A Real-Time Vehicle Identification System Implemented on an Embedded ARM Platform

Danny Sotomayor
Grupo de Investigación en Sistemas Inteligentes (WiCOM-Energy)
Departamento de Eléctrica y Electrónica
Universidad de las Fuerzas Armadas ESPE
Sangolquí, Ecuador
dasotomayor@espe.edu.ec

Milton F. Rosero
Departamento de Eléctrica y Electrónica
Universidad de las Fuerzas Armadas ESPE
Sangolquí, Ecuador
faboxrp@gmail.com

Diego S. Benítez
Colegio de Ciencias e Ingenierías "El Politécnico"
Universidad San Francisco de Quito USFQ
Quito, Ecuador
dbenitez@usfq.edu.ec

Paola León
Grupo de Investigación en Propagación, Control Electrónico y Networking (PROCONET)
Departamento de Eléctrica y Electrónica
Universidad de las Fuerzas Armadas ESPE
Sangolquí, Ecuador
rpleon1@espe.edu.ec

***Abstract*—This paper explores the feasibility of using a low-cost embedded ARM-based system for real-time vehicle recognition and identification through image processing. The main features of the system include: vehicle detection, speed measurement, and vehicle identification by license plate number recognition, the information obtained is then send to a database on a server in a local network. An ODROID-U3 embedded board was used for general processing and control. Image processing algorithms for detection of moving vehicles on a road were implemented and optimized, in order to obtain shorter processing times than existing algorithms. OpenCV libraries were used for the implementation. Finally, an analysis of the processing times required by the algorithms and the error percentages obtained with the system implemented are presented and discussed.**

***Keywords*—vehicle identification, speed determination, ARM, Odroid-U3.**

I. Introduction

Driving at excessive speeds is one of the most common causes of accidents worldwide [1], [2]. In Ecuador, for example, the National Traffic Agency (Agencia Nacional de Transito) (ANT) has identified that speeding is the second cause of road traffic injuries and deaths [3]. Therefore, different vehicle speed measurement techniques have emerged to try to prevent this situation as a countermeasure, since visible traffic measurement equipment has a dissuasive effect [2], among the technologies usually employed are: radar detectors, laser scanners, active sensors e.g pressure-sensitive piezo-electric sensors, light barriers at the side of the road, induction loops using inductive sensors e.g. magneto-resistive sensors installed on the road, and CCTV-based systems [2], [4]

Although, there are several commercially available systems that focus on the detection, counting and classification of vehicles [4], and on the identification of traffic violations such as to detect over the speed Limit infractions on a row, create photographic records of infractions and collect traffic parameters based on artificial vision technology and automatic license plate recognition [5], the cost of such systems, is in some cases prohibitive and therefore low-cost alternatives are necessary. In this paper, we will like to explore the feasibility of implementing such kind of vehicle detection and identification system using a low-cost commercially available ARM based embedded platform, with the aim of being accurate, reliable, portable, reproducible and to have a low power consumption. The total estimated cost of the embedded system proposed in this paper is less than $100 (US dollars).

An automatic system for detecting traffic violations must consist of three main parts: a system for detecting infringements, an identification system to identify offenders, and finally a system for sending this information to a central office for further processing. With the information received from the offender, the corresponding infringement notices, fines or penalties can then be issued. In this work, these tasks will be implemented and are carried out in a single embedded system using digital image processing algorithms which perform in real-time. We will focus on identifying vehicle speed and plate number, and to recognize those that are speeding. If a vehicle is considered an infringer, its information will be stored in a database on a local server for further actions.

The paper is organized as follows: Section II describes the hardware, software and the characteristics of the ODROID-U3 embedded system used. Section III describes in detail the algorithms implemented within the system. In Section IV, the test performed and the results obtained with the system are presented and discussed. Finally we conclude and draw some perspectives.

978-1-5386-3123-2/17/$31.00 ©2017 IEEE

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DE GOIAS. Downloaded on September 25,2024 at 23:32:58 UTC from IEEE Xplore. Restrictions apply.

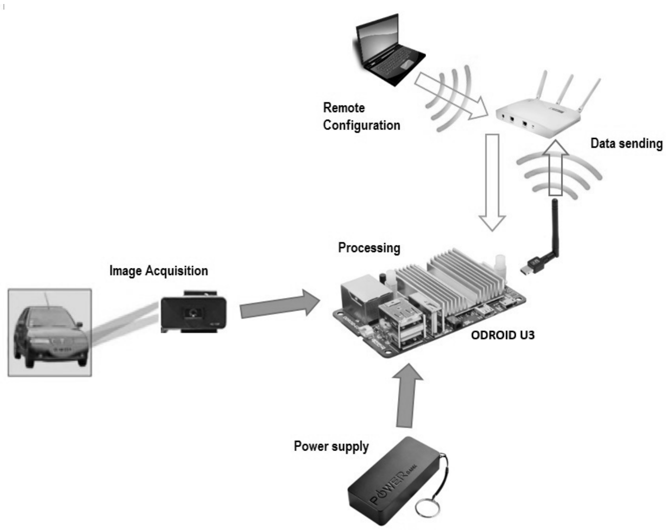

Fig. 1. Schematic representation of the components of the Vehicle identification system.

II. Embedded System

A. Hardware

The system was implemented using a general purpose embedded platform based on the Odriod-U3 board (Hardkernel co., Ltd.) [6], a powerful ARM based computing device with low power consumption and small form factor. It outperforms similar devices such as Arduino, Raspberry Pi, BeagleBone Black, among others, mainly due to its Samsung Exynos4412 Prime Cortex-A9 Quad Core 1.7 GHz ARM processor with 1MB L2 cache and its 2 GB LPDDR2 880 Mega Data Rate RAM, which make the Odriod-U3 a system capable of performing image processing algorithms with low energy consumption thanks to the ARM [7] and Low Power Double Data Rate memory (LPDDR) technology. It uses open-source software for its operation, which allows the system to be reproducible, and its physical dimensions 83 x 48 mm and 48 g of weight increase its portability. The Realtek RTL8188CUS-GR Wireless LAN module (Hardkernel co., Ltd.), an integrated single-chip Wireless LAN (WLAN) USB 2.0 network interface controller, compatible with the with IEEE 802.11b /g/ n specifications, was used for communication allowing to send the information obtained with the system to a server in a local network, as well as to remotely connect to ODROID system for configuration and commissioning. To capture images, the ODROID USB-CAM 720P (Hardkernel co., Ltd.), a CMOS based USB camera with 1.0 Megapixel (1280 x 720 HD resolution), fully compatibility with the ODROID-U3, was used. It supports MPEG and MJPEG (Video) and BMP and JPEG (images) capture formats with speeds up to 30 frames per second (fps). A 5 Volts V/ 10400 mAh battery bank was used to power the system, allowing about six hours of autonomous operation approximately. The schema used is illustrated in Fig. 1.

B. Software

To implement the algorithms for image processing, we used the Open Source Computer Vision Library (OPENCV) [8], optimized in C / C ++ to take advantage of multi-core processing. For optical character recognition (OCR) we used the Tesseract OCR library [9], which allows reading a wide variety of image formats and converting them to text in more than 60 languages. It also allows training to support custom or specific languages.

III. Image Processing Algorithms

The system works as follows, after video acquisition and image capture, the presence of vehicles is identified within the images and the instantaneous speed is determined. Then the vehicle plate is located, if one is found, it is segmented into smaller regions corresponding to the individual characters. The OCR algorithm [9] is then performed and the sequence obtained is effectively validated for a plate number. This information is then sent to a database in a local network. Finally, all the information, including photographs of the vehicle and the detected plate, is displayed on a user interface. The algorithms used are explained next.

A. Vehicle Detection

The ability to extract objects in motion in a video sequence is a fundamental to many vision systems, including video surveillance [11] [10], traffic monitoring [13], detection of people [14], among other applications. The usual approach to discriminate moving objects in the background scene is background subtraction. The idea is to subtract the current image from a reference image, which was acquired from a static background over a period of time. The subtraction results in the silhouette of the region of non-stationary or new objects [12]. This technique has been widely used in many vision systems as a preprocessing step for the detection and tracking of objects. The results obtained by existing algorithms are quite good and many of them work in real time. However, taking into account that the detection of objects is only one part of the algorithm, and that an embedded system will be used to perform the image processing, all the techniques and algorithms used in this work were prioritized to use the shortest processing time possible. For this reason, low complexity techniques were used mainly in the vehicle detection stage (detection of moving objects), and therefore only key parameters (the coordinates of two regions of interest where the analysis will be perform) were established in the image for the analysis, these should be set initially by the user in the user interface in the initial configuration of the system.

After the image was obtained from the video camera, a uniform 3x3 Gaussian filter [15], [16] was applied to the image to eliminate noise and highlight the regions of interest. Prior to vehicle detection, a reference background image corresponding to the road without vehicles in circulation was obtained. For this, the standardized correlation technique [17] was used, 10 consecutive images were compared by

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DE GOIAS. Downloaded on September 25,2024 at 23:32:58 UTC from IEEE Xplore. Restrictions apply.

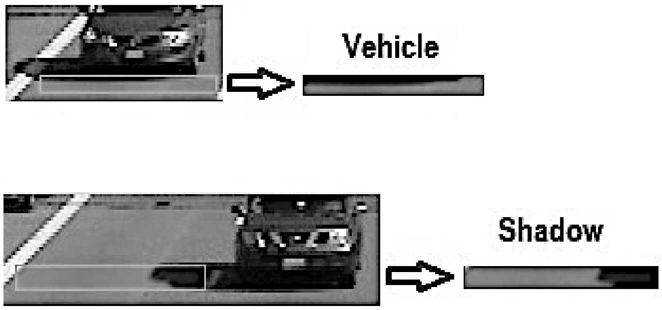

Fig. 2. Image changes in the regions of interest.

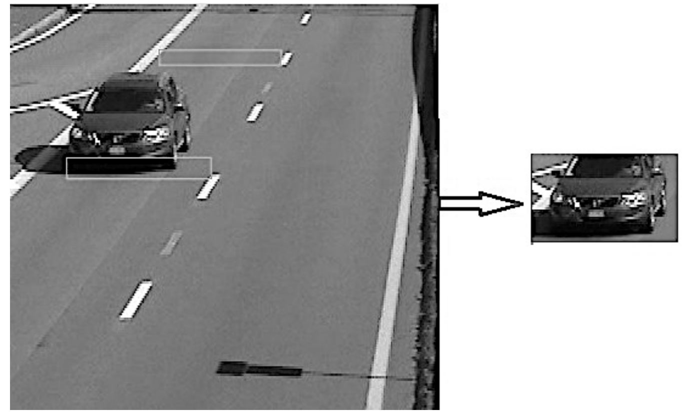

Fig. 3. Image segmentation of the vehicle detected.

Fig. 4. Vertical Edge Detection.

correlation, it was considered that there were not changes in the image if there are not correlation values lower than 0.98. This process is constantly repeated in order to keep the background updated, avoiding false detections due to changes in illumination throughout the day.

Then, the algorithm checks if the region of interest 1 (RI1) has changed with respect to a previous image to determine the presence of an object in that region. A valid change is recorded when the correlation between the background image and the new image has values lower than 0.98 in the region of interest.

The algorithm then analyzes if the object presents in the RI1 corresponds to the patterns of a vehicle. The object in the image is considered as a vehicle when the number of pixels that change in the vertical projection is greater than 0.65 of the total width of the lane (figure 5). Otherwise, it is assumed that the object that does not correspond to a vehicle, it can be a person, bicycle, motorcycle, or the shadow of another vehicle that circulates in the adjacent lane, as illustrated in Fig. 2

When a vehicle is detected, a time counter is started o restarted, an initial time reference is taken to subsequently determine the total time used by the vehicle to pass through the images and with this, to determine the instantaneous speed. This reference is obtained from the number of cycles that the processor registers up to that moment.

After this, the algorithm verifies if the region of interest 2 (RI2) has changed with respect to a previous image, which means the presence of an object in that region. Similarly, a change is recorded when the correlation between the background image and the new image has values lower than 0.98 in the region of interest. A similar analysis as the one previously described for RI1 is then performed in the RI2 to determine if these changes correspond to a vehicle pattern. If the analysis is positive, then a second time reference is taken, which corresponds to the time elapsed since the vehicle has passed through RI1 1 to RI2. The number of cycles recorded by the processor should be greater than the first reference taken. The total traveling time is obtained by subtracting the time reference 1 from the time reference 2, and dividing this value for the operating frequency of the processor. Having previously known the distance between the marks and determined the time required to cross them, the instantaneous velocity of the vehicle can then be calculated.

The region of the image corresponding to the vehicle, i.e the region above the region of interest 2 (Fig. 3), is then trimmed from the image and will be used in the next process for the location of the license plate.

B. License Plate Localization

The resulting segmented image containing the vehicle detected is then further processed, a uniform 3x3 Gaussian filter [15] is first applied to the image to eliminate noise and a 1x3 mean filter [17] is then applied to emphasize vertical regions, and at the same time reduce the small horizontal regions in order to obtain a correct detection of vertical edges. A Sobel filter [15], [16] is then applied to the image in order to detect vertical edges, only values greater than 0.3 times the maximum value are keep in the image, this process highlights the area in the image where the plate is located, as is shown in Fig. 4.

From Fig. 4, a horizontal projection is made using the vertical edges previously located, the peaks that appear in this projection are candidate regions where the plate is probably located. In FIG. 5 there are two candidate rows to be further analyzed to determine which one actually corresponds to the plate.

In the image of the candidate rows a new vertical edges detection is again performed, but using different parameters

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DE GOIAS. Downloaded on September 25,2024 at 23:32:58 UTC from IEEE Xplore. Restrictions apply.

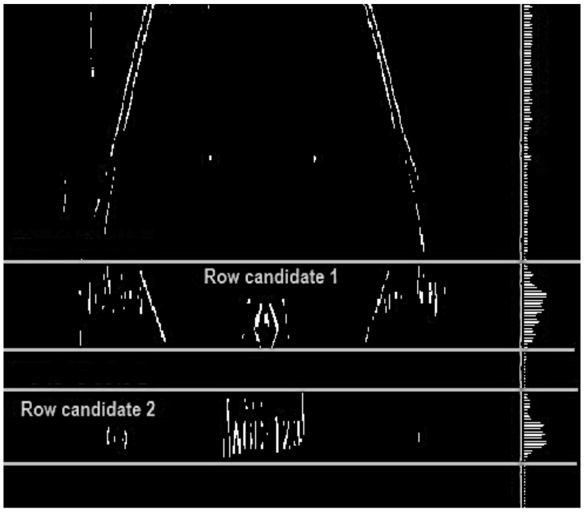

Fig. 5. Horizontal projection made using the vertical edges previously located.

from the first process, since a smaller area is analyzed and it is necessary to emphasize the features that correspond to the characters of the plate. After, a new vertical projection is obtained, a Gaussian filter [15], [16] is then applied to obtain a uniform distribution where the center zone is highlighted, as illustrated in Fig. 6.

Fig. 6. Vertical projection of the candidate row and location of the plate.

This region corresponds to the position of the plate in the candidate row. The image is then converted to binary by thresholding [10], [17], the regions adjacent to the edges are removed, since the characters will appear in the center, this process also removes very small peaks corresponding to the dash symbol and stains that may be present in the plate. The resulting image is shown in Fig. 7

Fig. 7. Resulting Image after thresholding; edges and small peaks are also removed.

Finally, the resulting zone located is evaluated to determine if it has the features of a license plate, first by determining the number of regions it contains which represent the license plate characters, these should be not less than four but not more than seven; then the dimensions should be determined, the width should greater than the height of the plate by not more than 10 times (very elongated area) and not less than 0.75 times (zone tending to square). If the result is positive the process is finished obtaining the image of the plate, if not, the row under analysis is discarded, and if the number of recognition attempts limit has not yet been reached, the process with the next candidate row is repeated, otherwise it is terminated without having found a license plate.

C. License Plate Characters Recognition

The vertical projection of the plate is performed and the peaks and valleys area located, the peaks correspond to the characters, and the spaces or places with little concurrence between them are represented by the valleys, as Illustrated in Fig. 8.

Fig. 8. Vertical projection of the plate, identification of peaks and valleys.

The characteristics of the Ecuadorian license plates allow to separate the letters of the numbers without a first OCR process, as it shown in Fig. 9, in the center of the plate there is a valley that is wider than the others, it corresponds to the dash symbol that was previously eliminated. The left part of this space corresponds only to letters, and the right part relates only to numbers. Making this separation guarantees that in the OCR process there will be not mistakes in the recognition of numbers by letters and vice versa (e.g. the number 8 being recognized as the letter B, or the number 5 recognized as the letter S), reducing therefore the recognition time required and not further validation will be then necessary.

Then the peaks in the vertical projection of the plate are separated and the region containing each one are individually processed, the corresponding OCR for letters or numbers is made and all the identified characters are stored in a single matrix, forming the license plate again. Once all the characters have been processed, the identified text sequence is evaluated to determine if it corresponds to a license plaque, in other words, it should have 3 letters and 3 or 4 numbers. If the result is positive the process is finished, otherwise it ends without a positive identification result.

IV. Results and Discussion

To tests the performance of the system, a portable variable height (1.70 to 4,5 m) metallic pole, with a 1.5 m folding

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DE GOIAS. Downloaded on September 25,2024 at 23:32:58 UTC from IEEE Xplore. Restrictions apply.

Fig. 9. Identification of regions with letters and numbers on the Ecuadorian license plates.

horizontal arm was used. The entire system, i.e. Odroid-U3 processing card, the web camera, the Wi-Fi antenna and the 5V battery, was placed inside a metallic box and suspended from the arm at a 4.50 m height in order to obtain a top view of a road with vehicles in circulation and to facilitate the recognition of movement along the road minimizing possible interference with the image. Given the characteristics of our system, a parking road with low speed limit within the university campus was chosen for our application, since the aim of the system is to detect speeding offenders.

For the vehicle detection stage, 200 out of 200 vehicles circulating through the detection zone were correctly detected by the system, obtaining therefore a 0% error in detection. Furthermore, there were 2 bicycles and 6 people that crossed the road and were not registered by the system, verifying that the vehicle identification stage works correctly. A vehicle circulating at constant speeds of 30, 40 and 50 km/h through the detection zone, was used to test the speed measurement performed by the system, unfortunately we did not have a radar system to accurate measure the vehicle speed, however, this test provides an estimate about the performance of the algorithm. The results obtained that are shown in Table I.

TABLE I
RESULTS OF THE VEHICLE SPEED MEASUREMENT PROCESS

Known Speed (km/h)	Measured Speed (km/h)	Error (km/h)	Error (%)
30	34.6	4.6	15.33
40	44.1	4.1	10.25
50	55.7	5.7	11.4

Given the limitations of the low-cost web camera used, not clear images of the area of the plate for vehicles in motion were obtained from 4.50 m height, therefore plate identification was not possible from this height, this is a limitation of the system that should be corrected by replacing the camera. To test the license plate identification performance, the camera was placed at 1.70 m a height, where clear frontal images of the vehicle plate were obtained. 200 front images from vehicles at different speeds were analyzed taking into account the 3 main stages of this process: (1) the identification of the zone of the plate, (2) the segmentation of plate characters and (3) the optical character recognition (OCR), the results obtained are shown in Table II.

As it can be seen from the Table II, the number of images analyzed in each process is not the same in all cases since they depend on the effectiveness of the previous process, i.e. if the identification in one stage is not satisfactorily fulfilled, the next stage one will not be continued. In stage (1), there system always identifies the area of the image corresponding to the vehicle plate, therefore there was not error in this stage. In stage (2), the error obtained for character segmentation was 5%, this was mainly due to vehicles with bad, discolored, or bent license plates or when parts of the same vehicle obstructed the readability of the license plate in the image. At this time, the recognition system does not have morphological or region restoration algorithms implemented within the system and therefore the algorithm to do not work effectively for such cases. If only plates in good condition are taken into account, the percentage of correct character segmentation would be 100%.

TABLE II
RESULTS OF LICENSE PLATE RECOGNITION PROCESSES

Stage	Number of Images Analyzed	True Detection Output	Stage Error (%)	Total Error (%)	True Detections per stage (%)
(1)	200	200	0	0	100
(2)	200	190	5	5	95
(3)	190	172	9.47	14	90.53

Stage (3) in the other hand, presents an error of 9.47%, this is because there are different types of letters used for the characters of the license plates in Ecuador, and it was not possible to perform a custom language training for the Tesseract OCR algorithm, for security reasons the ANT does not disclose what types of letters are used for license plates. Nevertheless, a training was carried out using 15 samples for each character from previously obtained images of correctly identified license plates from detected vehicles, a total of 540 images corresponding to 10 numbers and 26 letters were used for training. However, it cannot be asserted that within these samples all types of letters used for the license plates are covered. This percentage of error can be reduced with proper training of the custom language, or if the right letter types used in the plates are known.

A. Processing time required by the algorithms implemented

The time required by the embedded system for the execution of the vehicle detection process was analyzed using images with different size resolution. The average results obtained from this test are shown in Table III. As it can be seen from this Table, there is almost a linear relationship between the size of the image and the time required for its processing, which increased about the double and triple, as the size of the images increases.

Similarly, the times required by the embedded system for vehicle Identification stages were also analyzed using images of different size resolution, larger images acquired with cameras other than the one used by the system were also used, in order to evaluate the algorithm developed. The results obtained are shown in Table IV. The graphs of Fig. 10 shows that while the time required for the character segmentation and OCR

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DE GOIAS. Downloaded on September 25,2024 at 23:32:58 UTC from IEEE Xplore. Restrictions apply.

TABLE III
TIME REQUIRED FOR THE VEHICLE DETECTION PROCESS

Image Resolution (pixels)	Time (ms)
320x240	12
640x480	25
1280x720	34

stages are similar for images with different size resolution, the time required for the license plate localization stage increases almost linearly as the image resolution increases. Therefore, we can conclude that the time required for vehicle identification depends mainly on the location of the plate, which in turn also depends on the size of the image.

TABLE IV
PROCESSING TIME REQUIRED BY THE VEHICLE IDENTIFICATION STAGES

	Image Resolution (pixels)					
	259x194		640x480		3264x2448	
Stage	Time (ms)	%	Time (ms)	%	Time (ms)	%
License Plate Localization	5	22.72	38	64.4	120	89.55
Character Segmentation	2	9.09	5	8.47	3	2.24
OCR	15	68.19	16	27.13	11	8.21
Total	22	100	59	100	134	100

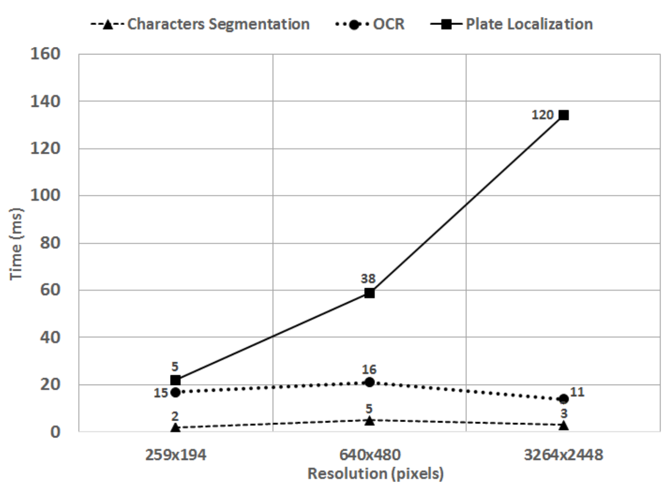

Fig. 10. Comparison between image size and processing time required for the vehicle identification stages.

Considering vehicle detection as the main process, since it is the one that is executed independently of the presence of vehicles on the road, it was determined in the previous section that in the Odroid-U3 it requires a maximum of 34 ms for 1280x720 pixel resolution images, which results in about 29 images being processed per second, this is the largest image size that the camera used allows to capture. After a vehicle is detected, the time for vehicle identification (96 ms) and the time for sending this information to the database (about 110 ms) must be added. In total, the entire process requires about 240 ms, and therefore up to 4 vehicles per second can be detected with the system. Estimating that in practice vehicles may circulate with a difference of about one second, there would be 760 ms in which the vehicle detection would continue to be performed, obtaining an average of 23 processed images per second.

Performing the same analysis for 320x240 pixel resolution images yields a total processing time of 149 ms, with a maximum of 6 vehicles that can be detected per second, with an average of 71 images per second. For medium-sized images with 640x480 pixel resolution, a total time of 194 ms is obtained, with a maximum of 5 vehicles that could be detected, with an average of 33 images per second.

It is important to note that the number of images processed per second is directly limited by the resolution of the camera used, which has a maximum capture capacity of 30 images per second. In addition, under low ambient conditions, only up to 12 images per second can be obtained. Therefore, in order to improve vehicle detection and identification performance, is recommended to better use a specialized camera for vehicular recognition applications with faster image capture rates, such as the KOMOTO Number Plate ReadCam AVN-80RL25/AVP-80RL25 [18].

V. CONCLUSION

In this work, an ARM based embedded system for vehicle detection and license plate identification of vehicles was proposed. Experimentation results indicate that the optimized algorithms proposed for the embedded solution are effective. A 100% success rate for vehicular detection and 86% for license plate identification were achieved, allowing also to differentiate vehicles from bicycles and people. Although, the quality of images obtained by the low-cost web camera initially used with the system might not be enough for the complex task of license plate identification, the results obtained demonstrated that, without taking into account the limitations of the camera, it is feasible to carry out vehicular detection at low speeds and license plate identification by processing of images with the Odroid-U3 based embedded system.

Further work will increase the quality of images by using a specialized camera for vehicular recognition applications, and extend the training library for the OCR algorithm.

REFERENCES

[1] B.R. Sharma, “Road Traffic Injuries: A Major Global Public Health Crisis,” Public Health, Vol. 122 (12), pp. 1399-1406, 2008.
[2] Federal Roads Office (FEDRO), “Traffic Measurement Technology,” Federal Institute of Metrology METAS, 2013.
[3] ANT, “Probable Cause of Nationwide Traffic Accidents (Siniestros por causas probables a nivel nacional),” Report, 2014, available at http://www.ant.gob.ec.
[4] L.E.Y. Mimbela, L.A. Klein, “A Summary of Vehicle Detection and Surveillance Technologies used in Intelligent Transportation Systems,” U.S. Department of Transportation, Federal Highway Administration, 2007.
[5] C. Parsons, J. Savirimuthu, R. Wipond, K. McArthur, “ANPR: Code and Rhetorics of Compliance,” European Journal of Law and Technology, Vol. 3 (3), 2012.
[6] Hardkernell, “Odroid u3 technical detail,” HARDKERNEL, Tech. Rep., 2014.

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DE GOIAS. Downloaded on September 25,2024 at 23:32:58 UTC from IEEE Xplore. Restrictions apply.

[7] D. Abdurachmanov, P. Elmer, G. Eulisse, S. Muzaffar, “Initial explorations of ARM processors for scientific computing, ” Journal of Physics: Conference Series 523, 2014.
[8] G. Bradski, “Opencv library,” Dr. Dobb’s Journal of Software Tools, 2000, available at http://opencv.org/.
[9] R. Smith, “An overview of Tesseract OCR engine,” Ninth International Conference on Document Analysis and Recognition (ICDAR), pp. 629-633, 2007.
[10] P.L. Rosin. “Thresholding for Change Detection,” in Proceedings of the Sixth International Conference on Computer Vision (ICCV ’98), IEEE Computer Society, Washington, DC, USA, pp. 274-, 1998.
[11] I. Haritaoglu, D. Harwood, and L. Davis, “W4: Who? when? where? what? a real time system for detecting and tracking people,” in Proceedings of the Third International Conference on Automatic Face and Gesture Recognition (FG98), pp. 222-227, 1998.
[12] T.B. Moeslund, A. Hilton, V. Krüger, “Survey of advances in vision-based human motion capture and analysis,” Computer Vision and Image Understanding, Vol. 104, (23), pp. 90126, 2006.
[13] N. Friedman and S. Russell, “Image segmentation in video sequences: A probabilistic approach,” in Proceedings of the 13th Conference on Uncertainty in Artificial intelligence, UAI’97, pp. 175-181, 1997.
[14] J. Davis and A. Bobick, “The representation and recognition of action using temporal templates,” in Proceedings of Conference on Computer Vision and Pattern Recognition, 1997.
[15] M. Sonka, V. Hlavac, R. Boyle, “Image Processing, Analysis, and Machine Vision,” 4th Edition, CL Engineering, 2014.
[16] W. Burger, M.J. Burge, “Digital Image Processing: An Algorithmic Introduction using Java,” Springer London Ltd, 2012.
[17] R. Gonzalez, Digital Image Processing Using Matlab”, Prentice Hall,New Jersey, 2002.
[18] KOMOTO, “Number plate readcam avn-80rl25/avp-80rl25,” KOMOTO ENTERPRISE CO., LTD., Tech. Rep., 2010.

Authorized licensed use limited to: UNIVERSIDADE FEDERAL DE GOIAS. Downloaded on September 25,2024 at 23:32:58 UTC from IEEE Xplore. Restrictions apply.