

# Vehicle Classification Through Detection and Color Segmentation of Registration Plates Running on Raspberry Pi 3 Model B

Louielito Ferrolino, Allysa Kate Brillantes, Melvin Cabatuan, John Anthony Jose, Elmer Dadios

*Abstract- Classification of license plates provides useful information regarding the nature of the vehicle, whether it is used for public transport, a privately-owned vehicle, an official vehicle, or a special vehicle. In the Philippines, the registration plates of vehicles are classified by colors. Colors such as red, green blue, black, yellow are used to identify what vehicle classification the plate belongs to. The information is useful to applications in statistics and transport regulation. This paper discusses a convolutional neural network based embedded system that runs on Raspberry Pi 3 Model B. The said system provides a process of classifying vehicles using plate detection using Convolutional Neural Networks and color thresholding of registration plates using the RGB color space. TensorFlow and OpenCV libraries were utilized for the detection and classification.*

*Index Terms: Convolutional Neural Network, License plate detection, Raspberry Pi 3, vehicle classification.*

## I. INTRODUCTION

Naming and color schemes are implemented to registration plates in the Philippines [1]. The old naming scheme of license plates is in the form LLL-DDD, where L stands for a letter and D stands for a digit. The color scheme for public utility vehicles is black characters with a yellow background (or vice-versa), green and white for private vehicles, red and white for government vehicles and blue and white for diplomatic vehicles. In 2013, the Land Transportation Office spearheaded the Plate Standardization Project, implementing a new format and design [2]. The new format includes a naming scheme LLL-DDDD. The color scheme is also changed for the private vehicles, making it black characters on white background as contrast to the green characters (or background). However, the old naming scheme is still existent and implemented. This paper discusses an embedded system running on Raspberry Pi 3's Raspbian Stretch OS. The embedded system contains algorithms on registration plate detection using TensorFlow and color

detection based on thresholding the color values of yellow, green, blue, red, and black on the BGR color space utilizing

**Revised Manuscript Received on August 19, 2019.**

**Louielito Ferrolino**, Electronics and Communications Engineering, De La Salle University, Phillipness

**Allysa Kate Brillantes**, Electronics Engineering Technological University, Phillipness

**Melvin Cabatuan**, Electronics and Communications Engineering, Cebu Institute of Technology, Cebu, Philippines

**John Anthony Jose**, faculty in the Electronics and Communications Engineering (ECE), IEEE Philippines.

**Sabrina Ahmad**, Electrical and Electronics Engineers, the founder and chair of IEEE Computational Intelligence Society, Phils

OpenCV library.



Fig. 1. Sample registration plates from private vehicles



Fig. 2. Sample registration plates from public utility vehicles



Fig. 3. Sample registration plates from government vehicles



Fig. 4. Sample diplomat plate

Raspberry Pi 3

The Raspberry Pi 3 Model B, also called RPi3, is a small single-board computer that is used in a wide variety of applications. Such applications include image processing and robotics. The use of the Raspbian Stretch OS provided better scalability and adequate support since this is an open source and is based on Linux. This has been used to provide a cost-effective approach in this study. A brief table of specifications for the Raspberry Pi 3 Model B is illustrated below.

SoC	Broadcom BCM 2387
CPU	1.2 GHz quad-core ARM Cortex A53
Instruction Set	ARMv8-A
GPU	400 MHz Video Core IV
RAM	1 GB LPDDR2-900 SDRAM
Storage	Micro-SD
Ethernet	10/100
Wireless	802.11n / Bluetooth 4.0
Video /Output	HDMI / Composite
Audio Output	HDMI / Headphone
GPIO	40

**Table 1. Specifications of Raspberry Pi 3 Model B**

**II. RELATED STUDIES**

*Plate Detection*

Various techniques in plate detection have already been studied. Two fuzzy logic systems have been used to segment the license plates and extract its area [3,4]. The studies yielded an 85% and 94.84% correctness, respectively. Other Artificial Neural Network (ANN)-based techniques were used to detect the registration plate area and character segmentation [5, 6]. A comparative study of ANNs and SVMs were conducted and results showed that the majority of the experiments are in favor of ANNs [7]. A subset on ANNs is Convolutional Neural Networks (CNNs). CNNs class of deep, feed-forward (not recurrent) artificial neural networks that are applied to analyze visual imagery [8]. CNNs are generally important in the field of image processing, particularly in object detection [9]. This study concludes that CNNs are preferred in studies of moving object detection and tracking. Moreover, this study compared different frameworks, models and deep learning techniques. The paper also enunciated the importance of Graphical Processing Units (GPUs) in the training and testing phases of object detection. CNNs are particularly well suited to hardware implementations because of their regular structure and their low memory requirements for the weights [10]. Therefore, with this conclusion, CNN will be used in this study.

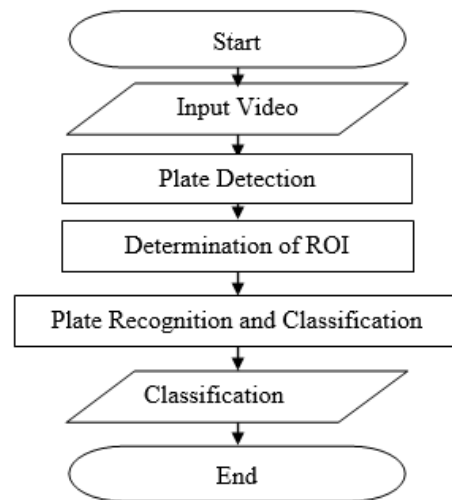
In recent years, there has been a great interest in developing efficient CNN-based image detection methods. Some of the most utilized CNN-based systems are Region-based Convolutional Neural Networks(R-CNN), Region-based Fully Convolutional Networks(R-FCN), You-Only-Look-Once (YOLO), SingleShot Multibox Detector (SSD) and MobileNets. The original R-CNN proposes various locations, runs a neural network classifier on scaled test images, refines and rescore the boxes based on other objects the scene. Faster R-CNN and R-CNN systems both improved the RCNN method and reduces the computation of region proposal [11, 12]. Unlike the previous models, YOLO uses single neural network to predict bounding boxes and class probabilities directly from full images in one single run. The YOLO model is the first framework that processes images in real-time at 45 frames per second but has drawback in detecting small objects [12]. Like YOLO, SSD also uses a single deep neural network but it has better accuracy with a smaller input image size [13]. MobileNet bested models in terms of computational speed [14].

*Color Detection*

Color detection has a wide variety of uses. In image processing, this process is the first stage of various applications. The accuracy of detecting colors correctly plays an integral part of most applications in image and video processing [15,16]. OpenCV offers different color spaces for use in color segmentation [16,17,18]. This paper focuses on the RGB color space. An application of color detection has been studied by Yu et al [17,18]. The study involves the detection of traffic light during daytime and night time conditions using RGB color space. The experiment had favorable results.

**III. METHODOLOGY**

The system consists of three main components: Detection, Determination of the Region of Interest (ROI) and Classification. All license plates will be detected using TensorFlow, the regions of interest will be cropped and then be classified using different algorithms. A simple flowchart of the entire system is illustrated below.



**Fig 5. System Flowchart**

*Plate Detection*

The first stage is the detection of the registration plates. TensorFlow Object Detection API was used because it offers open source framework with high accuracy pre-trained models. The model used was SSD with MobileNet, where the meta architecture is SSD and the feature extractor type is MobileNet [13]. Based on experiments conducted by Howard et al[14], MobileNet showed strong performance compared to other popular models on ImageNet classifications with only fraction of computational complexity and model size. Due to computational constraints of RPi3, MobileNet is the best model to work with. The 36-second traffic video to be experimented was is an excerpt taken from a Closed Circuit Television (CCTV) camera located in Epifanio Delos Santos Avenue. It is a recorded video from the CCTV camera whose resolution is 720p at 60 frames per second. It was resized to 640 by 480 pixels at 25 frames per second for the experiment. Due to its low resolution, the corresponding areas of the registration plates are pixelated. The training and testing data consisted mostly of Philippine license plates images and few

frames from different traffic surveillance videos. A total of 200 images were used. Ninety percent of the data collected was used as training sets and ten percent is used as test set.

The images being were trained using the SSD-with-MobileNet Model with 3,000 steps. The trained model was tested using the plate detection algorithm. It is expected that the features of the license plate are extracted from the video manifested by a green bounding box. The first two hundred frames from the resulting video were extracted to test the precision and recall of the algorithm.



**Fig. 6. Plate Detection (from left to right: one bounding box detected-true positive, two bounding boxes detected-false positive, no bounding box detected-false negative).**

*Region of Interest*

After the registration plates are detected and the features are extracted, they will be localized from the vehicle. The region of interest will come from the frames extracted from the detection algorithm. This procedure uses the image processing capability of the OpenCV library where the image is represented as a two-dimensional matrix of M rows by N columns. A pixel on one frame is defined by x and y, where x represents the horizontal coordinate and y represents the vertical coordinate of the pixel. Moreover, since the video is resized to 640 by 480 pixels, then each frame is represented by 480 rows and 640 columns. The upper left-most corner of the frame represents the first row and first column while the lower right-most represents the 480th row and the 640th column of the matrix. The license plate will only be processed to the classification algorithm if it satisfies two conditions. First, it will be processed if the frame has a bounding box in it. The coordinates of the four corners of the bounding boxes are acquired. Second, if the all the four corners of the bounding box are enclosed in the lower half area of the video, which means all the y-coordinates of all the ordered pairs of each corner of the bounded box is in the range of 241 to 480. This is to ensure that the region of interest is maximized and clear before it goes to the next phase. The figure below illustrates how the plate areas are extracted. The yellow shaded area is where the process of ROI determination takes place.



**Fig. 7. Original Frame**



**Fig. 8. Captured Region of Interest**

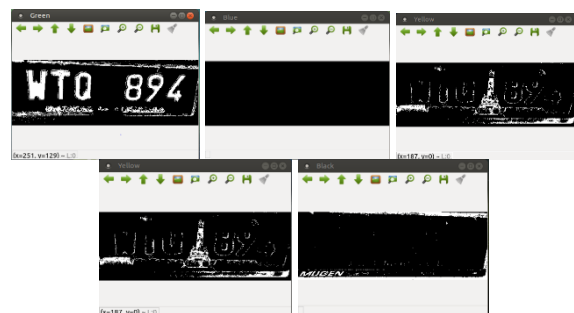
*Plate Classification*

The images will be processed using the thresholding algorithm of the RGB color space in the OpenCV library. With this, it is assumed that a certain BGR component is much larger than other color components on the RGB color space. A certain range of values are applied to mark color thresholds for Red, Blue, Yellow, Green and Black colors. The algorithm will then process the color detection by determining the highest value of white pixels in the image based on the different thresholds of color defined by a certain range in the BGR color space using binarization. For example, if the count of white pixels taken from the yellow threshold is greater than the pixel count of the thresholds of the colors red, blue, green and black, the registration plate will be defined as a plate that is belonging to a public utility vehicle.

Prior integrating the classification algorithm to the detection algorithm, it has been tested for accuracy. The data set came from the training set of the plate detection algorithm. It consists of 137 from private vehicles, 52 from public utility vehicles, 10 from government vehicles, and 1 diplomatic plate. The figures below show an example of color thresholding to classify the vehicle. The plate being tested is WTQ-894. The image was taken during night time. The threshold for yellow, black, red, green and blue were calculated by getting the number of white pixels which came from the range of RGB values being set.



**Fig. 9. Original Test Image**



**Fig. 10. Different Color Thresholds. First Row (left to right): Green, Blue, Yellow. Second Row (left to right): Red, Black**

After testing the algorithm on sample license plates, it was tested on the detected images. The detected images are blurry and have a lot of noise. However, the algorithm was able to identify the colors accurately. Below is a sample image tested based on a frame extracted.

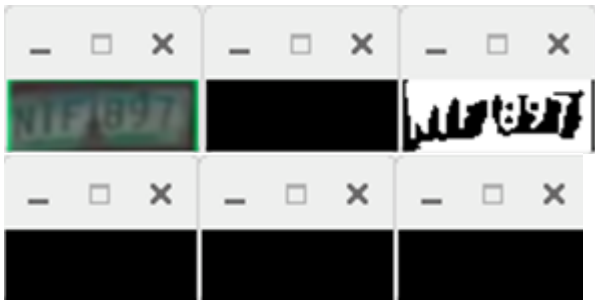


Fig. 11. Plate Classification Test on one frame extracted. (from left to right: original cropped image, yellow threshold, green threshold, black threshold, red threshold, blue threshold.)

IV. DISCUSSION AND RESULTS

Plate Detection

The results were reviewed manually by going through each frame and counting the true positives, false negatives and false positives. One license plate can have only one bounding box associated with it. If several bounding boxes are predicted for the plate, only one is considered True Positive and the others are False Positive. A fully-visible plate without any predicted bounding boxes associated with it is considered a False Negative. The application is evaluated using the following evaluation metrics: precision, recall and F1-score.

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 - score = 2 * \frac{Precision \times Recall}{Precision + Recall}$$

Upon reviewing 200 video frames from the first training experiment, there are 222 true positives, 39 false negatives and 140 false positives. The detection’s precision is 61.8%, recall is 85.05% and F1-score is 71.6%.

		PREDICTED	
		No	Yes
ACTUAL	No	TN = 0	FP = 140
	Yes	FN = 39	TP = 222

Table 1. Confusion Matrix for Plate Detection

To test if the model’s metrics can be increased, 100 frames with license plates that came from the video were added to the training data. The model was retrained using the new training data with 5,000 steps. This approach had 256 true positive, 20 false negatives and 37 false positives. The calculated precision is 87.37%, recall is 89.51% and F1-score

is 88.42%. The false positives were eliminated by inputting still images in the model. This approach detected more plates that are not fully visible.

		PREDICTED	
		No	Yes
ACTUAL	No	TN = 0	FP = 37
	Yes	FN = 30	TP = 256

Table 3. Confusion Matrix for Plate Detection with additional training

Comparing the two experiments show that fine-tuning the model with more training data and more training steps yield more desirable results.

Region of Interest

The data from this phase is purely dependent on the output of the detection algorithm. The images were selected by determining the four corners of the bounding boxes that went through the lower half of each frame. Out of the 200 hundred frames, which represent the first 8 seconds of the video, there were fourteen unique license plates being detected. Eleven plates came from private vehicles and three plates from public transport vehicles. There is a noticeable repetition of car plates since extraction were done by subsequent frames.

Plate Classification

A confusion matrix was utilized to test the accuracy of the algorithm involving private vehicles. The procedure used the images from the original training and testing data set. It is calculated by getting the number of true positives and true negatives divided by the total number of images in the data set. True positives are those plates whose predicted and actual green and black thresholds are greatest. True negatives are those plates whose actual and predicted thresholds are blue, red and yellow.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

The results showed a total of 122 true positives, 15 false positives, 49 true negatives and 14 false negatives, giving an accuracy of 85.5%. The false positives and false negatives are those plates whose images are either blurry or shot in low-light conditions.

Total Tested Plates: 200		PREDICTED	
		No	Yes
ACTUAL	No	TN = 49	FP = 15
	Yes	FN = 14	TP = 122

Table 4. Confusion Matrix for Plate Classification Testing using Images from the Training Set

Out of the fourteen unique license plate plates detected from the frames, 12 plates were classified correctly, giving the algorithm an accuracy of 85.71%. This is because two of the detected images were only temporary license plates. The background of the license is not compliant to the standard

color scheme set by the government. This signifies one limitation of classifying vehicles using colors of the license plates.

## V. CONCLUSION AND RECOMMENDATION

Using Raspberry Pi 3 over other proprietary counterparts gives it an edge when it comes to cost-effectiveness. Since this study utilized an open-source operating system, support will not be a problem. Also, the compact design of Raspberry Pi 3 makes it more efficient and provides better scalability. The approach of classifying a vehicle using the color of its registration plate is useful in statistics, especially in the applications on traffic laws and road safety. This provides a basic overview of what types of vehicles are concerned in a certain road situation. However, it has its own limitations. Temporary license plates might not be able to be classified correctly. Moreover, since the study is only based on the colors of license plates, it would only give an overview of the classification of a certain vehicle. Determination of the characters of the registration plates also provide more information of the nature of a certain vehicle, such as the place where the vehicle was registered. The fine-tuned SSD with MobileNet model worked reasonably well in detecting license plates in this study. Training with more data and iterations will make a big difference in the results. The license plate detection is more precise if still images were inputted to the architecture. Determining the region of interest is also a major concern in determining plate classification. The decision on when to get the region of interest should be strategically defined to get the image of the best quality for the next step. The decision of getting the region of interest on the lower part of the frames had been proven successful since the images are more readable and colors are more accurate on that area. Also, good quantitative metrics for color thresholding provide useful indication and high accuracy for registration plate classification. Correct ranges of the thresholds provided better and more accurate results.

In future studies, a model with another architecture could be used to detect the license plate area. A challenge of this study is that the output image from the detection algorithm is relatively small in pixel count. This results to blurred images that might have affected the classification result. It is also therefore recommended to use high resolution and a high-speed camera in future studies to capture the frames with minimum distortion of the image. Also, pre-processing techniques, such as image enhancements, might be able to help attain more accurate plate detection and classification. The detected images can also be used to recognize the characters of the license plate, hence making it a complete embedded system for Philippine Vehicle Plate Recognition.

## REFERENCES

1. Philippine Government: An act to compile the laws relative to land transportation and traffic rules, to create a land transportation commission and for other purposes (1964). Information on <http://www.lto.gov.ph/issuances/memorandum-circular/file/38-memorandum-circular-no-vpt-2013-1772-motor-vehicle-license-plates-standardization-program.html>
3. D. Costa, C. Mello, A complete system for vehicle license plate recognition, 2009 16th International Conference on Systems, Signals and Image Processing (2009) 3-6.

4. Y. Syed, M. Sarfraz, Color edge enhancement based Fuzzy segmentation of license plates. Proceedings of the International Conference on Information Visualisation (2005) 227-232.
5. B. Pechiammal, J. Arokia Renjith, An efficient approach for automatic license plate recognition system.
6. E. Volna, M., Kotyrba. Vision system for license plate recognition based on neural networks. International Conference on Hybrid Intelligent Systems (2013) 140-143.
7. M. Phangtriasu, J. Harefa, D. Tanoto, Comparison between Neural Network and Support Vector Machine in Optical Character Recognition. Procedia Computer Science, (2017) 116, 351-357. <https://doi.org/10.1016/j.procs.2017.10.061>
8. Information on <https://davidstutz.de/wordpress/wp-content/uploads/2014/07/seminar.pdf>
9. A. R. Pathak, M. Pandey, S. Rautaray, S. Application of Deep Learning for Object Detection. Procedia Computer Science, (2018) 132(Iccids), 1706-1717.
10. Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, Gradient Based Learning Applied to Document Recognition. Proceedings of the IEEE (1998) 86. <https://doi.org/10.1109/5.726791>
11. P. Mustamo, Object detection in sports: TensorFlow Object Detection API case study, (2018).
12. Y. Zhang, H. Peng, P. Hu, CS 341 Final Report: Towards Real-time Detection and Camera Triggering, <http://cs231n.stanford.edu/reports/2017/pdfs/808.pdf>, last accessed 2018-04-08.
13. J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You Only Look Once: Unified, Real-Time Object Detection 2016 IEEE Conference on Computer Vision and Pattern Recognition (2016) 779-788
14. A.G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, ... H. Adam, MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications (2017).
15. K. Pulli, A. Basheev, K. Korniyakov, V. Eruhimov, Realtime computer vision with OpenCV. Queue, 2012, pp 40-56.
16. Information on <https://www.learnopencv.com/color-spaces-in-opencv-cpp-python>
17. C. Yu, H. Chuan, L. Yao, Traffic light detection during day and night conditions by a camera, IEEE 10th International Conference on Signal Processing Proceedings (2010) 821-824.
18. J. Binangkit, D. Widayatoro, Increasing accuracy of traffic light color detection and recognition using machine learning

## AUTHORS PROFILE



**Louielito U. Ferrolino** has received his BS Electronics and Communications Engineering in Pamantasan ng Lungsod ng Maynila (University of the City of Manila). He is currently taking up his Master of Science in Electronics and Communications Engineering in De La Salle University. His current research work is inclined to edge computing.

# VEHICLE CLASSIFICATION THROUGH DETECTION AND COLOR SEGMENTATION OF REGISTRATION PLATES RUNNING ON RASPBERRY PI 3 MODEL B

computer vision, machine learning and integration of IoT.



**Allysya Kate Brillantes** has done BS in Electronics Engineering from Technological University of the Philippines and taking her master's degree in Electronics Engineering in De La Salle University. Her research interests are in image processing, computer vision, machine learning, machine learning and intelligent transport system.



**Melvin K. Cabatuan** received the B.Sc. degree in Electronics and Communications Engineering (ECE) from Cebu Institute of Technology, Cebu, Philippines, in 2004; MS degree in Engineering from Nara Institute of Science and Technology (NAIST), Nara, Japan, in 2010; and Ph.D. degree in ECE from De La Salle University (DLSU) - Manila in 2016. He joined the Electronics and Communications Engineering department of DLSU in 2011, where he is currently an Assistant Professor. His current research interest involves Machine/Deep Learning, Computer Graphics, and Computer Vision applications to Health Informatics, Education Technology, Game/Graphic Content Development, and Mobile Computing. He is knowledgeable in programming languages - C/Cpp, Java, and MatLab/Octave scripting. He has more than 5 year experience in computer vision application development with OpenCV library in the mobile framework.



**John Anthony Jose** is currently working as a full-time faculty in the Electronics and Communications Engineering (ECE) Department. He received a degree of BS/MS in Electronics Engineering ladderized program in De La Salle University last 2015 with an award of Cum Laude, Outstanding master's thesis award, and Eduardo Cojuangco Jr. Award. On his licensure examination, he received an award for being "Top 3 in the ECE board examination" held in October 2015. At present, he is teaching various ECE courses and an active member of IEEE Philippine section. His research interests is on computer vision and visual recognition.



**Elmer P. Dadios** received his Ph. D at Loughborough University, United Kingdom. His main works are "Fuzzy Logic - Tool for Getting Accurate Solutions", "Fuzzy Logic - Controls, Concepts, Theories and Applications", "Fuzzy Logic - Algorithms, Techniques and Implementations", "Fuzzy Logic - Emerging Technologies and Application.", "Fuzzy-neuro model for intelligent credit risk management" and "Non-conventional control of the flexible-pole cart balancing problem:experimental results". He is university fellow and professor at De La Salle University, senior member of The Institute of Electrical and Electronics Engineers (IEEE) and the founder and chair of IEEE Computational Intelligence Society, Phils. Executive Committee.