Embedded system design to control the entry and exit of vehicles online, at the main access of ESPOCH

Diseño de sistemas embebidos para controlar la entrada y salida de vehículos en línea, en el acceso principal de la ESPOCH.

Javier J. Gavilanes[1,*], Jairo R. Jácome1[1,†], and Alexandra O. Pazmiño[1,†]
[1]Escuela Superior Politécnica del Chimborazo, Ecuador.
{javier.gavilanes, jjacome, apazmino_a}@espoch.edu.ec

Received: August 15, 2017 — **Accepted:** September 15, 2017

How to cite: Gavilanes, J. J., Jácome, J. R., & Pazmiño, A. O. (2018). Embedded system design to control the entry and exit of vehicles online, at the main access of ESPOCH. *Journal of Science and Research: Revista Ciencia e Investigación, 3(CITT2017), 113-120.* https://doi.org/10.26910/issn.2528-8083vol3issCITT2017.2018pp113-120

Abstract—**In this research a embedded real-time system was developed by using Raspberry Pi3 (a reduced board computer), which is an equipment with a camera placed in strategic points of the mechanic arms at the main entrance and exit of Escuela Superior Politécnica de Chimborazo, this equipment captures images of vehicles that enter and exit the campus and the information is extracted through the implementation of a segmentation algorithm written in Python programming language and the collaboration of artificial vision bookstores offered by OpenCV, processing techniques were applied to extract the vehicle plate from the location scenery. Then, an Optical Character Recognition (OCR) algorithm also known as K-Nearest Neighbours (KNN) was applied, which after a training phase is able to identify letters and numbers on the automobile plates, the information is stored in the entrance database and it is deleted when the automobile exits the campus.**

Keywords—**Automobile, Database, Internet of things, MySQL, Opencv, Web server.**

Resumen—**En esta investigación se desarrolló un sistema en tiempo real encajado usando frambuesa Pi3 (una computadora de tablero reducido), que es un equipo con una cámara colocada en puntos estratégicos de los brazos mecánicos en la entrada principal y salida de la Escuela Superior Politécnica de Chimborazo, este equipo captura imágenes de los vehículos que entran y salen del campus y la información se extrae a través de la implementación de un algoritmo de segmentación escrito en lenguaje de programación Python y la colaboración de visión artificial las librerías ofrecen OpenCV, se aplicaron técnicas de procesamiento para extraer la placa del vehículo de la escenografía del lugar. Entonces, un algoritmo de reconocimiento óptico de caracteres (OCR) también conocido como K-Nearest vecinos (KNN) se aplicó, que después de una fase de formación es capaz de identificar Letras y números en las placas de automóvil, la información se almacena en la base de datos de entrada y es se eliminan cuando el automóvil sale del campus.**

Palabras Clave—**Automóvil, La base de datos, Internet de las cosas, MySQL, Opencv, servidor web.**

INTRODUCTION

At the country's automotive fleet rate shows levels of imminent annual growth. In the year 2013 alone, according to the Instituto nacional de estadística y censo (INEC), there were 1 717 886, of which 511511 were registered in the province of Chimborazo, representing 2.53 % of the total of vehicles registered (INEC, 2012).

From then on, several social problems have arisen in consequence of the large amount of vehicles circulating in the cities such as: environmental, visual and auditory pollution, traffic and insecurity.

According to INTERPOL data during 2014, more than seven million cars were stolen, and most of them were sold outside the countries they came from, unfortunately Ecuador is part of these statistics. In the country, the number of cars stolen in 2015 were around 7478 of which only 2695 were retrieved according to reports of Integral Security Statistics which generated an economic loss of approximately USD 22 million according to local investigators (Interpol, 2014).

The system to be built must register the vehicles' plates that enter the institution and then store them in a database, for which it is taken as a baseline some prototypes previously developed in other institutions, such as the case of UEPSE, where a system that register vehicles through artificial vision was Implemented (Yagual et al., 2013), a similar project in the UPS of Cuenca's city (Álvarez Durán, 2014) and a work of artificial vision implemented in MATLAB that recognizes vehicular registrations at the UPS of Guayaquil (Barcia, 2014).

In addition, previous research was found at the institution where the present project will be implemented. A system programmed in LabVIEW which detects the characteristics of the plates that was implemented in the Laboratory of Industrial Automation that is part of the Systems Engineering School (Barragán Del Pozo, 2012), a payment and parking access system using pattern recognition (Contreras Hurtado et al., 2015) and a comparative study between Sobel and Canny

[*]Ingeniero en Electrónica, Control y Redes Industriales, Máster en Automática y Robótica

[†]Ingeniero en Electrónica y Computación, Magíster en Sistemas de Telecomunicaciones

[†]Ingeniera en Electrónica y Computación, Magíster en Informática Empresarial

algorithms to determine speed and accuracy of the response (Paredes Muñoz and Guerrero García, 2012).

A disadvantage of previous systems is the need for a computer to perform the processing, which limits the practical application. One solution to avoid this is the use of embedded systems.

In an embedded system, we distinguish three characteristics that define it and distinguish it from other computational systems.

- In this system, specific tasks are executed repetitively, unlike an ordinary computer system that executes
- a certain number of programs.
- Embedded systems are generally inexpensive, small in size, perform well for real-time data processing, and also consume a minimum of energy to extend battery life or prevent the need for additional cooling (Pérez A., 2009).
- Ability to react to environmental changes, without interrupting its operation or delaying the functions granted.

An embedded system is still a computer, so it is useful to apply artificial vision algorithms.

Fu, mentions that: Artificial vision can be defined as the processes of obtaining, characterizing and interpreting information from images taken from a three-dimensional world (Fu et al., 1989, p306).

As a general concept, Pajares and De la Cruz define it as ”the machine's ability to see the world around it, more precisely to deduce the structure and properties of the three-dimensional world from one or more two- dimensional images (Pajares Martinsanz et al., 2002).

The detection software is written in Python because it is the default programming language supported by the embedded system. This code is made up of two algorithms, the first one focuses on the segmentation of images, covering the image manipulation process to delimit the area of the vehicle plate, and the second algorithm is designed for the detection of alphanumeric characters shown in the image taken of the license plate.

The present project is characterized by working with vehicular plates which is the only distinctive feature that distinguishes and differentiates each vehicle in accordance with the Agencia Nacional de Tránsito del Ecuador (ANT) (Agencia Nacional de Tránsito, 2015). This entity provides the color and characters that will be present on the plate, according with the use of the vehicle and its place of registration respectively.

As of 2012, characteristics of vehicles' plates (Figure 1) were modified, among them are: its dimension of 154 mm high and 404 mm wide constructed of reflective material and depending on the type of vehicle the top edge color with white background varies (Foros Ecuador, 2017).

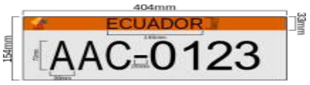


Figure 1. Vehicular plate.
Source: Agencia Nacional de Tránsito del Ecuador.

The alphanumeric characters are formed of three letters and four digits from 0000 to 9999, with the word ECUADOR at the top. The first letter corresponds to the province where the automobile was registered for the first time, the second one identifies the type of registration and the third one is relative.

Vehicle Plate Detection

The embedded system (Pérez A., 2009) will be installed at the main access of Escuela Superior Politécnica del Chimborazo, being a suitable place to obtain the information of interest of the vehicles that go in.

When a vehicle goes in the campus, a camera arranged in a strategic location will take an image of the scene, where a clear image can be obtained, trying to avoid factors that may interfere with the acquisition process such as the distance between the camera and the vehicle, the angle of the shot, among others. It should be noted that at the time the vehicle leaves the campus the same action will be performed.

Once the digitized scene is obtained, several processing techniques are applied to it such as: segmentation, filtering, noise elimination and so on, in order to obtain the area of interest that will be the vehicle's plate, then a character recognition algorithm will extract the information of the plate which is the unique distinctive that differentiates each vehicle.

After the recognition of the vehicle plate, this information will be stored in a database and available online, ready to be accessed at anytime from anywhere simply by having a device that allows access to the web such as: cell phone, personal computer, tablet, and so on.

All this process will be executed by an embedded system in order to adapt all concepts of artificial vision, databases and internet networks in a single robust, reliable and efficient system capable of being implemented at any site that requires information about the permanence of vehicles.

Figure 2 shows a flowchart by software to segment a vehicle plate from the image captured.
Filtered → Filtration
Dimensions Ratio → ???

Image Capture

The resolution of the images to be captured is 3280 x 2464 with the maximum aspect ratio allowed by the camera of 4:3, to avoid losing details. Upon culminating with the process of obtaining images, it is necessary to determine where the vehicle plate is located through several procedures.

Reading images

The stored images are read with the command imread, for later processing. The variable that will save the image is nbr_img, so the command would look like this: nbr_img = cv2.imread ('img_fnt.extension').

Cutout

After making several captures of the scenarios to be processed, it is determined that the area of interest is located below the midpoint of the image, the vehicle plate is located in the middle of the car's body. After taking this into consideration, the image is trimmed by removing the lateral and upper parts of it as shown in Figure 3.

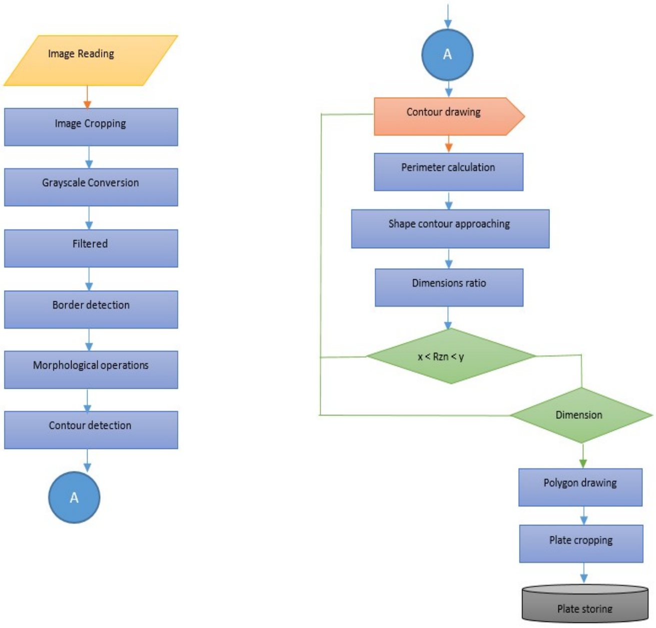


Figure 2. Steps to segment a vehicle plate from the image captured. **Source:** Prepared by the authors.

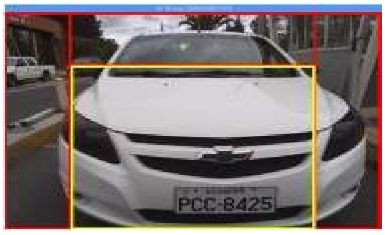


Figure 3. Cropping the image. **Source:** Prepared by the authors.

Grayscale Conversion

The image to be processed is converted to grayscale to facilitate its manipulation and increase the speed of processing due to the reduction of information to be processed. In the program the code cv2.cvtColor (’img_fnt’, cv2.COLOR_BGR2GRAY) is used to convert it to grayscale. The image in Figure 4 shows the original image and the one in grayscale.

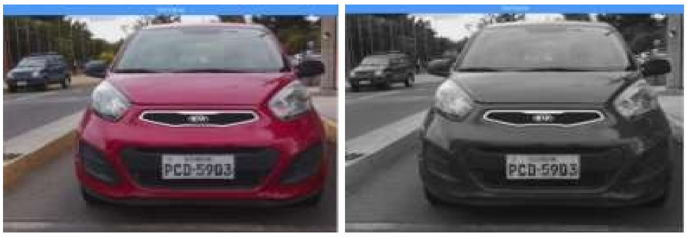


Figure 4. Grayscale Conversion. **Source:** Prepared by the authors.

Filtering the image

The grayscale image is subjected to a filtering stage in order to reduce the noise presented by different factors when capturing the image. The type of filter applied is Gauss which works just like a media filter, with the difference that a Gaussian distribution of each one of the close pixels is calculated instead of calculating the average, using Ec.1 that is find below:

$$T(x,y) = e^{\frac{x^2 - y^2}{2\pi\delta^2}} \tag{1}$$

The size of the mask determines the level of smoothing applied in the image, this time a 5x5 has been applied which does not soften and eliminate the noise of the picture obtained. The difference between smoothing can be seen in Figure 5 which contains images filtered by Gauss with two different mask sizes.

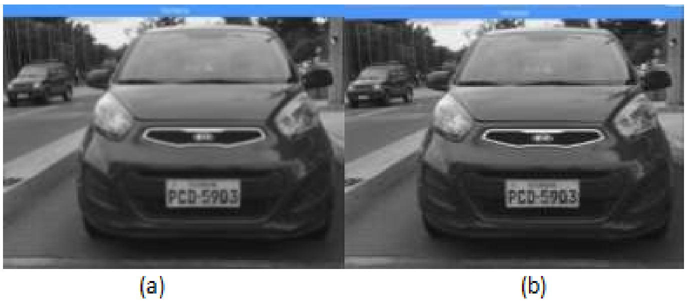


Figure 5. Gauss filter. (A) Mask 3x3. (B) 9x9 mask. **Source:** Prepared by the authors.

The filter application is performed by the command cv2.GaussianBlur (img_fnt, (m x n), 0). Three arguments correspond to the source image, the size of the transformation matrix, and the standard deviation generally with a value of zero.

Edge detection

Objects in the images are differentiated by the application of edge detection, in this case the Canny algorithm has been used. It is one of the best ones available for this type of processing. The algorithm is based on the contour definition, that is described as the pixel zones in which there is a sudden change of gray level. In Figure 6 we see the result of applying the cv2.Canny command to the image of a vehicle.

Closing the image

This process is known as a morphological operation used to eliminate small white spots located outside the area of interest or black spots present at the edges. This operation is made up of two sub-operations which correspond to erode and dilate the image. The command used is cv2.morphologyEx (img_fnt, cv2.MORPH_CLOSE, kernel); the first argument refers to the image to be processed, the second one selects the morphological operation and the third one corresponds to the matrix's dimension to perform erosion and image dilation. For Figure 7 a matrix of size 6x6 has been considered and the image can be observed before and after applying the command.

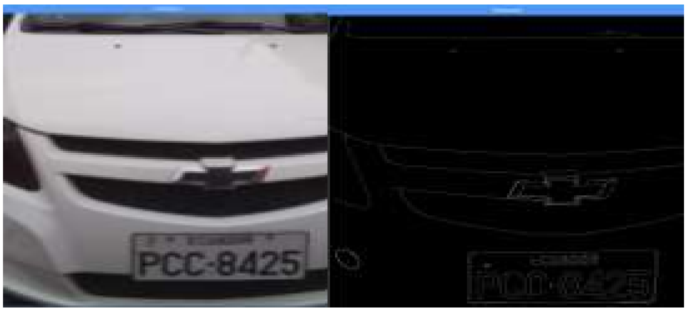


Figure 6. Canny Edge Detection.
Source: Prepared by the authors.

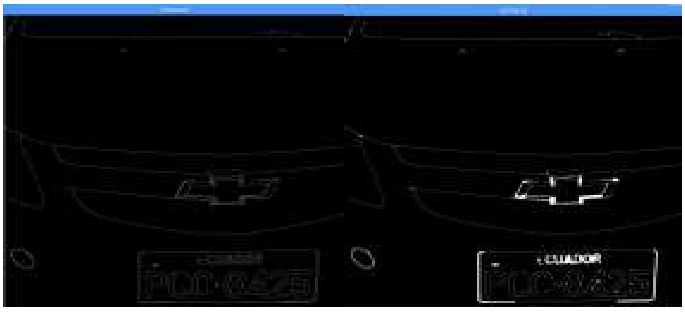


Figure 7. Canny Edge Detection.
Source: Prepared by the authors.

Contour Detection

Contour detection analyzes the points where there is a variation in gray levels which determines the existence of the edge of an object. The command used in this process is cv2.findContours (), which requires three arguments, the first corresponds to a variable that refers to the image on which to operate, the second defines the mode of relationship between them and the third is the argument that allows to specify the method of compression that will be used to generate a contour.

A precision variable must be defined to find contours, under some tests performed with different images, a value of 0.07 has been chosen, which allows to predict one form inside another with the least number of possible vertices. Figure 8 shows the results of the detection algorithm in two different vehicles.

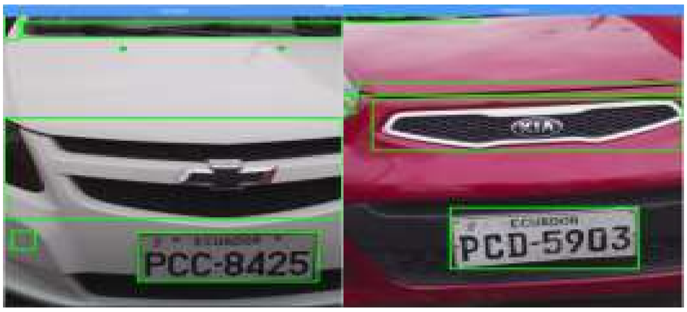


Figure 8. Possible vehicle plates.
Source: Prepared by the authors.

The detection results in several edges of which only one is the plate, the others are called false positives, which must be eliminated. Purification of areas is achieved by applying two filters. The first by dimension of the plate and the second by comparison of dimensions between the resulting edges.

The first filter corresponds to the relationship between the length and width dimensions of the plate, which has a defined size of 404 mm and 154 mm, respectively, by calculating the relation with Eq. 2.

$$R = \frac{lrg}{anc} \tag{2}$$

Where:
lrg is the length of the plate.
Anc is the width of the plate.
R is the ratio obtained from the plate.

Figure 9 shows the discrimination of edges by using the ratio of their lengths.

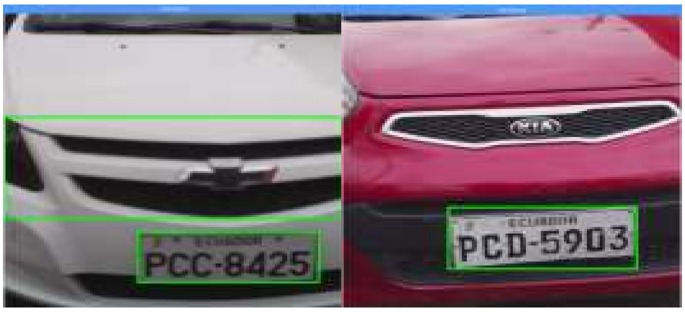


Figure 9. Filtering by size ratio.
Source: Prepared by the authors.

However, in one of the vehicles there is still a false positive, which is why the second type of filter is based on comparing its dimensions of length and width, in this way the pixels that will occupy the vehicular plate inside the image can be predicted, obtaining results like those of Figure 10.

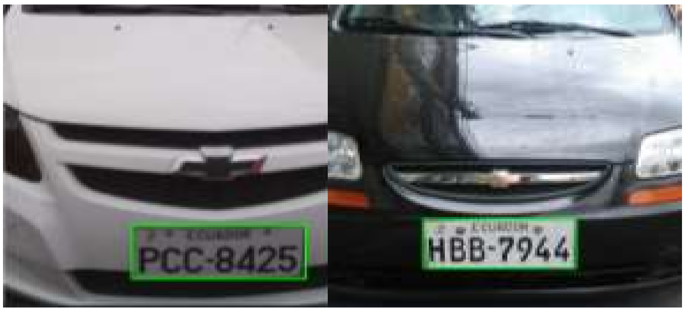


Figure 10. Detection of the vehicle plate.
Source: Prepared by the authors.

SEGMENTATION OF VEHICLE PLATE

Once the vehicle plate in the image has been determined, it is extracted to work with the same applying techniques similar to the previous ones in such a way that an image with characters easy to recognize can be obtained.

Grayscale Conversion

The extracted plate's image of the previous scene is converted to grayscale, aiming to eliminate the borders of it and leave only letters.

Filtering the image

The Gauss filter is applied to eliminate the noise of the image obtained from the plate, using a transformation matrix of 5x5. The size of this matrix was the one that provided better results in the experimental stage.

Binarization

The binarization stage converts the image to black and white, in other words the pixels have values of one or zero. For the correct conversion, the Otsu algorithm is used, which adapts an optimal threshold to avoid the loss of information at the time of converting the image, resulting in images like Figure 11.

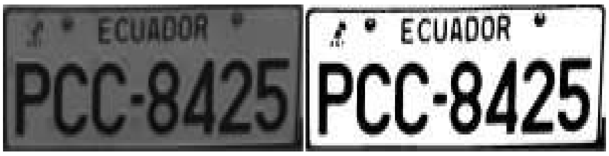


Figure 11. Binarization by the Otsu Algorithm.
Source: Prepared by the authors.

The command used for the conversion is cv2.threshold() which has as arguments the source image, the value that will be assigned to values that exceed or are below the threshold in this case 0 and 255. It is specified through OpenCV (Bradski and Kaehler, 2008) commands the binarization type cv2.THRESH_OTSU + cv2.

Erosion

The morphological erosion operation is applied with a transformation matrix of 4x4 dimension to the previous image, so that the edges of the characters are accentuated. Figure 12 shows the result of applying this operation, however, it has a drawback because all other figures are also accentuated in the image.

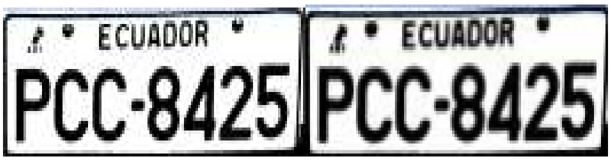


Figure 12. Erosion of plate characters.
Source: Prepared by the authors.

Edge and contour detection

The Canny algorithm is applied with the aim of detecting the edges of the numbers and characters in the image, then shown in Figure 13.

The detected edges are enclosed in a polygon, as shown in Figure 14.

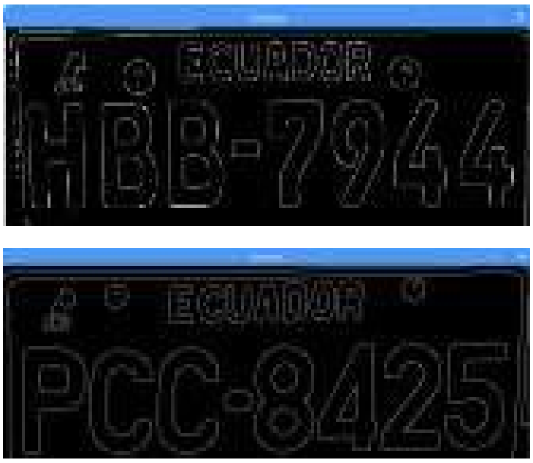


Figure 13. Edges of the vehicle plate.
Source: Prepared by the authors.

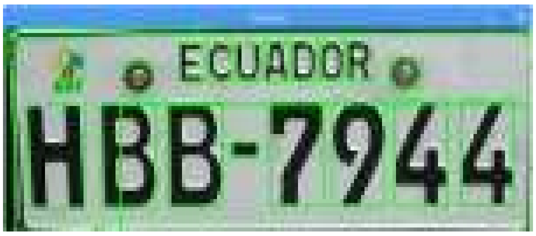


Figure 14. Plate contours.
Source: Prepared by the authors.

Elimination of false positives

The plate contains extra characters which are detected, they must be deleted in order to avoid errors in the recognition, this is achieved by applying a filter based on the size of the characters of interest and subsequently use a closing as shown in Figure 15.

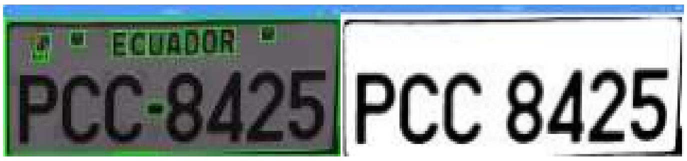


Figure 15. Elimination of false positives.
Source: Prepared by the authors.

KNN Algorithm Implementation

The algorithm based on the principle of the nearest neighbors is implemented for the recognition phase, which requires to be trained to comply with correct detection of letters and numbers.

Algorithm training

The algorithm requires a previous stage of training, which consists of loading an image containing the numbers and characters to be recognized, and indicating the corresponding value of each one, indicating the outline where the image is located, the information of each letter and digit is stored in a plain text file, to later be used at the time of comparing the vehicle plate.

In order to train the algorithm, the image shown in Figure 16 which resembles the typeface of car plate characters has been used. The advantage is that having only uppercase letters the time it takes to train and recognize each spelling decreases. These characters are stored in the back as an array of vectors.

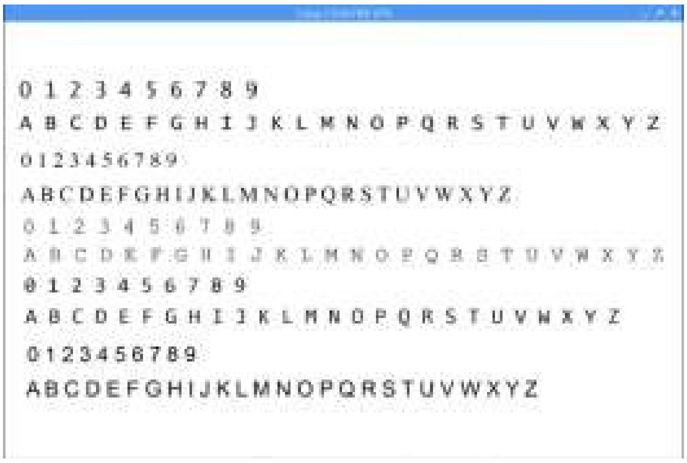

0 1 2 3 4 5 6 7 8 9
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ
0 1 2 3 4 5 6 7 8 9
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
0 1 2 3 4 5 6 7 8 9
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
0123456789
ABCDEFGHIJKLMNOPQRSTUVWXYZ

Figure 16. Algorithm training classes.

Source: Prepared by the authors.

The image is uploaded to the algorithm, and it is operated on it with the aim of eliminating a possible noise, softening and binarizing the image in reverse, meaning white letters on a black background so that the complement of each image corresponds to the same binarized the other way. Then, each of the characters that are going to be valid for recognition are declared, in this case is the whole alphabet except the letter Ñ, and the numbers 0 - 9. This is done using the command ord which returns the ASCII value of the character in its argument.

To recognize each of the characters (Cambronero and Moreno, 2006) within the training image, you define the contours in each of them by drawing a rectangle that demarcates the area or pixels that correspond to a letter or number, followed by this is trimmed each one of them to be resized, which will favor the consistency of recognition and storage of vectors. The process performed is shown in Figure 17. Each character is saved as a vector of floating numbers in a flat file using the numpy.savetxt () command, which stores itself the intensity of the pixels of each digit and letter. This file is going to be useful later for a comparison between the plate and the preset classes.

Characters Comparison

The text file stored in the entry stage must be loaded into the program that will identify the characters, the plain text file contains vectors that characterize numbers and letters depending on the intensity of their pixels.

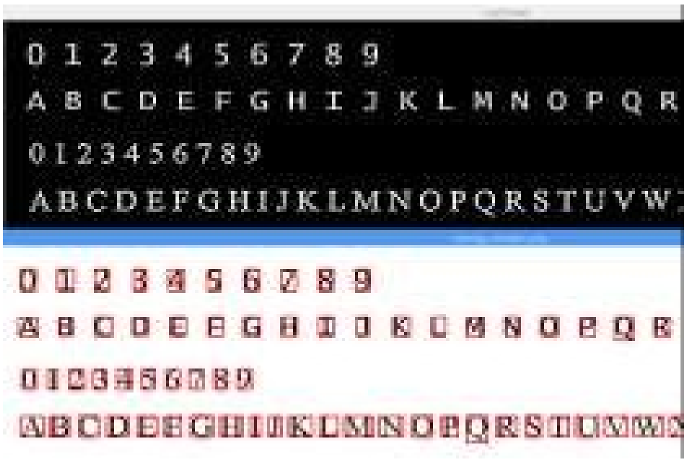


Figure 17. Recognition of training characters.

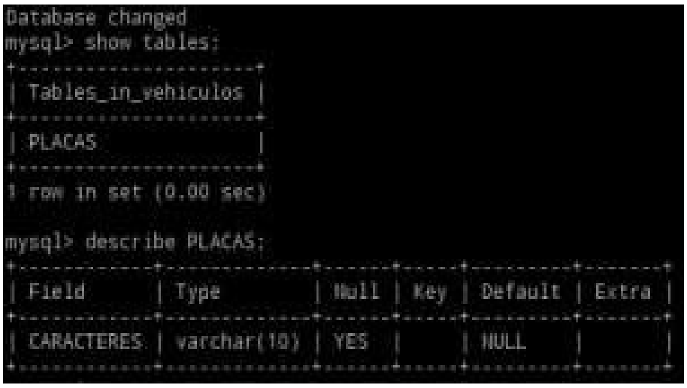
Source: Prepared by the authors.

Database Design

The data obtained from the vehicle plates are stored in a database which is located on a web server, allowing users to query from any device with an internet connection. The management software that manages the database is MySQL due to its compatibility with the Debian operating system.

Creating the Database

The database is created through the command: create database ”db_name”. The create database command allows you to create a database in memory, followed by the name that will be given to it, in this case vehicles. The PLACES table is defined in the database, this table is char extension type 10. Figure 18 shows the description of the Plates table of the vehicle database.



```
Database changed
mysql> show tables;
+---------------------+
| Tables_in_vehiculos |
+---------------------+
| PLACAS              |
+---------------------+
1 row in set (0.00 sec)

mysql> describe PLACAS;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| CARACTERES | varchar(10) | YES  |     | NULL    |       |
+------------+-------------+------+-----+---------+-------+
```

Figure 18. Description of the Database.

Source: Prepared by the authors.

The connection of the database with the program in Python is done with the command MySQLdb.connect, which has as arguments the site where the data, user, password and database are collected. As for the modification of the same is done by SQL statements.

Anchoring to the network

A server was hired due to the optimal characteristics presented by the payment service versus the free service, from which the domain 143.95.106.248 was obtained, where the database is stored. To access it, do so by entering the address http://143.95.106.24/cpanel.

Website

The webpage shown to the user is written in PHP language, which allows you to work with dynamic content. The page mentioned has a textbox to enter the plate for searching and a button to start the search, as shown in Figure 19.

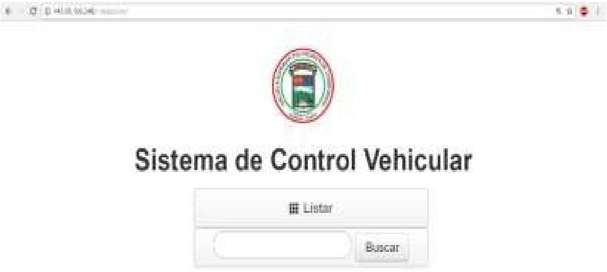


Figure 19. Website.
Source: Prepared by the authors.

Construction

The hardware system consists of a Raspberry Pi3, an RPi camera of 8 Mpx and a case printed in 3D. The entire system is placed on a mechanical arm that allows vehicular passage as shown in Figure 20 The operating system software running on the Raspberry is Raspbian Jessyie, the plate detection and identification program is written in Python language and the database is managed using MySQL.

Figure 20. Mechanical arm system for vehicular Access
Source: Prepared by the authors.

Results

The system is tested with the objective of verifying the percentage of accuracy in the detection and the time required by the system, for which it is taken into account that in average 60000 vehicles circulate in the city, of which a percentage is taken in the performance of tests using Eq 3.

$$n = \frac{z^2.N.P.Q}{E^2(N-1)+Z^2.P.Q} \tag{3}$$

Where:

N = Sample size (Value to be found).

Z = Confidence level (90 % confidence is accepted for engineering). N = Population or universe (Number of vehicles).

P = Probability of success (0.5).

Q = Probability of error (0.5).

E = Sample error. (10 % according to the normal distribution corresponding to a value of Z = 1.65).

The previous result sets the number of vehicles at 68, of which the test is performed with 17 vehicles. Captured images are 1.5 to 2.5 meters apart.

For a better recognition of the characters certain requirements were necessary to comply with the plates of vehicles, among which stand out:

- Correct location of the vehicle plate as dictated by the rules of the Agencia Nacional de Transito.
- The plates should not have frames, ornaments or extras that make it difficult to appreciate the characters.
- The plates must not be damaged.
- Readability of the characters on the board.

Response time

After analyzing the time required to process each plate, the result is an average of 5.84 seconds. The response time of the algorithm for the plates is shown in Figure 21.

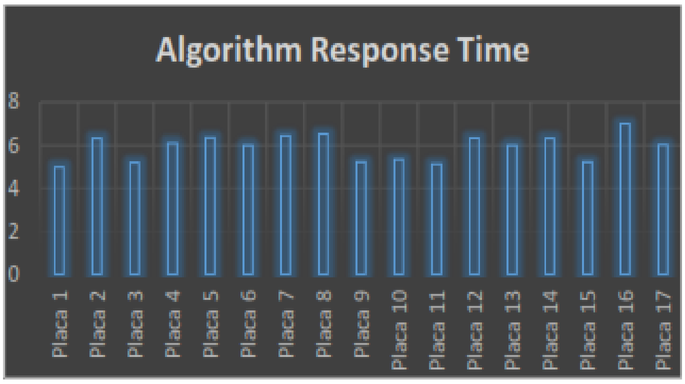


Figure 21. Response time in the processing of the vehicle plates.
Source: Prepared by the authors.

Percentage of success

The average detection process detects 91 % of the data received. So, you can establish that given a margin of error of 10 % the level of trust effectively is 90 % with a tolerance of ± 1 %. The percentages of success are shown in Figure 22.

Being a supervised algorithm, KNN requires an exhaustive training phase, since it depends on the optimal recognition of characters. The more training data per class is implemented the better the result.

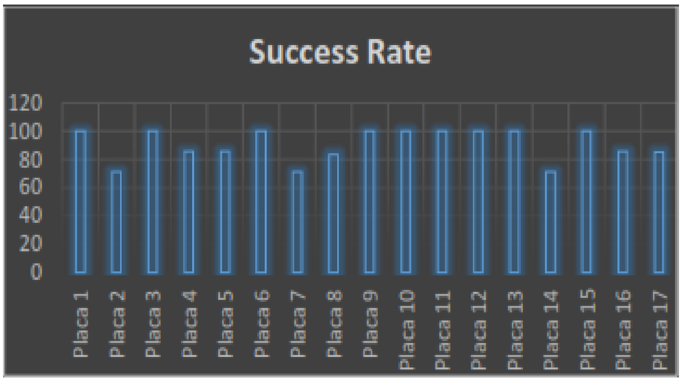


Figure 22. Percentage of accuracy in the detection of vehicular plates.
Source: Prepared by the authors.

The embedded system has been implemented at the main door of ESPOCH, which does not limit that it can be installed in the other entrances only by directing the database to the ”n”systems that are required.

The hardware features of Raspberry Pi3, limit in a certain way the execution of processes entrusted and much more in the case of image processing as these require high computational costs. In contrast to this, this card presents an easy integration with other devices, in this case the use of the camera does not generate delays until you acquire the images, as in the case of IP cameras and USB.

The characteristics of the scenario where the system was put into operation have been considered.

Lighting is the main problem that can be managed by image processing, that is why the segmentation is based on the detection of edges and contours but not on detections of regions.

The embedded system proposes an alternative solution for other access systems that have the same objective. Unlike other commercial vehicles, you should not purchase additional devices to register a vehicle.

CONCLUSIONS

- As it has already been shown, it is very easy to carry out an attack at layer 2 of the OSI model, but equally simple it is to implement measures of safety in the equipments.
- The exploitation of the vulnerabilities in this case can be taken for the comfort of an administrator maintainance configurations by default or by ignoring the characteristics and functionality of the devices.
- The elements of hardware and software used in this work contributed largely to the results of the same. It has allowed us to mount an environment as if you were working with real devices.
- Between various documents of the material collected in this work are a very marked trend regarding the origin of the attacks on network infrastructures, with variations related to the percentages but holding that in the majority of cases the attacks are generated from within the network.
- It should be noted that failures of the security settings in network devices are mentioned a lot. However, we need to bear in mind that vulnerabilities exist in communication protocols and different network devices on the market will be very varied in ways that mitigate these vulnerabilities.

BIBLIOGRAPHIC REFERENCES

Agencia Nacional de Tránsito (2015). Reglamento Para La Elaboración, Entrega Y Control De Placas De Identificación Vehicular.

Álvarez Durán, M. A. (2014). Análisis, diseño e implementación de un sistema de control de ingreso de vehículos basado en visión artificial y reconocimiento de placas en el parqueadero de la universidad politécnica salesiana-sede cuenca. B.S. thesis.

Barcia, O. (2014). Reconocimiento automático a través de visión artificial, correlación estadística y matlab aplicado a las matrículas de vehículos.

Barragán Del Pozo, E. E. (2012). Diseño e implementación de un sistema de control vehicular utilizando reconocimiento optico de caracteres en el laboratorio de automatización industrial de la eis. B.S. thesis.

Bradski, G. and Kaehler, A. (2008). *Learning OpenCV: Computer vision with the OpenCV library*. .^{O}’Reilly Media, Inc.”.

Cambronero, C. G. and Moreno, I. G. (2006). Algoritmos de aprendizaje: knn & kmeans. *Intelgencia en Redes de Comunicación, Universidad Carlos III de Madrid*.

Contreras Hurtado, P. E., Sánchez, M., and Damian, M. (2015). Desarrollo de un sistema de acceso para parqueaderos y cobro tarifario utilizando reconocimiento de patrones. B.S. thesis, Escuela Superior Politécnica de Chimborazo.

Foros Ecuador (2017). Placas de Ecuador (por provincias).

INEC (2012). Ecuador - Encuesta de Transporte 2012 - Información general.

Interpol (2014). Annual Report 2014.

Pajares Martinsanz, G., García, C., Jesús, M., et al. (2002). Visión por computador: Imágenes digitales y aplicaciones.

Paredes Muñoz, J. P. and Guerrero García, L. T. (2012). Estudio comparativo entre algoritmos de reconocimiento de borde para identificación de placas de autos. B.S. thesis, Escuela Superior Politécnica de Chimborazo.

Pérez A., D. A. (2009). Sistemas Embebidos y Sistemas Operativos Embebidos.

Yagual, R., Antonio, C., and Tarira Guerrero, E. E. (2013). Implementación del sistema de registro automático de las placas vehiculares utilizando reconocimiento óptico de caracteres y visión artificial, en la garita 1 de la universidad estatal península de santa elena. B.S. thesis, La Libertad: Universidad Estatal Península de Santa Elena, 2013.