REVISTA
ODIGOS

Período junio - septiembre 2020
Vol. 1, Núm. 2
rodigos@uisarel.edu.ec
https://doi.org/10.35290/ro.v1n2.2020.326 • e-ISSN: 2697-3405

# Face recognition for automatic vehicle ignition based on Raspberry Pi

*Fecha de recepción: 10/04/2020 • Fecha de aceptación: 05/05/2020 • Fecha de publicación:10/06/2020*

**Alex V. Nuñez**[1]
*Essex County College, United States*
anunez16@email.essex.edu
*https://orcid.org/0000-0001-7274-5878*

**Liliana N. Nuñez**[2]
Rutgers University, United States
*lnn21@scarletmail.rutgers.edu*
*https://orcid.org/0000-0002-7519-8360*

## ABSTRACT

In this project a facial recognition application for automatic vehicle ignition is developed. This application is built using a Raspberry Pi as the hardware platform and the OpenCV library for computer vision as the software component. In this research the different methods for automobile security are analyzed, as well as, the different methods used to perform face recognition.  The main goal of this application is to enhance the security system of the vehicle, allowing to ignite the vehicle only by register users. To achieve this goal three main processes are carried out, face detection, data gathering, and training the system to grant access through face recognition.

**KEYWORDS:** facial recognition, Raspberry Pi, OpenCV, vehicle, automatic ignition

## RESUMEN

En el presente trabajo se desarrolla una aplicación de reconocimiento facial para el encendido automatico de vehículos. Esta aplicación se construye utilizando como plataforma de hardware Raspberry

Pi y OpenCV como almanenamiento de imágenes. En esta investigación se analizan los diferentes métodos de seguridad para los automóviles, así como los diferentes procedimientos utilizados para realizar el reconocimiento facial. El objetivo principal de esta aplicación es mejorar el sistema de seguridad para los automóviles, permitiendo un encendido directo solo con el usuario registrado. Para lograr esto se llevaron a cabo tres procesos principales: detección de rostros, recopilación de datos y capacitación del sistema para conceder el acceso.

**PALABRAS CLAVE:** reconocimiento facial, Raspberry Pi, OpenCV, vehículos, encendido automatico

# Introduction

One of the most important fields in the automotive industry is security. Since the invention of the automobile, many systems have been developed around this aspect. Security is one of the most critical fields when developing a car. To satisfy the required security standards various security systems have been developed over the years, one of the most widely and longest used systems is the key ignition switch system, although this system has security that blocks access to the car's ignition switch if the key used is not the correct one, it has been shown that it is not a foolproof system (Lemelson & Hoffman, 2004).

In response to this problem, a wide variety of supplementary devices that allow increasing the levels of security already established are used. Such as steering-wheel-securing "clubs" and alarms activated by moving the locked car. However, not all of these security measures have been sufficient to end the theft of cars. Since the delinquents are finding new ways to alter and ignore these security systems. As a result, auto theft remains a multi-billion-dollar "business" despite the best efforts of the auto industry and the police to stop them (Lemelson & Hoffman, 2004).

Another security system that has been developed is the passive starting system (PEPS). In this system the doors of the vehicle are automatically unlocked when an authorized key fob is brought close to the vehicle and its starting system is carried out through a button. Nevertheless, it has been proven that even this technology can be violated through transceivers causing a retransmission attack (Oman & Haves, 2015), that is why the creation of a new and better security system is necessary.

The invention of new and better technologies, as well as the great advance that the field of electronic engineering and software development has had, have allowed access to development alternatives such as artificial intelligence and computer vision. With the help of these new technologies and seeking to provide a solution to the security problems experienced, a new security system for vehicle ignition is proposed in this article.

## 1. Literature review

In this section, similar researches to this proposal are analyzed. These researches focus on the study of the facial recognition technique using computer vision tools. Similarly, they use different hardware components such as Raspberry Pi or personal computers and artificial vision libraries such as Dlib or OpenCV. The most relevant research will be analyzed below.

The research developed by Boyko to study the two most widely used computer vision libraries: OpenCV and Dlib; These libraries define the general concepts as well as the scientific principles behind facial recognition theory. In this research the characteristics of these two libraries are explored, and the pros and cons of each one are analyzed. Additionally, we analyze application examples based on histogram-oriented gradient techniques for face search, face landmark for facial recognition and deep convolutional neural network to compare known faces. As a result of the study, the OpenCV library presents better performance and productivity than the Dlib library, making it ideal for facial recognition (Boyko, Basystiuk, & Shakhovska, 2018).

For his part, Gulzar proposes the development of a car security system based on facial recognition. The main objective of this project is to develop a low-cost system based on open source and adaptable software platforms for all types of vehicles. It is also intended to study the limitations of facial recognition techniques, providing solutions that allow the system to be efficient in terms of speed of execution and response time (Gulzar, Jun, & Tariq, 2017).

While Pawar in their research work propose to create an embedded security system for vehicle security and surveillance. His article proposes the development of an anti-theft system and an embedded surveillance system. The same that uses biometric authentication to access the vehicle. This system uses a camera to perform facial recognition of the person requiring access. Which in case of denying access captures photos of the person who tried to access the vehicle and sends them to the owner to notify them of possible theft. This system has been designed and developed using the Raspberry pi card and a high-resolution camera, as well as open-source software tools (Pawar & Rizvi, 2018).

While Pawar in their research work propose to create an embedded security system for vehicle security and surveillance. His article proposes the development of an anti-theft system and an embedded surveillance system. The same that uses biometric authentication to access the vehicle. This system uses a camera to perform facial recognition of the person requiring access. Which in case of denying access captures photos of the person who tried to access the vehicle and sends them to the owner to notify them of possible theft. This system has been designed and developed using the Raspberry pi card and a high-resolution camera, as well as open-source software tools (Pawar & Rizvi, 2018).

In summary, the analyzed works present relevant ideas for the achievement of the proposed project. Emphasizing the use of the appropriate software to perform facial recognition, conjunction with the Raspberry pi card used to develop the present project.

## Methodology

The limited development of proposals such as the one presented, as well as the significant advances that have taken place in the field of technology, and especially in the area of software development, has made possible the realization of this research. In this work, the OpenCV library for computer vision and a Raspberry pi 4 model B has been used to achieve our objective.

The fundamental principle of facial recognition is based on identifying and determining the characteristics of a face through statistical or intellectual methods. These methods allow us to build face models and compare them to the coincidence level of the detection region of a face. Subsequently the possible region of the face is obtained (Fan, Zhang, Wang, & Lu, 2012).

The AdaBoost face detection algorithm is a method used to obtain face regions, which later will allow us to perform face recognition, it includes Harr-like selection features and calculates the features of a typical face through the image integral. Extended Harr-like features can also be used to improve accuracy when detecting a face, which are divided into edge features, linear features, and center-surround features (Fan et al., 2012). A graphic representation of these characteristics can be seen in *Figure 1*.

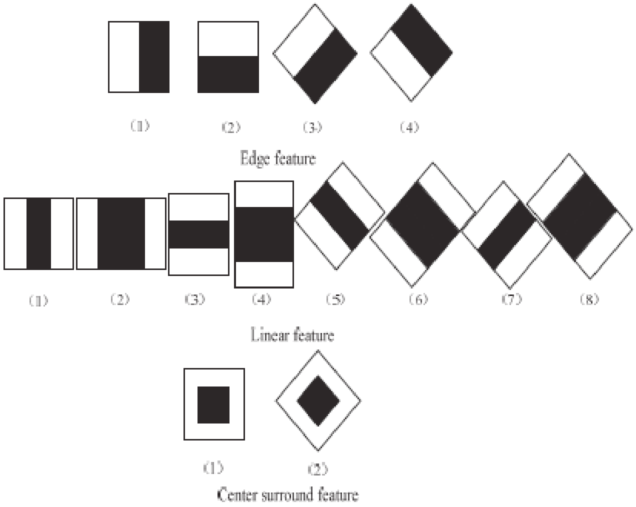


**Figure 1.** *Extended Harr-like features*

**Source:** (Fan et al., 2012)

Another method widely used to perform face recognition is the integral image method. This method is used to quickly calculate the value of the characteristic, which is defined by the difference between the sum of the white pixels and the black pixels. In *Figure 2* it shows that the characteristic value composed by II and IV is the variation between the sum of IV pixels and the sum of II pixels. The former is the difference between the sum of integral image value of A and D and the sum of integral image value of B and C. The later is the sum of integral image value of B and integral image value of A (Fan et al., 2012).

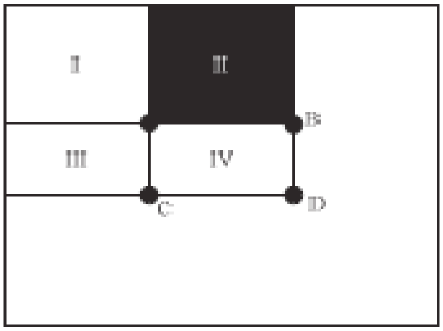


**Figure 2.** *Four arrays figure*

**Source:** (Fan et al., 2012)

### 1. OpenCV

OpenCV (Open source computer vision) is a library of programming functions mainly aimed at real-time computer vision, originally developed by Intel, it was later supported by Willow Garage and Itseez. The library is cross-platform and free for use under the open-source BSD license. OpenCV supports some models from deep learning frameworks like TensorFlow, Torch, PyTorch, and Caffe according to a defined list of supported layers. It promotes OpenVisionCapsules which is a portable format, compatible with all other formats (Rathod & Agrawal, 2018). The logo for OpenCV can be seen in *Figure 3*.

**Figure 3.** *OpenCv Logo*

**Source:** (OpenCv team, 2020)

### 2. Raspberry Pi

The Raspberry Pi (RPi) 4 Model B is a credit card-sized, single-board computer. It possesses a 1.5GHz 64-bit quad-core ARM Cortex-A72 CPU processor and has 40 General Purpose Input/output (GPIO) pins. It can support up to 4GB RAM (Jabeen, Ramamurthy, & Latha, 2017).

The device is powered by a 5V micro USB with an ampere rating of 2A. a monitor with a micro HDMI port can be used as a display and a USB-based keyboard and mouse can be used to control the RPi. Since the monitor has a touch interface the USB cable for touch has to be connected to the USB port of RPi, this new model has 2 USB 3.0 ports and 2 USB 2.0 ports. It also has 2.4 GHz and 5.0 GHz IEEE 802.11ac wireless interface and an Ethernet Gigabit port (Jabeen et al., 2017). We can appreciate it in Figure 4.

The operative system for the RPi needs to be loaded in a microSD, although it can support many OS the most used is Raspbian which is a Linux based Operative System. Before running the RPi the SD card should be inserted into the MicroSD card slot, also, the main language used for programming the RPi is Python which acts as the control system of the model (Jabeen et al., 2017).

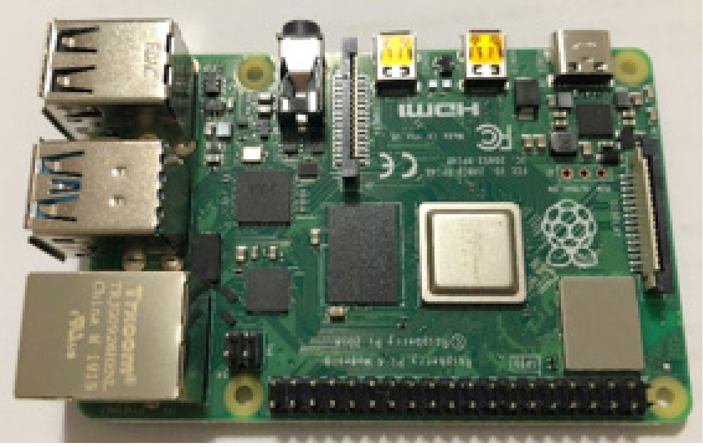


**Figure 4.** *Raspberry Pi 4 model B*
**Source:** Ñuñez, Alex V & Nuñez Liliana N, 2020

The project consists of three essential processes, first is in charge of data acquisition, this process is done through the web camera, which is in charge of acquiring the photographs of the users who will be granted access to the vehicle. Once you have acquired the photographs of the users is necessary to store them to create a database of potential users. This is done because it is necessary to compare the faces of the people who want to grant access to the photographs of the registered users stored in the database.

Subsequently, the system must process all this information so that it is capable of performing facial recognition on a specific person. Therefore, a code has been designed that processes the photographs of the different users stored in the system database and extracts information on the unique characteristics of each face, regenerating a binary file with a yml extension.

Finally, the third process is responsible for performing facial recognition of a user, making use of the information collected before. This process begins when a new user wants to access the system. The system acquires the facial information of this potential new user. This information is compared with the database of registered users and in case the facial information of this user coincides with the information of registered users, access is granted, otherwise, access is denied. If access is granted the electronic card sends a signal to the vehicle's computer which is the process and grants access to the user.

The architecture of the project developed is shown in *Figure 5*. It will allow us to understand the basic operation of the proposed system and how it exchanges information between its different components.

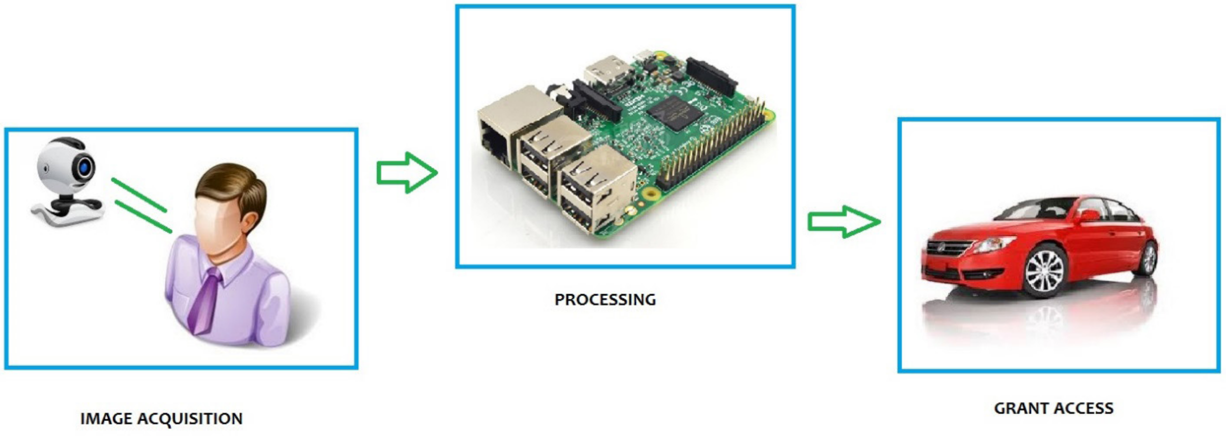


**Figure 5***. Project Architecture*
**Source:** Ñuñez, Alex V & Nuñez Liliana N, 2020

To carry out the facial recognition system, three primary code structures were developed. The first structure is responsible for the acquisition of the users' photographs and the creation of the database. The second structure in charge of generating the yml file with the information on the characteristics of the faces of the stored users and the last one is in charge of carrying out the facial recognition process of the users and guaranteeing access. In this section, we will proceed to describe in detail each of these processes.

### 3. Face detection and data gathering

In this process the main objective is to generate a database with the images of registered users. Therefore, the first task to be performed is to determine the correct software tool that will allow us to detect a face, from the analysis carried out, it was determined that the best tool for face recognition is the Harr Cascade classifier function (Rovai, 2018).

Object detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their 2001 article "Rapid *Object Detection using a Boosted Cascade of Simple Features*". It is a machine learning-based approach where cascading is formed from many positive and negative images. Then it is used to detect objects in other images. In *Figure 6* presents the outline of this process (Rovai, 2018).

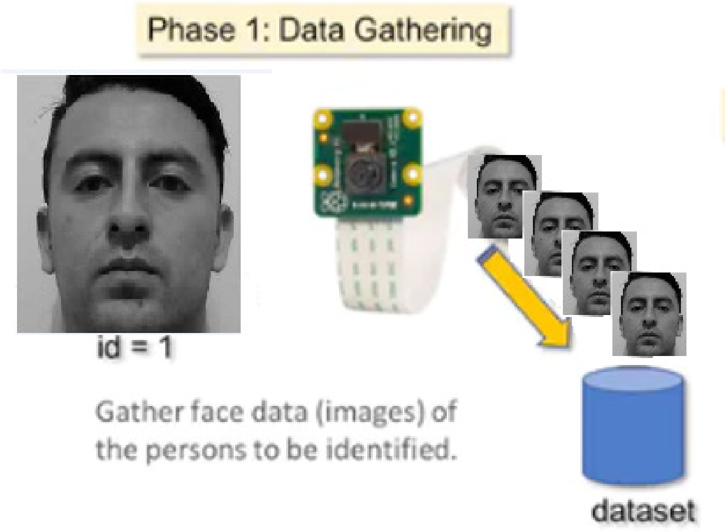


**Figure 6.** *Face Detection and Data Gathering*
**Source:** Ñuñez, Alex V & Nuñez Liliana N, 2020

The face detection option of this tool has been used, to use this algorithm it is necessary to have a considerable number of positive images, which must contain the faces of the users. Similarly, it is also necessary to have a large number of negative images without the faces of the users, to train the classifier. Subsequently, it is necessary to extract the characteristics of the faces from the images provided. In this case, the OpenCV library has been chosen for artificial vision, since it includes both the detector and the trainer necessary to carry out the aforementioned process (Siswanto, Nugroho, & Galinium, 2014).

When a face is detected by the webcam the code fires an event that captures a certain number of user photographs to store them in the database. When the photographs are acquired, they are assigned an id and converted to grayscale to be able to work with them.

### 4. Train the recognizer

This process will be in charge of processing the information of the images contained in the database in order to train the recognizer of the OpenCV library. This process is done automatically by a code function contained within OpenCV. As a result of this process, an yml extension file will be obtained. Which contains the information used by the recognizer and that will allow us to make a comparison between the different faces of the users (Raja, 2020). The diagram of this process is shown in *Figure 7*.

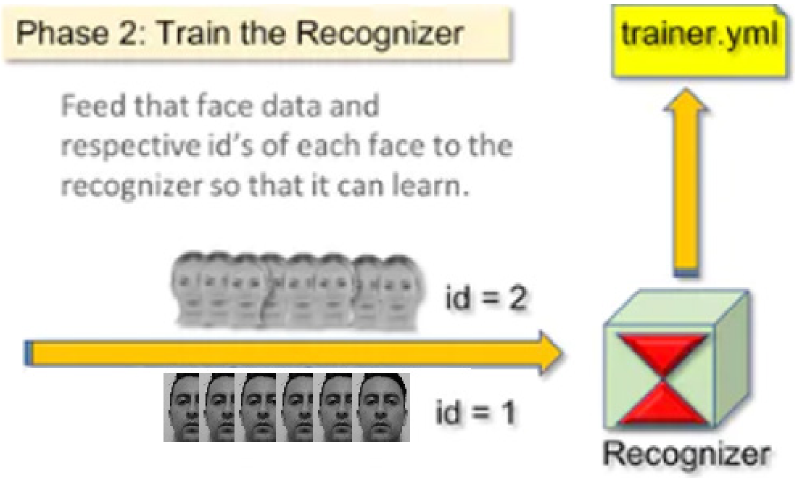

Phase 2: Train the Recognizer
trainer.yml
Feed that face data and respective id's of each face to the recognizer so that it can learn.
id = 2
id = 1
Recognizer

**Figure 7.** *Training process*

**Source:** Ñuñez, Alex V & Nuñez Liliana N, 2020

## 5. Face recognition

The facial recognition process is the last phase of the project. The capture of a new face is made through the webcam and a comparison is made with the images previously stored and trained. The OpenCV recognizer will make a prediction and return the index of the image related to the processed face and the percentage of coincidence between these two images(Raja, 2020). The diagram of this process is shown below in *Figure 8*.

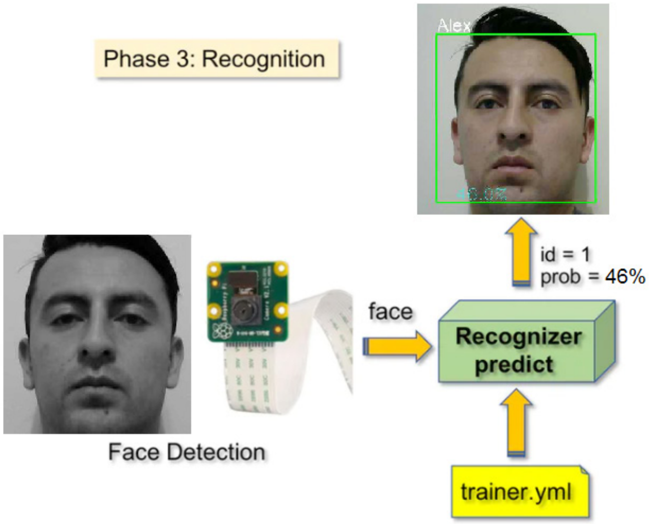
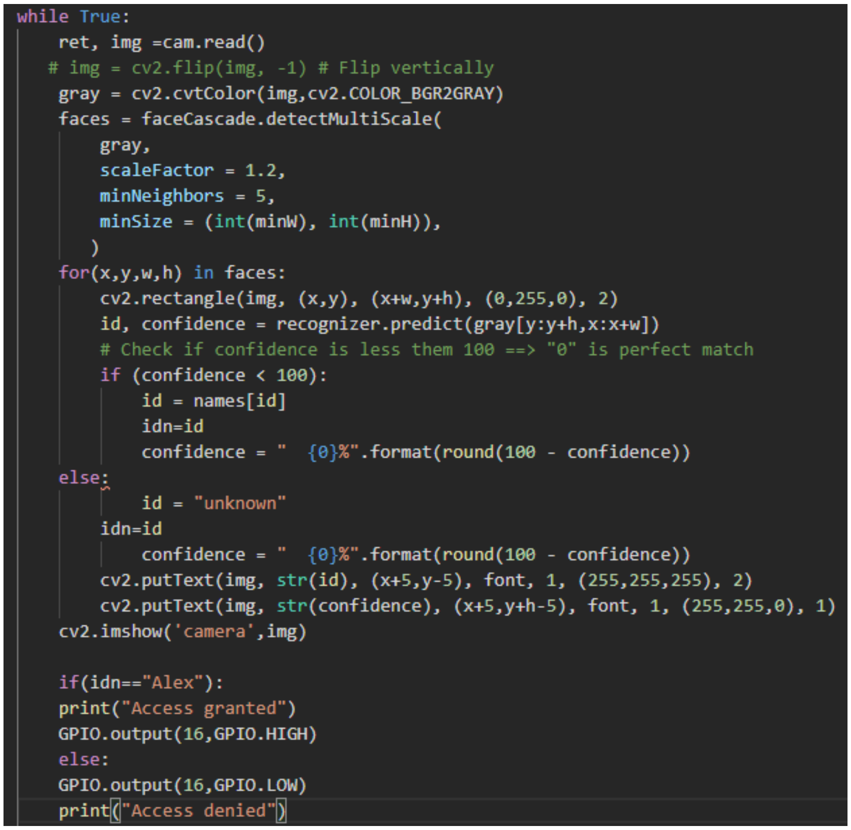


**Figure 8.** *Face Recognition process*

**Source:** Ñuñez, Alex V & Nuñez Liliana N, 2020

The first step in this process is to detect the face of the potential user who is seeking access. For which the haarCascade classifier function will be used. Subsequently, it is necessary to determine the percentage of coincidence between these two images. For which the recognizer.predict() function will be used. This function will take as a parameter a portion of the captured face to analyze and determine the likely user associated with that face, indicating the name of the user and the id that has been assigned to it, as well as the degree of confidence that exists between the acquired face and the face stored in the database (Raja, 2020). The code belonging to this third process is shown in *Figure 9*.



```
while True:
    ret, img =cam.read()
   # img = cv2.flip(img, -1) # Flip vertically
    gray = cv2.cvtColor(img,cv2.COLOR_BGR2GRAY)
    faces = faceCascade.detectMultiScale(
        gray,
        scaleFactor = 1.2,
        minNeighbors = 5,
        minSize = (int(minW), int(minH)),
       )
    for(x,y,w,h) in faces:
        cv2.rectangle(img, (x,y), (x+w,y+h), (0,255,0), 2)
        id, confidence = recognizer.predict(gray[y:y+h,x:x+w])
        # Check if confidence is less them 100 ==> "0" is perfect match
        if (confidence < 100):
            id = names[id]
            idn=id
            confidence = "  {0}%".format(round(100 - confidence))
    else:
            id = "unknown"
        idn=id
            confidence = "  {0}%".format(round(100 - confidence))
        cv2.putText(img, str(id), (x+5,y-5), font, 1, (255,255,255), 2)
        cv2.putText(img, str(confidence), (x+5,y+h-5), font, 1, (255,255,0), 1)
    cv2.imshow('camera',img)

    if(idn=="Alex"):
    print("Access granted")
    GPIO.output(16,GPIO.HIGH)
    else:
    GPIO.output(16,GPIO.LOW)
    print("Access denied")
```

**Figure 9.** Face recognition code

**Source:** Ñuñez, Alex V & Nuñez Liliana N, 2020

Once a registered user has been detected, an electronic signal is generated and sent to the car's computer using the GPIO ports of the Raspberry pi card. For which it is determined if there were detected users, in this case, we proceed to grant a high logical state to one of the GPIO ports, otherwise, a low logical state is established.

## Results

As the first test of the system, the face of a not registered user in the database will be processed and the behavior of the system will be observed *Figure 10*.

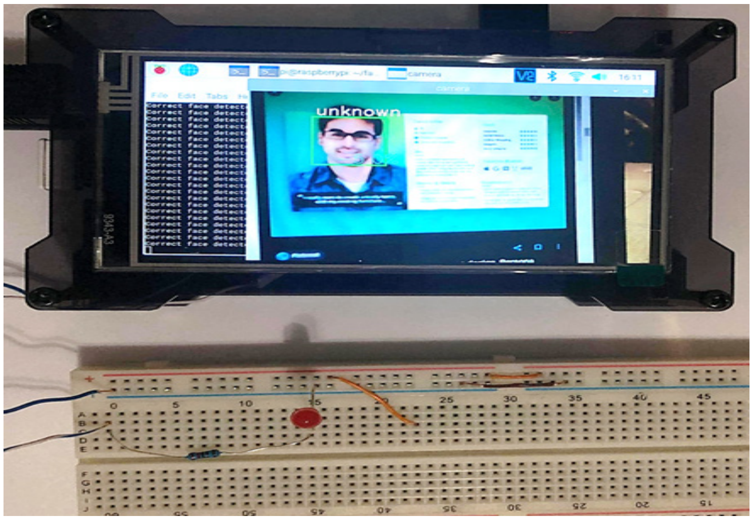


**Figure 10**. Not registered user

**Source:** Ñuñez, Alex V & Nuñez Liliana N, 2020

As it could be seen, the user who tried to access the system was not registered in the database, so the system could not recognize it, showing it as an unknown user and throwing a low-status logical signal. Later we will proceed to process the face of a registered user, this process is shown in *Figure 11*.

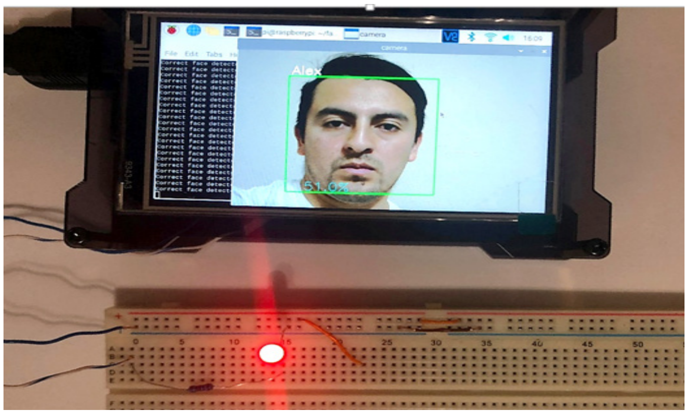


**Figure 11.** *Processing registered users*

**Source:** Ñuñez, Alex V & Nuñez Liliana N, 2020

As evidenced in the image the system is working expectedly. Since it was able to recognize the registered user and show his name and the percentage of coincidence determined by the OpenCv recognition. Since a registered user has been determined and recognized, a high-level logical signal is generated with which it is determined that the system is working correctly, recognizing users and guaranteeing access.

# Conclusions

In conclusion, the methods specifically focused on automotive safety are very fragile and can be altered very easily. Therefore, the need for much more robust security systems is necessary, and the proposed proposal is a viable option that involves the solution to all these problems.

The development of systems such as the one proposed implies an improvement in security levels, implementing new features that improve the user experience. Since it can allow working together with departments such as the police, having the ability to generate reports of possible criminals by storing their photographs when they try to violate the security of the vehicle.

Finally, the increase in security would lead to a reduction in vehicle insurance costs, this as a results of having much more reliable security systems, the crime rate regarding car theft would tend to decrease.

# References

Boyko, N., Basystiuk, O., & Shakhovska, N. (2018). Performance Evaluation and Comparison of Software for Face Recognition, Based on Dlib and Opencv Library. *Proceedings of the 2018 IEEE 2nd International Conference on Data Stream Mining and Processing, DSMP 2018*, 478–482. https://doi.org/10.1109/DSMP.2018.8478556

Fan, X., Zhang, F., Wang, H., & Lu, X. (2012). The system of face detection based on OpenCV. *Proceedings of the 2012 24th Chinese Control and Decision Conference, CCDC 2012*, 648–651. https://doi.org/10.1109/CCDC.2012.6242980

Gulzar, K., Jun, S., & Tariq, O. (2017). A cost effective method for automobile security based on detection and recognition of human face. *2017 2nd International Conference on Image, Vision and Computing, ICIVC 2017*, 259–263. https://doi.org/10.1109/ICIVC.2017.7984557

Jabeen, F., Ramamurthy, B., & Latha, N. A. (2017). Development and implementation using Arduino and Raspberry Pi based Ignition control system, *10*(7), 1989–2004. https://www.ripublication.com/acst17/acstv10n7_03.pdf

Lemelson, J., & Hoffman, L. (2004). Vehicle security systems and metods employing facial recognition using a reflected image. United States. https://doi.org/10.3

Oman, T. P., & Haves, K. J. (2015, May 1). Relay attack prevention for passive entry passive start (PEPS) vehicle security systems.

OpenCv team. (2020). OpenCV. Retrieved May 17, 2020, from https://opencv.org/

Pawar, M. R., & Rizvi, I. (2018). IoT Based Embedded System for Vehicle Security and Driver Surveillance. *Proceedings of the International Conference on Inventive Communication and Computational Technologies, ICICCT 2018*, (Icicct), 466–470. https://doi.org/10.1109/ICICCT.2018.8472984

Raja, R. (2020). Face recognition using OpenCV and Python: A beginner's guide. Retrieved May 26, 2020, from https://www.superdatascience.com/blogs/opencv-face-recognition

Rathod, V., & Agrawal, R. (2018). Survey : Automatic Understanding By Vehicle For Driver Distraction Problem. *International Research Journal of Engineering and Technology (IRJET)*, 54–59.

Rovai, M. J. (2018). Real-Time Face Recognition: An End-to-End Project. Retrieved April 23, 2020, from https://www.hackster.io/mjrobot/real-time-face-recognition-an-end-to-end-project-a10826

Siswanto, A. R. S., Nugroho, A. S., & Galinium, M. (2014). Implementation of face recognition algorithm for biometrics based time attendance system. *Proceedings - 2014 International Conference on ICT for Smart So-*

*ciety: “Smart System Platform Development for City and Society, GoeSmart 2014”, ICISS 2014*, 149–154. https://doi.org/10.1109/ICTSS.2014.7013165