haarcascade_frontalface_default.xml')
```

```
while True:
```

```
    ret,frame = cap.read()
```

```
    if ret == False: break
```

```
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
```

```
    auxFrame = gray.copy()
```

```
    faces = faceClassif.detectMultiScale(gray,1.3,5)
```

```
    for (x,y,w,h) in faces:
```

```
        rostro = auxFrame[y:y+h,x:x+w]
```

```
        rostro = cv2.resize(rostro,(150,150),interpolation= cv2.INTER_CUBIC)
```

```
        result = face_recognizer.predict(rostro)
```

```
        cv2.putText(frame,'{}'.format(result),(x,y-5),1,1.3,(255,255,0),1,cv2.LINE_AA)
```

```
        # EigenFaces
```

```
        if result[1] < 3700:
```

```
cv2.putText(frame,'{}'.format(imagePaths[result[0]]),(x,y-25),2,1.1,(0,255,0),1,cv2.LINE_AA)
```

```
    cv2.rectangle(frame, (x,y),(x+w,y+h),(0,255,0),2)
```

```
    ser.write(b'P')
```

```
else:
```

```
    cv2.putText(frame,'Desconocido',(x,y-20),2,0.8,(0,0,255),1,cv2.LINE_AA)
```

```
    cv2.rectangle(frame, (x,y),(x+w,y+h),(0,0,255),2)
```

```
    ser.write(b'N')
```

```
cv2.putText(frame,'{}'.format(imagePaths[result[0]]),(x,y-25),2,1.1,(0,255,0),1,cv2.LINE_AA)
```

```
(x,y),(x+w,y+h),(0,255,0),2)
20),2,0.8,(0,0,255),1,cv2.LINE_AA)
(x,y),(x+w,y+h),(0,0,255),2)

cv2.putText(frame,'{}'.format(imagePaths[result[0]]),(x,y-25),2,1.1,(0,255,0),1,cv2.LINE_AA)
# cv2.rectangle(frame, (x,y),(x+w,y+h),(0,255,0),2)
20),2,0.8,(0,0,255),1,cv2.LINE_AA)
# cv2.rectangle(frame, (x,y),(x+w,y+h),(0,0,255),2)

cv2.imshow('frame',frame)
k = cv2.waitKey(1)
if k == 27:
    break

cap.release()
cv2.destroyAllWindows()
```

3. Segmento 3 del código Python (Reconocimientofacial.py)

En la siguiente parte del código se hace la verificación de las imágenes, en donde se activa la cámara y se comparará con la base de datos de las imágenes y el rostro de la persona que este enfrente [15].

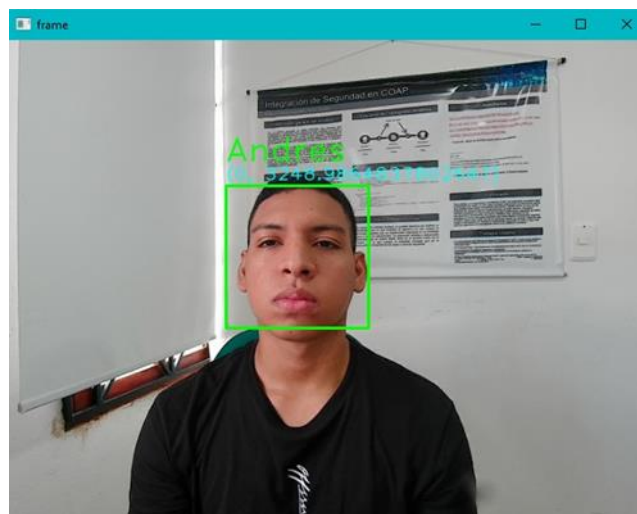


Imagen 5. Reconocimiento de la persona.

El código hace una comparación de las imágenes que tiene recopiladas en su base de datos, al momento de detectar los rostros las analiza y genera una opción de respuesta [16].

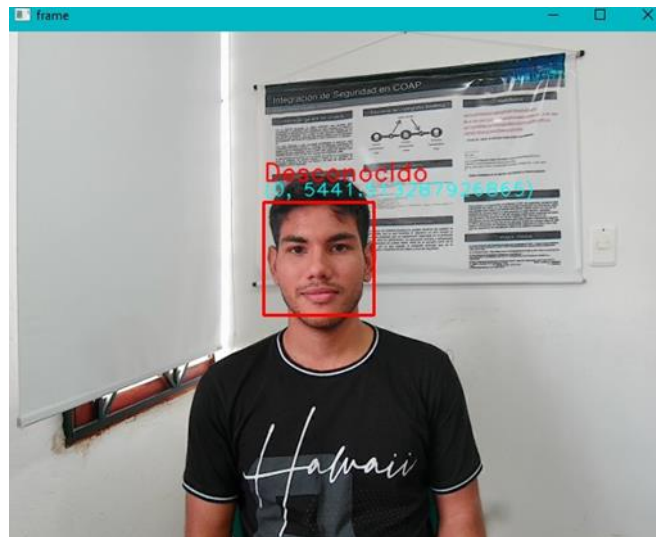


Imagen 6. Reconocimiento desconocido del individuo.

Puede ser negativa o positiva en las que sí reconoce manda la confirmación para desbloquear la cerradura y si no la reconoce se mantiene bloqueada, esto es con la comunicación del puerto serial COM8 de la tarjeta Arduino

Posterior a esto al momento de estar activado la cámara y de reconocer a la persona en función se realizará una comunicación con Arduino, para controlar el servo motor.

```
#include <Servo.h>
```

```
#define SER 13
```

```
Servo servoMotor;
```

```
int option;
```

```
void setup() {
```

```
  Serial.begin(9600);
```

```
  servoMotor.attach(SER);
```

```
}
```

```
void loop() {
```

```
  //Abrir Puerta
```

```
  if (Serial.available() > 0){
```

```
    option = Serial.read();
```

```
    if (option == 'p') {
```

```
      servoMotor.write(110);
```

```
      //delay(2000);
```

```
}  
//Cerrar puerta  
if (option == 'N') {  
  servoMotor.write(20);  
  //delay(2000); [17].  
  
}  
}  
}
```

4. Segmento del código Arduino

La Lógica de programación acá es más sencilla, donde le decimos al microprocesador que si la función option es válida desbloquee la puerta, pero sino manténgala bloqueada [18].

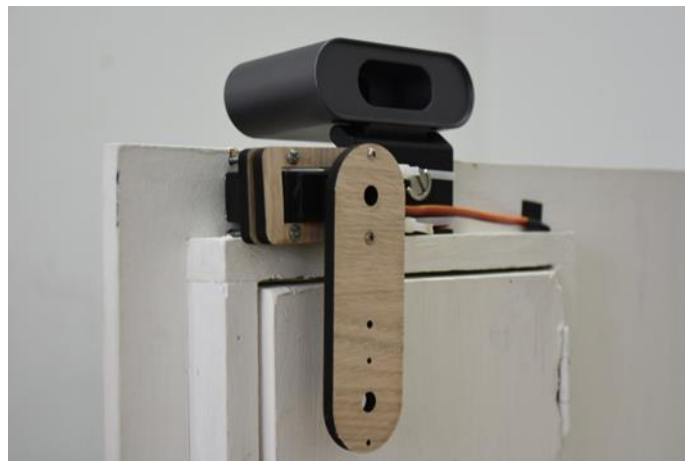


Imagen 7. Cerradura.



Imagen 8. Maketa parte frontal.

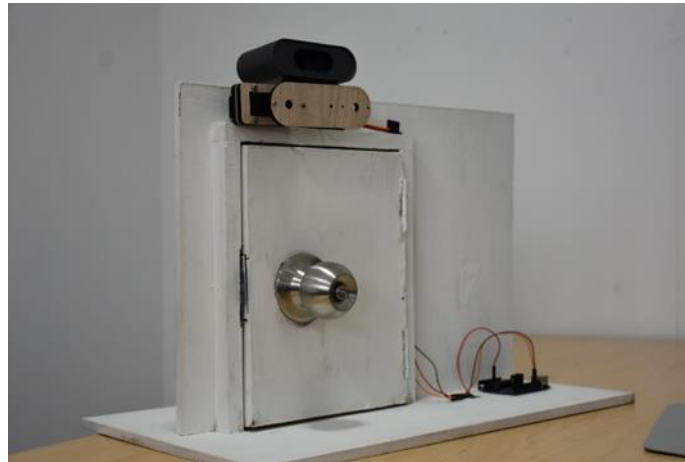


Imagen 9. Maketa de cerradura.

Conclusiones

El objetivo de este artículo está en analizar e implementar un sistema de reconocimiento facial (Domótica) [19] para residencias y microempresas ubicado en la ciudad de Barranquilla, el motivo de este análisis fue el sin número de casos de inseguridad que se vive en el país específicamente en la costa colombiana. Con base a esto, decidimos analizar e implementar el mejor método de seguridad en el cual le permitiera a los usuarios tener un gran porcentaje de confiabilidad y con el menor costo posible, ayudando a que este sea la solución a nuestro problema más práctica y tecnológica posible [20].

A partir de los resultados del desarrollo del proyecto se logró el diseño e implementación del prototipo de control de acceso para el reconocimiento facial, a través de esta aplicación permitió al grupo de trabajo complementar y reforzar los conocimientos adquiridos en programación mediante la utilización de lenguaje de programación de alto nivel (Python), adicionalmente logramos comprender el funcionamiento en general de estos mecanismos en las organizaciones grandes, medianas y pequeñas, de este modo logramos el desarrollo equilibrado de un prototipo el cual cumpla con una característica esencial, como lo es la implementación a bajo costo, una gran ventaja en comparación a las demás sistemas de reconocimiento facial previamente estudiados.

Referencias bibliográficas

1. D. L. Castellano Saavedra y J. D. Alonso Sierra, "Sistema de reconocimiento facial para acceso a viviendas". Universidad Católica De Colombia, Bogotá, 2019.
2. J. Martínez Garcés y J. Barreto Ferreira, "Modelo de planeación para la inversión tecnológica en centros de investigación universitarios". Investigación e Innovación en Ingenierías, vol. 7, n.º 2, 2019. DOI: <https://orcid.org/0000-0002-8120-3285>
3. P. A. Gomes Alzate, "Aplicación móvil de control de acceso por medio de reconocimiento facial". Universidad de Antioquia, Medellín, 2022.
4. V. D. S. C. V. C. D. A. H. & R. J. Suresh, "Reconocimiento facial de un sistema utilizando Python y OpenCV". Quest journals-journal of software Engineering and Simulation, vol. 5(2), pp. 18-29, 2019.

5. Y. J. A. RAMIREZ, L. K. T. KNIGHT y D. M. SAMBOLA, "Sistema de reconocimiento facial de entrada y salida de empleados". Wani, nº 76, 2022.
6. J. G. Steele Garza, "Mediación policial. Un protocolo de prevención del delito y cultura de paz". Justicia, vol. 23, n.º 34, pp. 471–489, jul. 2018.
7. J. J. R. B. P. D. T. M. y. X. J. M. A. Paricela Canazas, "Sistema de reconocimiento de emociones a través de reconocimiento facial utilizando inteligencia artificial". Innov Sofw, vol. 3, nº 2, pp. 140-150, 2022.
8. G. D. Orna Villalta, "Diseño e implementación de un sistema en bebido de reconocimiento facial para el control de acceso usando deep lerni". Quito, 2019.
9. B. XAVIER BECERRA, "Los países con los índices más altos de criminalidad organizada en todo el mundo". La Republica, 27 Abril 2022.
10. DANE, "Estas son las ciudades de Colombia con mayor inseguridad". El espectador, 5 Noviembre 2021.
11. F. N. Giraldo Ramos, "Prototipo de una cerradura electrónica para el reconocimiento facial". Universidad distrital Francisco José de Caldas, Febrero 2016.
12. Morcillo Vizuet F, "Desarrollo de un sistema de reconocimiento facial utilizando Deep Learning con OpenCV". Valencia, 2022.
13. A. Castro, "Algoritmo de reconocimiento facial para la gestión del control de acceso de la empresa Altoque" 2021.
14. R. Duran Viñuelas, "Interfaz de control domótica basada en visión". Universidad de Sevilla, Sevilla, 2020.
15. B. y. c. j. J. Chalen pacay, "Desarrollo de un prototipo de reconocimiento facial basado en machine learning para detectar estado de somnolencia de conductores en una cooperativa de transporte". Universidad de Guayaquil, Guayaquil, 2020.
16. J. Portilla, "Análisis y diseño de un sistema de reconocimiento facial aplicando Machine learning para detectar e identificar intrusos". Universidad de Guayaquil, Guayaquil, 2018.
17. A. R. Syafeeza, M. K. M. F. Alif, N. Athirah, A. S. Jaafa y A. H. Norihan, "Lot based facial recognition door access control home security system using raspberry pi". International journal of power electronics and creative system (IJPEDS), 2020.
18. L. Zapata y J. David, "Prototipo de sistema domótico centralizado a través de una página web con reconocimiento facial" de Ingeniería en Sistemas de Computación e Informática, Quito: Universidad de las Américas, 2020, 2020.
19. L. Cerda Muñoz, "Diseño e implementación de un sistema seguro de control domótico". Universidad Autónoma de Madrid., 2018.
20. T. A. W. David R. Anderson, "Métodos cuantitativos para los negocios". CENGAGE LEARNING, 2011.