Linux Embedded System for Vehicle License Plates Recognition

Josué Batista Mota
Computing Department
Federal Institute of Ceará
Maracanaú Campus, Brazil

Renata Imaculada Soares Pereira
Electrical Engineering Department
Federal University of Ceará
Pici Campus, Brazil

Sandro César Silveira Jucá
Computing Department
Federal Institute of Ceará
Maracanaú Campus, Brazil

ABSTRACT

This article proposes an embedded system with the purpose of vehicle license plates recognition. For the development of this project, it is used the OpenAlpr library and the Python programming language to implement the recognition of the license plates. For the system development, a Linux embedded system based on the free software Raspbian and the Raspberry Pi platform is selected. A conventional webcam for image recognition is applied. The proposed project hit rate reached around 92%.

General Terms

Image Processing; Conception of Circuits, Integrated Systems and Embedded Systems; Internet of Things.

Keywords

Raspberry Pi, OpenAlpr, vehicle license plates.

1. INTRODUCTION

Following the principle of Internet of things, a term created by Kevin Ashton in 1999, any object can have access to the internet, making it possible to establish a remote connection between two or more devices, creating innumerable possibilities for both the academic and industrial environments [1].

Many places have a big flux of vehicles, like for example, condos, enterprises, shopping centers and universities. However, many of these places do not have any kind of monitoring system. This research seeks to develop a low-cost embedded system to monitor the flux of vehicles by recognizing licence plates.

2. STATE OF THE ART

The project approached by Martins [2] has a differentiated focus which is the reading of traffic plates, however the subject remains the same. It utilizes a Linux embedded system, the Raspberry Pi 2 B+, to operate segmentation of the acquired pictures by the camera, sending commands via an Arduino Uno board and then controls the car doing maneuvers as told by the plates. The accuracy reached 96.6% of success and had 250 segmented images of plates for learning, and three intermediary layers having 350 neurons on the first layer, 250 on the second, 150 on the third.

A method developed by Nardi [3] creates a system to control the access of vehicles based on the digital reading of characters on images. A Linux embedded system was utilized based on the Raspberry Pi and the OpenAlpr library with some modifications to serve better the purpose of the project. The complete system is composed of a distance sensor which on triggering, allows the system to start capturing the image of the vehicle, to recognize the license plate.

The form introduced by Penha [4] aims to create software for plate readings. Is divided by three modules: localization, extraction of the plate which is based of the tonal variation method, the segmentation of the characters which implements the vertical projection method and the recognizing of the segmented characters realized by the ELM (Extreme Learning Machine) algorithm. The hit rates during the stage of learning were 100% while the average of the hit rate during the test stage reached 96%.

The project developed by Leonardo [5] makes the identification of the license plate of a vehicle and assesses if it can enter a certain local. It consists of a webcam and a computer with an Intel processor and Windows operating system. It has a steel rate of around 93.5%.

The proposal presented by Filho [6], intends to create a real-time vehicular tag localization algorithm using the OpenCV open source library with the CUDA, which is an application programming interface (API) designed for parallel computing developed by NVIDEA (hardware manufacturer). The solution uses the Canny edge detection algorithm where it finds a wide edge margin in the image, following the application of mathematical morphologies. The proposed system presented a 95.5% success rate in the location of vehicle plaques.

All the cited projects do the reading of a determined object, in this case, the vehicle license plate. Thus, the present paper propose a Linux embedded system based on the Raspberry Pi platform and a conventional webcam using the OpenAlpr library and the Python programming language to implement the recognition of the license plates.

3. MATHERIALS AND METHODS

This section has the objective to describe the components used to develop hardware and software of the proposed project.

3.1 Raspberry Pi

Raspberry Pi (RPi) is a microcomputer that has the size of a credit card, having the possibility of being connected to a monitor via HDMI, also having USB ports for keyboard and mouse. RPi uses a MicroSD card as the hard drive, storing the operational system (OS) and system data [7].

Every RPI hardware is integrated on a single board. The primary objective was to promote the teaching of basic Computer Science in schools [8].

In the last years, the RPI grew in popularity due to simplicity and practicality in a large array of projects [9]. Currently, the device is on the third generation, the Raspberry Pi 3 Model B+. However, the model Raspberry Pi 1 Model B+ utilized has the required specifications for this project.

3.2 Webcam

For the development of the proposed project, a conventional webcam which utilizes USB communication is used. The model chosen was the Playstation eye, a digital camera developed by Sony [10]. Any kind of webcam can be used, but we must consider resolution and image quality.

3.3 Raspbian

Raspbian is an operational system based on Debian Linux that is the official one for the Raspberry Pi Foundation embedded systems. It is responsible for processing the instructions for the license plate reading. Raspbian comes pre-installed with software for programming and educational purposes. Having Python, Scratch and Java, for example. There are two types of visualization:

- Raspbian stretch with desktop has a main environment, that is capable of fast access to many applications e has about 4 GB (Gigabytes) of storage.
- Raspbian stretch lite has functions executed in own terminal and has around 2 GB (Gigabytes). This version is the one utilized on the project due to versatility and lightweight version.

3.4 Libraries

The Raspberry Pi is accessed remotely through the Secure Shell protocol (SSH) present in Linux. This is a cryptographic network protocol for the operation of network services, in a safe manner on a unsafe network [11]. Starting there, the following libraries were utilized as well:

- OpenAlpr is a vehicle license plate recognizer library, written in C++ based on C, Java, Node.js, Go and Python. It analyses images and video to identify the license plate characters [12].
- Leptonica is the library that deals with image processing and helps the other libraries in the format and operation of input and output for images (jpg or png, for example). Both the OpenCV and the Tesseract utilizes this library.
- Tesseract is a software that realizes the recognizing of the characters on the images. It compiles the image in a way that it can capture columns of characters and show them as the output in a text file.
- OpenCV is a free software written in C++, with functions in Python, Java and Matlab of computer vision and machine learning, having more than 2500 optimized algorithms ready to use and to be modified. Many known enterprises such as Google, Microsoft and Intel Applications utilizes OpenCV, with monitoring devices on mines in China, to detect drownings on pools in Europe and many more applications.

3.5 Hardware

The basic hardware for the proposed project consists of the webcam and the Raspberry Pi (Figure 1). The only requirement is to connect both devices via USB cable.

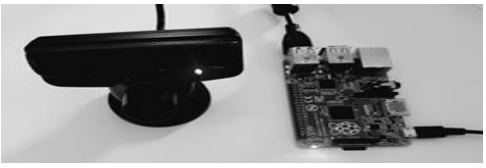


Fig 1: Utilized Webcam and Raspberry Pi

3.6 Implemented Software

The flowchart of the implemented code is showed on Figure 2.

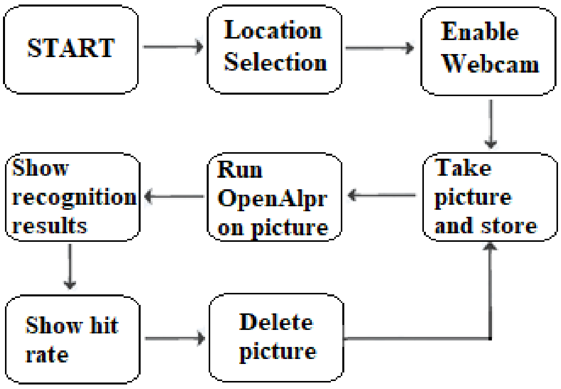


Fig 2: Flowchart of the system

As showed on Figure 2, the first step is to select the location of the license plate to be recognized. After that, the code executes an infinite loop which captures the images and executes the algorithm with OpenAlpr library. After identifying, the image is deleted, avoiding the accumulation of files in the SD memory card.

4. RESULTS

The miss rate is rated according to the OpenAlpr libary.

4.1 American license plate

In Figure 3, it is shown the license plate utilized for the project tests. It belongs to Louisiana in United States of America (USA). While Figure 4 shows the software confidence level.

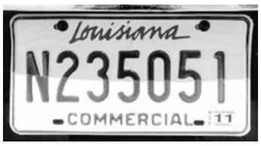
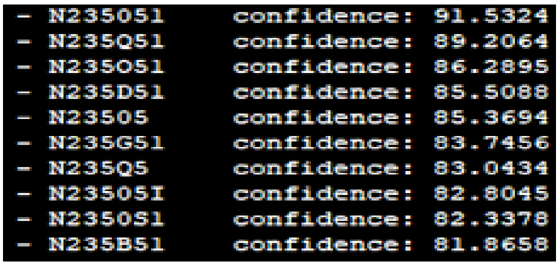


Fig 3: American license plate



```
- N235051     confidence: 91.5324
- N235Q51     confidence: 89.2064
- N235O51     confidence: 86.2895
- N235D51     confidence: 85.5088
- N23505      confidence: 85.3694
- N235G51     confidence: 83.7456
- N235Q5      confidence: 83.0434
- N23505I     confidence: 82.8045
- N235OS1     confidence: 82.3378
- N235B51     confidence: 81.8658
```

Fig 4: Results and level of confidence

For initial tests, the software was executed ten times (Figure 4). It is shown that on the first tries (looking down to up) the desirable output is not reached, showing instead the result N235B51 and with the confidence rate below 90%. The more readings are done, the percentage has the tendency to go up. On the last try, the correct result of the car license plate (N235051) with the confidence percentage of 91.5% is obtained. It is important to emphasize that doing multiple tests with images, the system acquires more precision on reading the car license plates, that is, learns to do the proposed function. Based on machine learning, evolves the knowledge on patterns and the theory of computational learning on artificial intelligence. Therefore, only the last reading is taken into consideration, due to a tendency of having the bigger recognition hit rate.

The program was done using the US vehicle license (Figure 3). Figure 5 presents the graphic with the hit rate results of eight executions of the software.

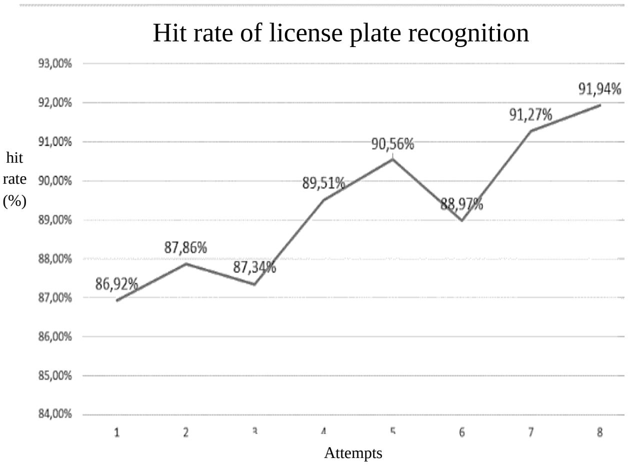


Fig 5: Hit rate results for eight executions

On Table 1, a comparison of the hit rate on nine different American license plates.

Table 1. Comparison of American license plates

License plate	Recognition	Hit rate
FDY418	FDY418	92%
IVA0B70	IVA0B70	92%
GNK4019	GNK4019	95%
6PYV308	6PYV308	86%
CNY7445	CNY7445	92%
8JP698	8JP698	90%
CLRFRM	CLRFRM	95%
PAPL8S	PAPL8S	94%
J39999J	J39999J	91%

4.2 Brazilian license plate

It can be mentioned that there is still no complete support for Brazilian vehicle plates, however, using the parameter for European plates, a relatively satisfactory result is obtained, as shown in Figure 6 and Figure 7.

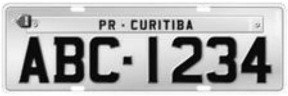
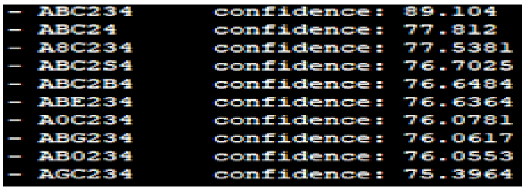


Fig 6: Brazilian license plate



```
- ABC234      confidence: 89.104
- ABC24       confidence: 77.812
- A8C234      confidence: 77.5381
- ABC2S4      confidence: 76.7025
- ABC2B4      confidence: 76.6484
- ABE234      confidence: 76.6364
- A0C234      confidence: 76.0781
- ABG234      confidence: 76.0617
- AB0234      confidence: 76.0553
- AGC234      confidence: 75.3964
```

Fig 7. Results of the Brazilian license plates

It can be noticed that the recognition presented hit rate from 75% to 89% since the software could not read all the characters. Generating in the last result the following analysis: ABC234. The hit rates of six different Brazilian license plates analysed are presented on Table 2.

Table 2. Brazilian license plates

License plate	Recognition	Hit rate
BAJ-4441	BAJ444I	84%
AXX-1773	AXX773	90%
CHS-9S64	CHS9564	86%
EDD-5690	EDDS690	92%
BAG-7751	BAG775I	88%
FMI-1303	FHI303	86%

It is seen that on some license plates, the recognition result is incomplete, generating hit rate from 84% to 92%. This is due to the lack of a license plate Brazilian pattern on the OpenAlpr library. However, if the software is trained, that is, doing multiple executions on similar plates, the software has the tendency to output better results.

Finally, we make a comparison between the presented project and the one researched, as can be seen in Table 3. The other project that implements RPi 1 B+ reached only 70% confidence, while the presented reached a 91.8% success rate. The best results from the other projects reached 96% and 96.6%, only 4.5% above the project shown.

Table 3. Similarities between projects

Projects	Hardware	Hit rate
Martins [2]	Raspberry Pi 2 B+	96.6%
Nardi et. al [3]	Raspberry Pi 1 B+	70%
Penha [4]	No information	96%
Leornado [5]	Computador	93,5%
Filho [6]	CUDA	95,5%
Project proposed	Raspberry Pi 1 B+	91.8%

5. FINAL CONSIDERATIONS

The project proposes a practical solution, original and with low-cost, using free software for vehicle license plate recognition. The library OpenAlpr, besides of showing good results, can be used on a variety of computing languages, adapting to every programmer. The application was satisfactory, because the developed software reached good results, with hit rate around 92%, making the project highly viable for applications with Linux embedded systems on the Raspberry Pi platform.

The design must be improved aiming at a perfect reading for the Brazilian plates. Soon the software must undergo a certain training, that is, make numerous recognitions to acquire a better accuracy in the analysis of the plates, having the possibility to be used in places with flow of vehicles.

6. REFERENCES

[1] Mancini, M. 2017. "Internet das Coisas: História, Conceitos, Aplicações e Desafios" *Pmi-Sp*, p. 9.

[2] Martins, M. A. 2017. "Identificação de placas de trânsito através da classificação de imagens usando redes neurais artificiais," p. 45.

[3] Nardi, E., Polak L. and Valiati, L. 2015. "Sistema De Controle De Acesso Utilizando Reconhecimento De Caracteres Em Placa De Automóvel".

[4] Penha, F. J. S. 2013. "Implementação de um sistema de reconhecimento de caracteres em placas de automóveis baseado em Extreme Learning Machine".

[5] Leite, L., Antonello, R. 2017. " Identificação automática de placa de veículos através de processamento de imagem e visão computacional".

[6] Filho, J.F.S.P., Souza, A.C.S. 2014. "Localização de Placas de licenciamento veicular em tempo real utilizando OpenCV com CUDA".

[7] Hossain Y. and George F. P., "IOT based Automated Intrusion Detection System". 2018. Vol. 180, no. 35, pp. 56–61.

[8] RPi Foundation. 2016. Raspberry Pi - Teach, Learn, and Make with Raspberry Pi. [Online]. Avaiable at: https://www.raspberrypi.org/.

[9] Rayaji S., "3D Image Reconstruction using Raspberry Pi". 2018. Vol. 181, no. 4, pp. 8–13.

[10] Sony Entertainment Network, "PlayStation Eye," 2007.

[11] Ylonen T., "The Secure Shell (SSH) Connection Protocol," 2006.

[12] Hill M., "OpenAlpr". 2018. [Online]. Avaiable at: https://github.com/matthill.