




A Low-Cost Smart Surveillance System Applied to Vehicle License Plate Tracking

Francis de Melo Marconcini , Marcelo Eduardo Vieira Segatto , Evandro Ottoni Teatini Salles 
Department of Electrical Engineering, Federal University of Espírito Santo, ES, Brazil
francis.marconcini@hotmail.com, segatto@ele.ufes.br, evandro@ele.ufes.br

Abstract— The growth of urban centers has a major impact on our lives and on our society, one of which is the increase in the vehicle fleet, which in turns is a major offender in increasing road congestion, pollution, accidents, theft of vehicles, among others. With a focus on minimizing problems with theft of vehicles in an urban environment this paper proposes a low cost Internet of Things for smart surveillance system. The central element of the system is attached to a vehicle that travels in an urban environment making Automatic License Plate Recognition of the vehicles license plates in front of it. The information collected is forwarded to a cloud service that is responsible for monitoring restricted vehicle license plates to view positioning, time, date and statistics related to restricted vehicle license plates. Experimental results obtained in test paths in an urban environment show promising results. The low cost of implementing the system allows it to be scaled to a fleet of vehicles, making it possible to build a mobile intelligent surveillance network, amplifying the monitoring capacity of an urban area.

Index Terms— Internet Of Things, Smart Surveillance, Automatic License Plate Recognition

I. INTRODUCTION

In recent years in Brazil, there has been a drop in vehicle theft, using violence of around 30% and a drop of 12% without using violence [1]. Several factors, such as the country's economy and public policies, have a strong influence on the country's decrease in crime and, if coupled with technological advances applied to security, such decrease maybe even more significant.

Surveillance systems are traditionally composed of Closed-Circuit Television (CCTV) cameras, which capture images to be supervised by security officers, who must be very attentive to the events that occur on each of the screens in the monitored locations. Currently, the technological development of electronic devices such as wireless sensors [2] combined with advances in computer vision and digital image processing has enabled the Internet of Things (IoT) to assume a fundamental role in the area of surveillance, creating the new concept of Smart surveillance. No less important than patrimonial monitoring, restricted areas monitoring, industries surveillance, the monitoring of vehicles on urban roads stands out, providing security agencies with the ability to track restricted vehicles. According to [3] automatic license plate recognition systems(ALPR) traditionally have some stages before the recognition of the vehicle plate itself. They are: image acquisition, extraction of the vehicle plate, segmentation of the vehicle plate, and recognition of the characters contained in the license plate. In real-time systems, the hardware and software that perform such stages must be robust enough to apply the necessary algorithms at each stage and on each frame of the captured video streaming. It must

be taken into account that the higher the resolution of the image, the more chance one has to extract the license plate and perform the recognition of its characters, however more computational power is required to process the entire high resolution image matrix.

To reduce costs with high computational power hardware for vehicle license plate recognition in real-time, this article proposes to restrict the area of application of the system to an urban environment, which limits vehicles speed because many retention points such as traffic jams, pedestrian crossings, traffic control lights, among others. The constant decelerations caused by these retention points cause vehicles to travel at very low speeds or stop in a row and close to each other, which makes continuous capture of video images unnecessary. Then the same image frame would be captured many times and the computational capacity would be wasted by performing the stages of extraction, segmentation, and character recognition of the license plate. Based on these characteristics, the proposed IoT system acquires of high-resolution images spaced by a period of time necessary for the low-cost hardware to be able to execute the routines of the license plate recognition software and send the data to the cloud service responsible for storage location information (latitude, longitude), time and license plate. The purpose of the system is to operate in a time close to real-time and at a low cost compared to ALPR real-time platforms that process high-definition videos. The rest of the paper is organized as follows: Section II describes the works related to enabling technologies for the implemented system. The architecture of the system will be described in section III. The implementation of the system will be described in section IV. Section V presents practical tests. Finally, the conclusions are given in section VI.

II. ENABLING TECHNOLOGIES

Intelligent surveillance systems consist of the use of automatic video analysis technologies [4], which are intended to fill human attention lapses. Legacy surveillance systems rely only on video capture by camera systems spread over strategic locations in the monitored environment, subjected to human analysis to investigate incidents that have occurred or even perceive any suspicious behavior in the environment in real-time. The evolution of information technology coupled with advances in the areas of robotics, electronics, computer vision, and IoT, allowed surveillance systems to evolve into much more elaborate architectures such as the proposal in [5]. For example, the use of wearable sensors in police forces, microphones, and cameras scattered throughout the city, sensors coupled to police vehicles and drones, all capturing data to submit to intelligence algorithms capable of crossing information to prevent incidents or even catastrophes. As discussed in [4], this is the break in the paradigm brought by smart surveillance systems that have moved from incident investigation systems to incident prevention systems. As can be seen in [6], in these types of smart systems, events are recorded and trackable and can be revised at any time through simple queries in the database. Not only limited to environments where heritage monitoring and access control are required, smart surveillance systems can also be applied to more dynamic environments such as urban roads through which vehicles travel. Some cities in Brazil already have smart monitoring systems which form a security fencing such as Vitória in the state of Espírito Santo, Brazil [7]. It is a system with 86 fixed, high-resolution cameras installed in 24 barriers around the city, where photos of car license plates are recorded and recognized by an ALPR system, so if there are any restrictions related to vehicle license plates, the institution responsible for security will be triggered. Decentralized and layered architectures, as proposed in [8], can bring a

gain in optimization in smart surveillance systems by reducing the response time and increasing the scalability of the system.

A. Decentralized architectures

Centralized monitoring systems depend on robust computational architectures, which need to meet the system response time requirements for the various monitored target areas. The infrastructure must grow proportionately as the system's coverage area increases. Considering these challenges, layered and decentralized architectures, as proposed in [8], can bring optimization gains to intelligent surveillance systems, reducing response time and increasing scalability. In [9] the architectures for smart surveillance are based on edge computing and fog computing [10], paradigms where complex computation is performed on the edge devices of the system. This new paradigm is a proposal to overcome the deficiencies of cloud computing [11], [12]. Challenges such as scalability, ubiquitous sensory data access, event processing overhead, high networking traffic, and massive storage can be overcome through Fog computing, which aims to support the critical latency of smart surveillance applications, promoting proximity between cloud computing and the data producers. It also eliminates long round-trip times previously introduced by the cloud infrastructure used for analytics, as well as cost savings, energy savings, and reduced bandwidth consumption [13]. Currently, Mobile crowdsensing (MCS) [14] architectures become feasible through decentralization and use of Mobile Edge Computing (MEC). The high demand for computing resources as concurrent connections on the backend of services running on the cloud, real-time processing power, as well as device management functions (where context tracking is kept in the application) where its refresh rate is very dynamic, only a decentralized architecture based on MEC can bring better performance and evolution for the application in question. Otherwise, problems such as increment high loads on the mobile network and backhaul occur, generating unwanted side effects such as bottlenecks and communication delays. The advance in mobile communications made it inevitable that the MEC would be adopted. As quoted by [15] the communication systems from generations 1 to 4 (1G to 4G) were intended to increase wireless speeds to support the transition from voice-centric to multimedia-centric traffic. The fifth generation (5G) of mobile communications has a different purpose from previous generations. This generation must support communication, computing, control, and content delivery, where applications require unprecedented high access speed and low latency. For these requirements, essential for the services of this new era of communications is that architectures based on MEC are adopted. Bringing storage and computing capacity closer to users and making the new communication trend in which traffic is consumed and produced locally is met within specified parameters for a 5G network.

B. IoT

The IoT has revolutionized ubiquitous computing and is expanding very rapidly. Some years ago it was projected that in 2020 we would have around 50 billion devices in the world [16], which is not possible to know for sure today, but it is noticed that exponential growth happened. Devices such as Radio-Frequency Identification (RFID) tags, sensors, actuators, mobile phones, etc., are increasingly integrated into our lives, changing the way we interact with the world. In [17], the reduction in terms of size, weight, energy consumption, and cost of radio devices is mentioned as a crucial point for the expansion of the IoT concept, which allowed practically all objects to have integrated radios, enabling

a simple object to have connectivity with people and any other "thing" that also had communication capacity. With the advent of 5th generation networks (5G networks), IoT technology may have improved connectivity capabilities in addition to security, reliability, coverage, low latency, and meet a very wide range of new applications. According to [18], 5G networks will guarantee IoT the necessary quality through its features such as enhanced Mobile Broadband (eMBB), massive Machine-Type Communications (mMTC), and Critical Communications (services with ultra low latency).

As a revolutionary technology, IoT could not be left out of the fourth industrial revolution, known as Industry 4.0. This technology is based on the Cyber-physical System (CPS) [19], a discipline that involves computer engineering and communication systems that interface with the physical world. This concept of CPS makes this technology closely linked to IoT which in a way also involves computer engineering and communication systems interacting with the physical world. In [20] the author deals with the concept of CPS more completely, leaving the vision of IoT with one of the technologies that integrate it. With the reduction in terms of size and cost and increase in the computational power of computers, IoT devices have been able to contribute strongly to the evolution of IoT architectures based on Edge Computing. Tasks with more complex operations could be solved directly on these edge devices, as long as they had no power restrictions. In [21] the Single Board Computer (SBC), can be used as a cheaper alternative and compatible with the existing computer vision software compared to the use of field programmable gate arrays (FPGAs) and General Purpose embedded hardware. In this category of SBC, the Raspberry Pi [22] stands out, a device that fits in the palm of the hand, but with a high computational capacity capable of performing computer vision tasks, thus being an alternative to decentralized systems based on edge computing. The following can be mentioned as advantages of using a Raspberry Pi in an architecture based on edge computing:

- A flexible device that supports several operating systems (among them, several Linux distributions and windows)
- A storage memory expansion capacity greater than 64 GB (depending on the SD card used)
- Supports working on multi operating processor (Broadcom BCM2837 64-bit processor), operates at 1.2 GHz
- Four USB ports, 40 GPIO pins for interface
- Support for C / C ++, Python and Java programming languages
- 5V power supply, which also makes it easy to connect to solar cells or batteries
- Can run in server mode, such as a web server, handling numerous requests.

C. Automatic License Plate Recognition

Vehicle license plate monitoring systems are being increasingly requested by public security and traffic inspection bodies to monitor vehicles with traffic restrictions and to inspect vehicle taxes respectively. The great challenge of automatic plate recognition is to extract the characters representing the identification of the vehicle license plate from the image. Numerous obstacles increase the degree of difficulty of the vehicle license plate recognition challenge. Among them are the lighting conditions of the environment where the images are being captured, the quality of the captured image, the angle at which the vehicle plate is fixed to the vehicle, dirt causing distortions in the interpretation of digits, plates of different types, colors, and formats, among others. According to [3], an ALPR system consists of four stages:

- Image acquisition: Acquisition of the image itself, encompassing the entire vehicle.
- License Plate Extraction: Identification of the license plate object and its extraction from the image.
- License Plate Segmentation: From the extracted license plate, it is segmented to extract the characters to be recognized.
- Character Recognition: Based on the segmented characters, they are recognized to obtain the final result, which is the license plate identification.

Each step must be carried out successfully so that a result of satisfactory quality can provide accurate information for the next step. In this work, the ALPR stage will be performed through OpenALPR [23], which is based on the OpenCV libraries [24], responsible for all image manipulation, and Tesseract [25], which is based on the technique Deep Learning (DL) of Long Short Term Memory neural networks [26] (LSTM). These, in turn, mimic the human system of long and short term memory and have the advantage of overcoming the problems of error backpropagation [26] found in the back-Propagation Through Time networks [26], [27]. ALPR systems such as the one implemented in [26] use LSTM networks to locate the characters on the vehicle plate, as well as extract character characteristics. Currently, widely used networks for image recognition and consequently in ALPR are the Convolutional Neural Networks (CNN). The authors of [28] and [29] use these networks in conjunction with the LSTMs to perform feature extraction, locate the characters on the plate and recognize them. Advanced image recognition techniques enrich ALPR a lot, however, using them with the tuning of the system's parameters set as discussed in [30] oriented to the application makes it possible to reach a state of the art.

III. SYSTEM ARCHITECTURE

The proposed system architecture offers a decentralized smart surveillance system model based on IoT services. Using an SBC (Raspberry Pi 3 Model B) with an integrated camera and GPS, OCR will be performed on the vehicles license plate that will have their images captured along with their GPS coordinates and then sent to a remote database. This system will act as an Edge Computing, taking care of all the image processing necessary for the recognition of the vehicle license plate. The purpose of the system is that only authorized public security institutions have access to the visualization of the plates recognized by the system, the collected data will be transmitted encrypted to the cloud for storage and analytics. An approach focused on data security and privacy will not be the focus of this work. Fig. 1 presents a diagram of the application context of the system.

The central element of the System is a Raspberry Pi 3 Model B, under which the Raspbian OS operating system runs. Fig. 2 shows the implemented architecture on the central element of the system.

The elements of this architecture are as follows:

- Raspberry Pi 3 Model B: A SBC Hardware responsible for providing the necessary computational resources for Optical Character Recognition (OCR) and transmission to the cloud. In this paper, this is our element of edge computing.
- GPS - NEO 6M: Hardware responsible for GPS communication. It is connected to Raspberry Pi 3 through a Serial interface and the acquisition of its data is done through the Python Pynmea2 library.
- Camera: Hardware for image acquisition. It is attached to the Raspberry Pi 3 through a CSI interface. The images obtained by the camera are managed through the Python PiCamera library.

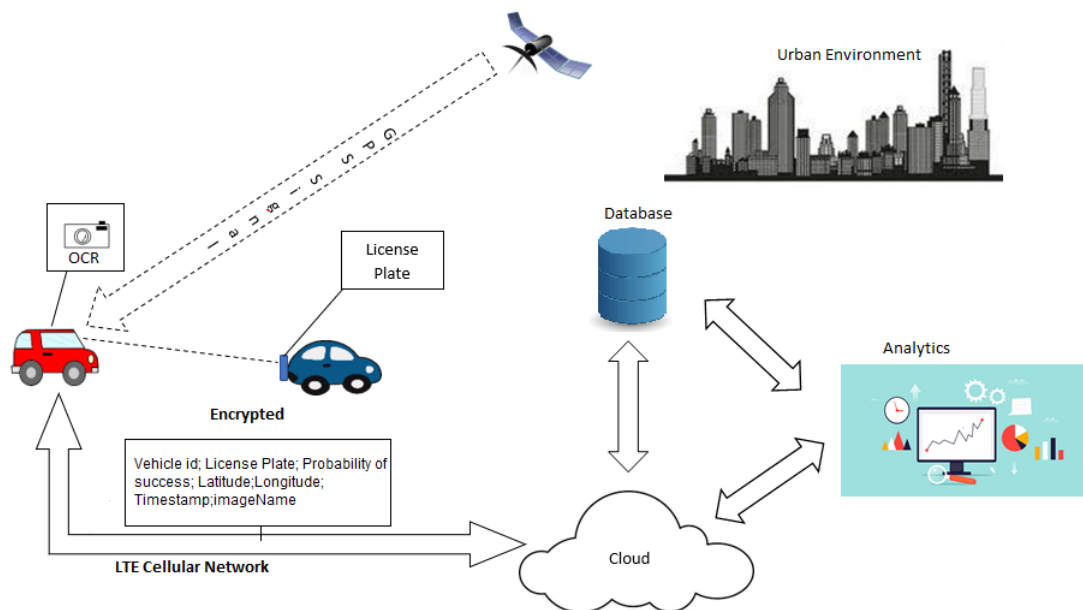


Fig. 1. System application context diagram.

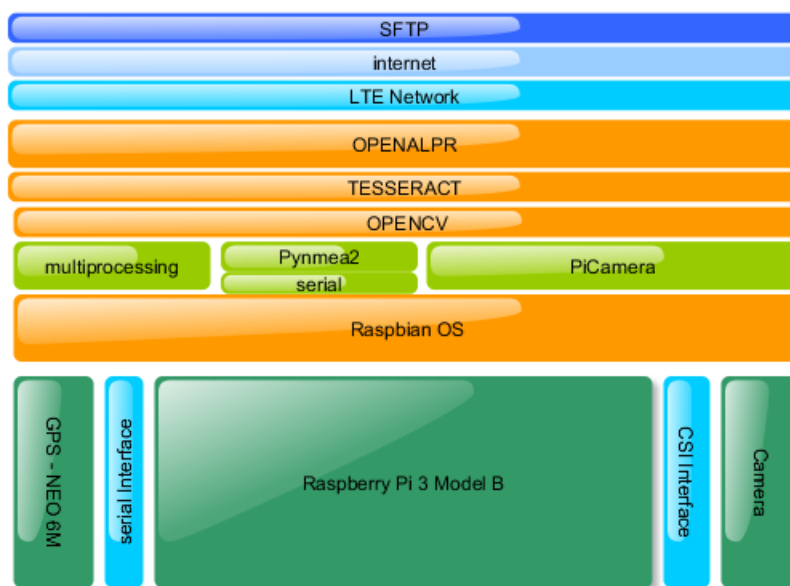


Fig. 2. Prototype architecture.

- Multiprocessing: A Python library with support multiprocessing, was used to optimize the recognition process of the Vehicles License Plates.
- OpenCV: an Open Source library with computer vision algorithms.
- Tesseract: A library that contains an OCR engine.
- OpenALPR: An Open Source library for Automatic License Plate Recognition.
- Internet: The internet will be reached by the system through an LTE cellular network, through a smartphone with a router function enabled.
- SFTP: SSH File Transfer Protocol. It will provide a secure channel for transmitting files containing License Plates information to the database.

IV. SYSTEM IMPLEMENTATION

The system was created to capture images of license plates of vehicles in an urban environment and transmit them to storage and analytic server in the cloud, requiring dedicated software for image processing, character recognition, and transmission of information safely. The system has its structure based on the architecture shown in Fig. 2, which will act in the scenario presented by Fig. 1.

A. Hardware design

As can be seen in Fig. 3, the central processing node is a Raspberry Pi 3 Model B, which has a five megapixel fisheye camera, CCD: 1/4 "connected via CSI interface and a maximum resolution of 2592 x 1944 pixels.

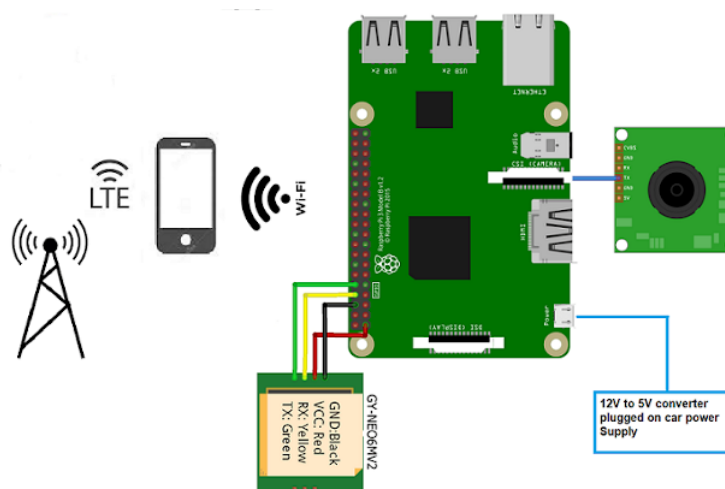


Fig. 3. Connection of hardware elements.

The central element cost is shown in Table I.

TABLE I. CENTRAL ELEMENT COST.

Item	Cost
Raspberry pi 3 Model B (1GB RAM)	\$ 80.00
Raspberry Pi Camera	\$ 19.00
Car power Supply 12V to 5v	\$ 12.00
GSP Module Neo-6M	\$ 12.00

B. Software

The main software component of the system is OpenALPR, which performs the OCR of the vehicle license plates. This is based on the OpenCV and Tesseract libraries, running under the Raspbian OS [31]. For OpenALPR operation, the following dependencies were installed: OpenCV in version 2.4.8 and Tesseract in version 3.0.4. Below is a diagram of the logical flow executed by the system represented in Fig 4.

When the system is turned on, it starts a loop that captures four images in sequence, through the camera. These images are submitted to processes in parallel running OpenALPR through the multiprocessing library used in the python software that manages the flow of Fig. 4. The multiprocessing

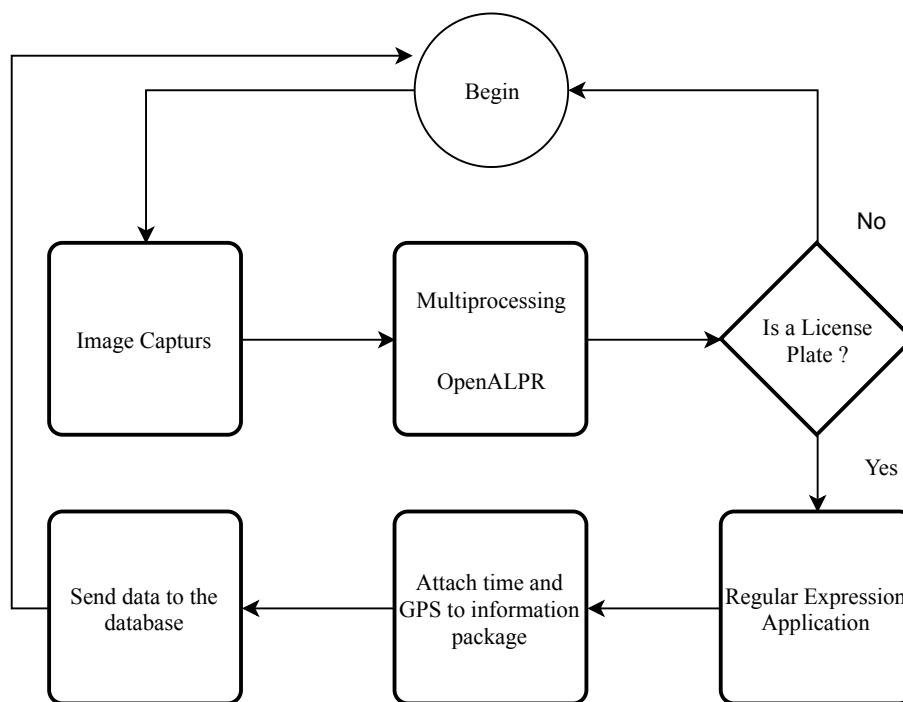


Fig. 4. Software flow.

library makes it possible to process the four images obtained initially in parallel, making use of the four cores of the 64-bit ARM Cortex-A53 1.53 GHz processor of the Raspberry Pi to run OpenALPR processes. During the image capture loop, the camera ISO parameter, which controls the camera sensitivity to light, is configured according to the current time, as shown in Table II. Low ISO values imply low sensitivity, while high values are used for low lighting conditions [32]. The values adopted in Table II were obtained empirically. Another parameter worked on in the system is the shutter speed. The shutter speed is limited by the framerate (fps), like a shutter speed less than 1/fps is not possible. A slow shutter speed leaves target objects in motion with a blurry appearance. To avoid blurring, it is ideal to have a shutter speed of fewer than 0.001 seconds for free traffic applications [33]. In this way a long shutter speed, the image is blurred, whereas, for a fast shutter speed, the image is more frozen, bringing clarity in presence of a moving object, but the sensors capture less light. Thus, during the period when there is no more sun, we keep a long shutter speed, so that it is possible to capture more lighting and view the vehicle’s license plate. During the day it is left it close to the inverse of the frame rate since there is a lot of lighting and the targets are almost always in motion. After capturing the images, if any plate has been recognized, that is, it follows the alphanumeric pattern of License Plates in the Brazilian standard as shown in Fig. 5 and Fig. 6, the time and GPS information obtained and added to the License Plate information for that are then routed securely to the remote database.

TABLE II. CAMERA ISO CONFIGURATION.

	06:00 AM > Hour > 19:00 PM	18:00 PM > Hour >= 19:00 PM	18:00 PM >= Hour >= 06:00 PM
ISO	400	300	60

The information is transmitted to the remote server in text file format, protected by the SFTP protocol. This file is kept in the Raspberry Pi until the SFTP transmission occurs successfully, otherwise new



Fig. 5. New Brazilian License plate model [34].



Fig. 6. Legacy Brazilian License plate model [35].

sending attempts are made at regular intervals until they are successful according to the diagram in Fig. 7.

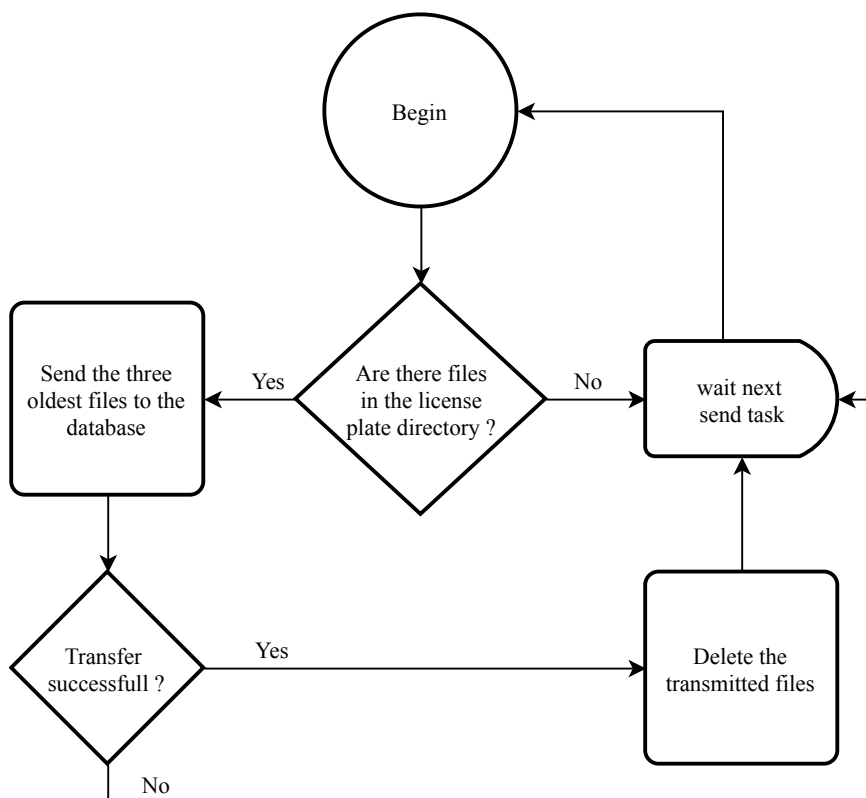


Fig. 7. File transmission flowchart.

The capture and recognition hardware will be coupled to a test vehicle hatch with 1474 mm high and 3892 mm long. This hardware has a slight inclination towards the ground, about 18 degrees, so the distance to the vehicle in front that provides an image with adequate quality is between 2 to 5 meters, as can be seen in Fig. 8 and Fig. 9.



Fig. 8. View of the hardware attached to the rear view of the test vehicle.

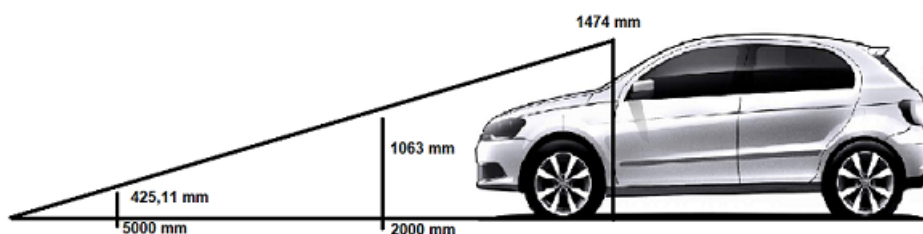


Fig. 9. Viewing angle of the hardware for image capture.

V. PRACTICAL TESTS

To validate the implementation of the system, it will be subjected to a test route in an urban environment. The route chosen was between the cities of Serra and Vitória in the state of Espírito Santo, as shown in Fig. 10.

The route corresponds to 17.3 kilometers between the cities of *Serra* and *Vitória*. Due to the flow of vehicles, the time spent on the route was 32 minutes and 34 seconds. During this journey, 280 images were captured, of which 51 of these images it was possible to recognize a License Plate through OpenALPR, that is, the software flow shown in Fig 4, reached the Regular Expression Application step. The images have a resolution of 1280 x 720 pixels with an average size of 900KB. In the test run, it was observed that the time taken for the flow described in Fig. 4 to complete ranges from 4 to 7 seconds, that is, the image captures have a maximum sampling time of 7 seconds. Remember that at each execution four images are taken in sequence. Table III shows the number of license plates recognized with the respective speeds at the time they were captured.

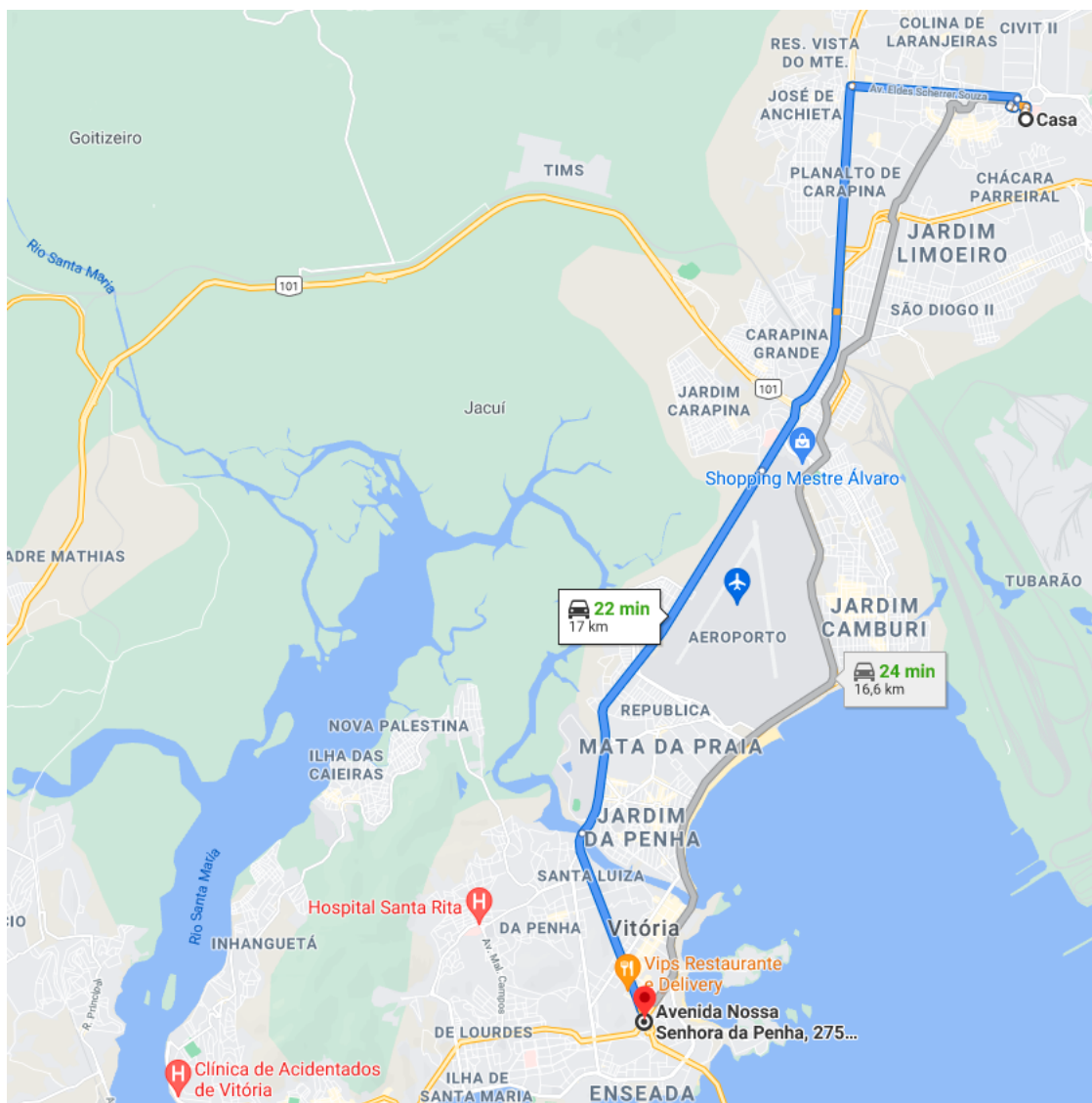


Fig. 10. Testing path (blue path).

TABLE III. SPEED RANGE AT WHICH THE LICENSE PLATES WERE RECOGNIZED.

	0 to 20 km/h	20 to 40 km/h	40 to 60 km/h	Above 60 km/h
number of vehicle license plates recognized	46	4	1	0

The graph in Fig. 11 shows the data presented in table III over time.

Recognition successes are achieved at low speeds. At high speeds, the system has a level of vibration due to irregularities in the highway and also at high speed a space greater than five meters is taken from the vehicle in front for safety reasons. Thus, it becomes more difficult to successfully recognize the license plate image. Further deepening the results presented in Table III and Fig. 11, we have Table IV.

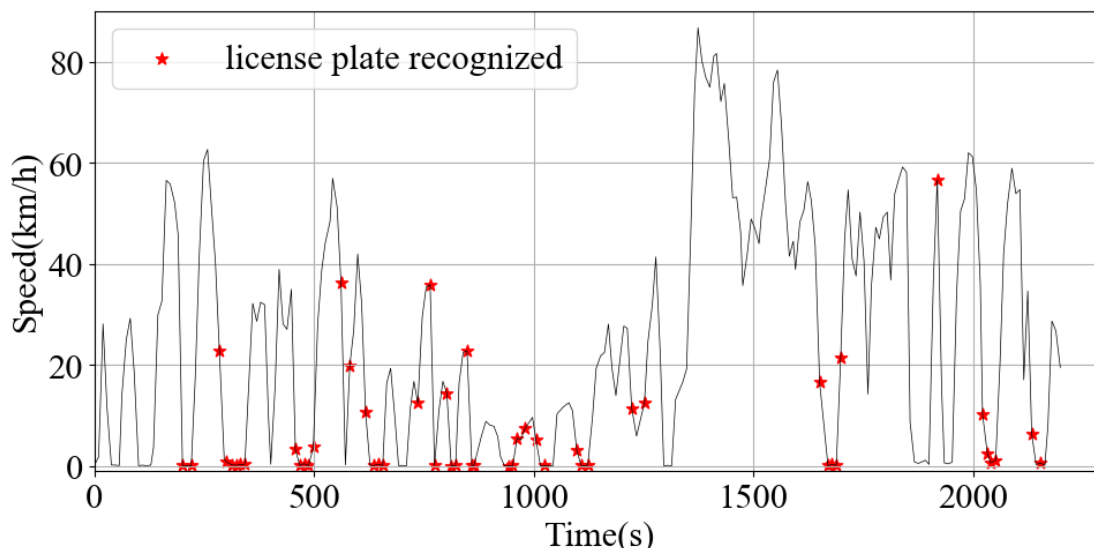


Fig. 11. Speeds measured during plate recognition.

TABLE IV. LICENSE PLATE RECOGNITION SUCCESS RATE.

	New Brazilian License Plate model	Old Brazilian License Plate model
Total license plates collected	29 (56.86%)	22 (43.14%)
Total license plates recognized correctly	27 (93.10%)	20 (90.9%)
Total characters recognized correctly	201 (99.01%)	152 (98.70%)
Total characters incorrectly recognized	2 (0.99%)	2 (1.3%)

In Table IV, it is possible to evaluate the performance of OpenALPR in activity in the system. The table shows that the system can recognize both license plates in the old and the current standard, shown in Fig. 5 and Fig. 6. It is also noticed a higher rate of correctness of characters in license plates in the new standard. This is because, in the old standard, the characters "I" and 1 are represented by the same symbol, which also occurs between the characters "O" and "0". Regular expressions were created to deal with this issue after the recognition stage. Comparing Fig. 5 and Fig. 6 they differ in the third character from right to left, in which on the old license plate, this character is a number, while on the new license plate, this is a letter. In this way, any "1" or "0" that appears in the three leftmost digits will be replaced by "I" or "O" respectively and if the digits "I" or "O" appear in the four rightmost positions, except the third position (if the old and current license plates coincide) is replaced by "1" or "0" respectively. Currently, the system is configured to send three image files (photo of the license plate), along with a text file containing the text of the recognized license plate, latitude, longitude, date, time, vehicle id which captured the license plate and probability of correctness of the same (OpenAlpr emits this probability when recognizing a license plate) every 10 minutes. Fig. 12 shows the volume of the data flow of the test route in the initial 30 minutes.

After transferring files to the remote server and storing information in an MYSQL database, it is possible with authorized credentials to perform license plate searches on the remote server, as shown in Fig. 13.

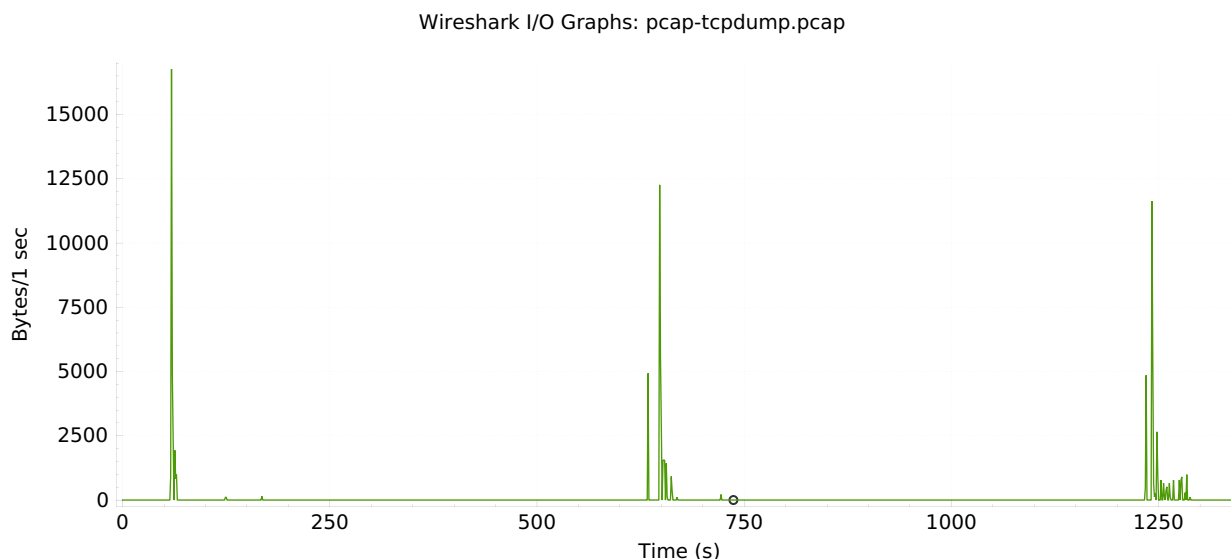


Fig. 12. Flow of sending data from the raspberry pi to the cloud.

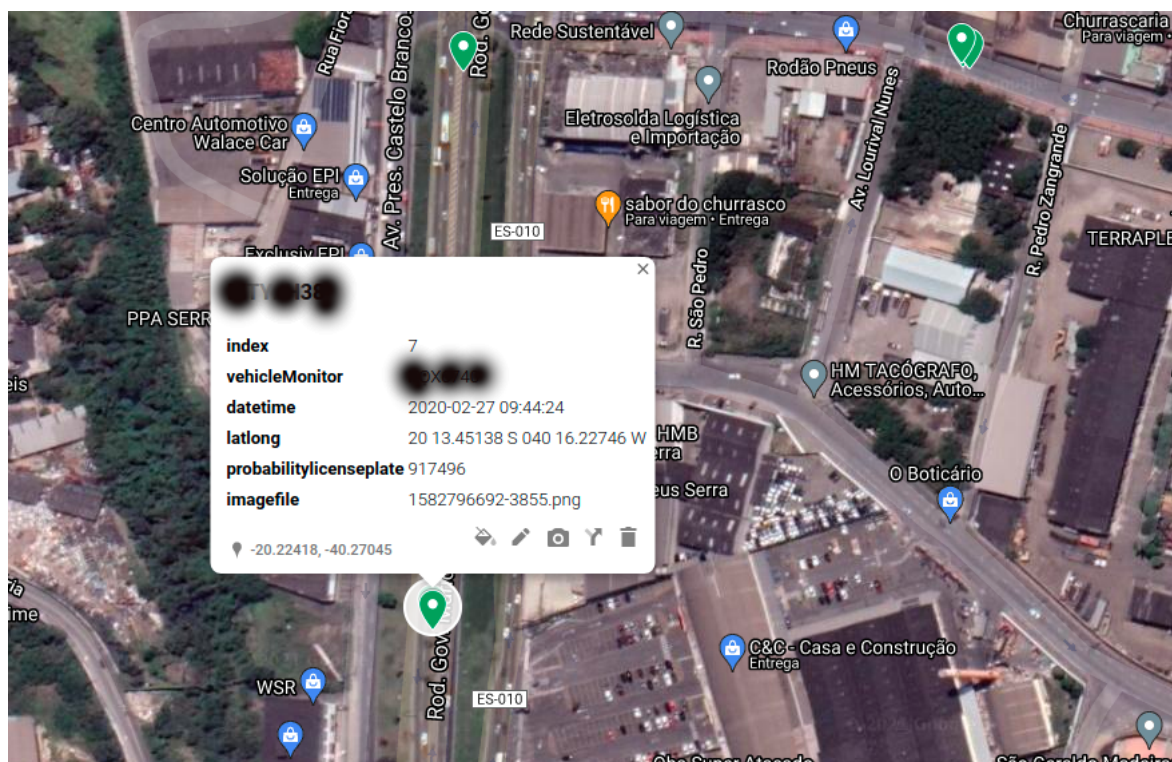


Fig. 13. Result of searching for license plates on the database.

The latitude and longitude positions from which the license plate image was obtained mark a point on the map, which contains the capture information, along with the name of the image file, if any consultation is required. As an example, in Fig. 14, the image file shown in Fig. 13 was opened, which demonstrates the correct recognition of the system (all characters was recognized correctly, only omitted for privacy reasons).



Fig. 14. Vehicle image together with license plate obtained by the system.

VI. CONCLUSION

The implementation of a low-cost IoT system applied in the context of intelligent surveillance was presented in this paper. The results obtained in the practical tests demonstrated the potential of the system to carry out the monitoring of vehicle license plates in an urban area. With a relatively low cost compared to the structures with fixed cameras and computational structure for the analysis and recognition of license plates currently existing, agencies responsible for public security can set up a monitoring network for a value well below centralized computational structures that perform all the work of image processing and recognition in the current scenario, also freeing it up costs of installing high-resolution fixed cameras in strategic locations. In terms of image recognition technology, this system is not improved as in [29], which makes use of sophisticated techniques such as convolutional neural networks and LSTM networks, in addition to the use of state-of-the-art hardware (NVIDIA TITAN GPU with 12GB of GPU memory) from an image processing point of view. This system aims to be scalable, to the point of having so many mobile monitoring points collecting images and performing their recognition that they would overcome the deficiencies of image recognition not seen in sophisticated systems. Another relevant feature from an IoT point of view is the characteristic of local data processing, which optimizes the use of network resources. This feature makes the system able to operate in low data rate networks without overloading it. This system combined with legacy fixed systems for intelligent monitoring can bring capillarity to the structure, providing the monitoring of isolated areas not reached by fixed camera systems. Thus restricted vehicles limited to these areas could then be reached.

ACKNOWLEDGMENTS

We are grateful to LabTel and the Electrical Engineering department at UFES for all their support.

REFERENCES

- [1] *Ministério da Justiça e Segurança Pública. Criminalidade Cai no País em 2019*. (Accessed Mar. 14, 2021). [Online]. Available: <https://www.justica.gov.br/news/collective-nitf-content-1563293956.35>

- [2] P. Swarnalatha, P. Sevugan, T. K. U. Chathurani, R. M. Babu, *et al.*, “Smart sensing network for smart technologies,” in *Applications of Artificial Intelligence for Smart Technology*. IGI Global, 2020, pp. 177–191.
- [3] S. Du, M. Ibrahim, M. Shehata, and W. Badawy, “Automatic license plate recognition (alpr): A state-of-the-art review,” *IEEE Transactions on circuits and systems for video technology*, vol. 23, no. 2, pp. 311–325, 2012.
- [4] A. Hampapur, L. Brown, J. Connell, S. Pankanti, A. Senior, and Y. Tian, “Smart surveillance: applications, technologies and implications,” in *Fourth International Conference on Information, Communications and Signal Processing, 2003 and the Fourth Pacific Rim Conference on Multimedia. Proceedings of the 2003 Joint*, vol. 2, pp. 1133–1138, 2003.
- [5] A. Braicov, I. Budanaev, M. Cosentino, W. Matta, A. Mattiacci, C. M. Medaglia, and M. Petic, “Smart surveillance systems and their applications,” in *EAI International Conference on Smart Cities within SmartCity360° Summit*, pp. 179–187, 2018.
- [6] Y.-l. Tian, L. Brown, A. Hampapur, M. Lu, A. Senior, and C.-f. Shu, “Tbm smart surveillance system (s3): event based video surveillance system with an open and extensible framework,” *Machine Vision and Applications*, vol. 19, no. 5-6, pp. 315–327, 2008.
- [7] Vitória, Prefeitura. *Cerco Inteligente Móvel de Vitória*. (Accessed Jan. 12, 2021). [Online]. Available: <https://www.vit{ó}ria.es.gov.br/noticias/noticia-41646>
- [8] A. F. Santamaria, P. Raimondo, M. Tropea, F. De Rango, and C. Aiello, “An iot surveillance system based on a decentralised architecture,” *Sensors*, vol. 19, no. 6, pp. 1469–, 2019.
- [9] S. Y. Nikouei, R. Xu, Y. Chen, A. Aved, and E. Blasch, “Decentralized smart surveillance through microservices platform,” in *Sensors and Systems for Space Applications XII*, vol. 11017, pp. 110 170K–, 2019.
- [10] Y. Ai, M. Peng, and K. Zhang, “Edge computing technologies for internet of things: a primer,” *Digital Communications and Networks*, vol. 4, no. 2, pp. 77–86, 2018.
- [11] H. B. Dixon Jr, “Cloud computing,” *Judges J.*, vol. 51, pp. 36–, 2012.
- [12] E. Bushhousen, “Cloud computing,” *Journal of Hospital Librarianship*, vol. 11, no. 4, pp. 388–392, 2011.
- [13] A. J. Neto, Z. Zhao, J. J. Rodrigues, H. B. Camboim, and T. Braun, “Fog-based crime-assistance in smart iot transportation system,” *IEEE access*, vol. 6, pp. 11 101–11 111, 2018.
- [14] M. Marjanović, A. Antonić, and I. P. Žarko, “Edge computing architecture for mobile crowdsensing,” *Ieee access*, vol. 6, pp. 10 662–10 674, 2018.
- [15] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, “A survey on mobile edge computing: The communication perspective,” *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.
- [16] A. L. Albertin and R. M. de Moura Albertin, “A internet das coisas irá muito além as coisas,” *GV EXECUTIVO*, vol. 16, no. 2, pp. 12–17, 2017.
- [17] L. Atzori, A. Iera, and G. Morabito, “The internet of things: A survey,” *Computer networks*, vol. 54, no. 15, pp. 2787–2805, 2010.
- [18] U. Stanley, N. Nkordeh, V. M. Olu, and I. Bob-Manuel, “Iot and 5g: The interconnection,” *development*, vol. 1, pp. 2–, 2018.
- [19] A. O. Adebayo, M. S. Chaubey, and L. P. Numbu, “Industry 4.0: The fourth industrial revolution and how it relates to the application of internet of things (iot),” *Journal of Multidisciplinary Engineering Science Studies (JMESS)*, vol. 5, no. 2, 2019.
- [20] J. Shi, J. Wan, H. Yan, and H. Suo, “A survey of cyber-physical systems,” in *2011 international conference on wireless communications and signal processing (WCSP)*, pp. 1–6, 2011.
- [21] S. Nair, N. Somani, A. Grunau, E. Dean-Leon, and A. Knoll, “Image processing units on ultra-low-cost embedded hardware: Algorithmic optimizations for real-time performance,” *Journal of Signal Processing Systems*, vol. 90, no. 6, pp. 913–929, 2018.
- [22] R. Pi. Raspberry pi 3 model b. (Accessed Dez. 03, 2020). [Online]. Available: <https://www.raspberrypi.org>
- [23] I. Rekor Systems. Openalpr. (Accessed Jan. 11, 2020). [Online]. Available: <https://www.openalpr.com>
- [24] O. team. Opencv. [Online]. Available: <https://www.opencv.org>
- [25] R. S. et al. Tesseract-ocr. (Accessed Apr. 11, 2021). [Online]. Available: <https://www.opencv.org>
- [26] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [27] S. Haykin, *Neural networks: a comprehensive foundation*, Macmillan, Ed. Prentice-Hall, Inc., 1994.
- [28] N. Dorbe, A. Jaundalders, R. Kadikis, and K. Nesenbergs, “Fcn and lstm based computer vision system for recognition of vehicle type, license plate number, and registration country,” *Automatic Control and Computer Sciences*, vol. 52, no. 2, pp. 146–154, 2018.
- [29] Y. Zou, Y. Zhang, J. Yan, X. Jiang, T. Huang, H. Fan, and Z. Cui, “A robust license plate recognition model based on bi-lstm,” *IEEE Access*, 2020.

- [30] G.-S. Hsu, J.-C. Chen, and Y.-Z. Chung, "Application-oriented license plate recognition," *IEEE transactions on vehicular technology*, vol. 62, no. 2, pp. 552–561, 2012.
- [31] R. P. Foundation. Raspbian. (Accessed Dez. 03, 2020). [Online]. Available: <https://www.raspberrypi.org>
- [32] D. Jones. Api picamera. (Accessed Dez. 03, 2020). [Online]. Available: https://www.picamera.readthedocs.io/en/release-1.10/api_camera.html
- [33] C.-N. E. Anagnostopoulos, "License plate recognition: A brief tutorial," *IEEE Intelligent transportation systems magazine*, vol. 6, no. 1, pp. 59–67, 2014.
- [34] *Ministério da infraestrutura. Resolução nº 729, de 06 de março de 2018.* (Accessed Dec. 04, 2020). [Online]. Available: <https://www.gov.br/infraestrutura/pt-br/assuntos/transito/conteudo-contran/resolucoes/resolucao7292018consolidada.pdf/view>
- [35] ——. *Resolução nº 241, de 22 de junho de 2007.* (Accessed Dec. 04, 2020). [Online]. Available: https://www.gov.br/infraestrutura/pt-br/assuntos/transito/conteudo-contran/resolucoes/resolucao_contran_241.pdf