Traffic Monitoring System by Computer Vision and Deep Learning / Xây dựng hệ thống giám sát giao thông sử dụng máy học và thị giác máy tính

Vuong Minh Phu[(1)], Le Hoang Giang[(1)]
Corresponding Author: **Ass.Dr. Ha Hoang Kha[(2)]**
(1) OISP student, Faculty of Electrical and Electronics Engineering, HCMUT, vmphu@outlook.com.
OISP student, Faculty of Electrical and Electronics Engineering, HCMUT, lhgiang149@gmail.com.
(2) Faculty of Electrical and Electronics Engineering, HCMUT, hhkha@hcmut.edu.vn

Abstract

With the increasing in the quantities of surveillance video and motivated by the industry need towards autonomous driving, vehicles, and intelligent transportation, it is necessary to implement a Intelligent Transportation System (ITS). We proposed a cooperative system focus on monitoring traffic flow from surveillance cameras including estimating vehicles velocity, wrong lane and red light violating detections. Our framework is able to work on real time from streaming data sources, however it only require manually calculating the scene scale to convert speed from pixel domain to real world. The result show that finding camera calibration parameters and camera angle are a crucial part which directly affecting to the accuracy of the whole system.

Cùng với việc tăng nhanh về số lượng của video an ninh và sự thúc đẩy của nền công nghiệp hướng tới xe tự hành và các hệ thống vận tải thông minh, để đáp ứng các nhu cầu đó chúng ta cần phải thiết kế một hệ thống giám sát giao thông thông minh. Chúng tôi đề xuất một hệ thống tích hợp việc ước tính tốc độ phương tiện, cảnh báo sai làn đường, phát hiện vượt đèn đỏ. Hệ thống này có thể nhận và tính toán dữ liệu từ nguồn trong thời gian thực, tuy nhiên việc tính toán thông số chuyển cảnh (scene scale) cần được làm thủ công. Kết quả cho thấy việc tìm các thông số và góc quay của camera là một phần quan trọng ảnh hưởng trực tiếp tới độ chính xác của toàn hệ thống

Keywords: *Camera calibration, vehicle velocity estimation, traffic flow analysis*

1. Introduction

Traffic data plays a very import role in Intelligent Transport System (ITS). Currently many efforts have been made to bring out the data to the public for traffic flow monitoring, including using roadside radar sensors on highways, inductive loop detection at intersections, GPS data collected from probe fleet, etc. Besides those devices, surveillance cameras are deployed broadly but have not been used as a mean of traffic data collection yet. Additionally utilizing these data for autonomous system is a challenging but beneficial task. From the surveillance cameras the system not only can extract the speed of vehicle but also can detect vehicles violations such as red light crossing or lane departure warning.

In order to estimate vehicle velocity, it is necessary to correctly detect the vehicles in the scenery. Object detection is one of the first studies of computer vision. Traditional techniques tend to find the contour of the object and use some noise reduction algorithms as well as background subtraction to locate the object. However these techniques often require lots of handcraft features in the image and not really robust enough for practice use. Thank to the development of artificial intelligence technology, especially the convolutional network (CNN), object detection has been improved rapidly. Leading by AlexNet [1] the trend continues and consequently many CNN architectures have been explored and shown to be practically usable in many domains, such as GoogleNet [2] , VGGNet [3], ResNet [4].

One main problem of CNNs in object detection is that the objects of interest might have different spatial locations within the image as well as different aspect ratio. Therefore, we have to select a huge number of regions and this could lead to computationally issue. Recently, researchers discovered a way to bypass this problem by choosing a limit bounding boxes or region of proposals in the image. Algorithms like RCNN [5], Fast-RCNN [6], Faster RCNN [7] are steps by steps improving the performance of the regarding task. Yet the speed of these algorithms is still not applicable for real time system. In 2016 Facebook AI Research introduced a new framework YOLO [8] which outperformed object detection algorithms by a large margin in FPS performance. This study uses YOLOv3 [9] model to achieve multiple vehicle detection.

Fig. 1. Object detection and tracking within cameras. The detected vehicles and their car types are shown on the left while their ID shown on the left

After detection, multiple objects tracking (MOT) is the next task for estimating velocity. MOT can be categorized into

two classes: 1) online methods 2) tracking by segmentation. One representative online method is simple online and real-time tracking (SORT) [10] that is based on object detection for target initialization and state estimation techniques to produce object identities. Upon that Milan et al proposed the first online MOT based on deep learning which extracts features of the object [11] and achieves top performance on the benchmarks.

Another MOT algorithm is tracking by segmentation. In [12] Chu et al develop human tracking by combining constrained multiple-kernel (CMK) tracking and Kalman filtering based on object segmentation but when background color is similar to some segments of the objects, failure may occur. Hence Lee et al [13] take advantage on 3D models of vehicles to define multiple kernels in 3D and Tang et al [14] extend their work and make use kernel-based MOT with camera self-calibration for automatic 2D-to-3D back-projection [15] which is selected as the winning method in NVDIA AI City Challenge 2017.

Converting the speed obtained in pixel domain to real world is the final step for traffic velocity estimation. As camera calibration parameters are very sensitive to the accuracy of the proposed system, the input videos must be well calibrated.

Besides, lane detection and red light violation detection are also carried out in this study. The violated vehicles license plate and appearance are captured and stored in the database.

The rest of the paper is organized as follows. Our proposed methods for each task are explained in Section 2. Experimental results are shown in Section 3. And finally the conclusion and future work will be in Section 4

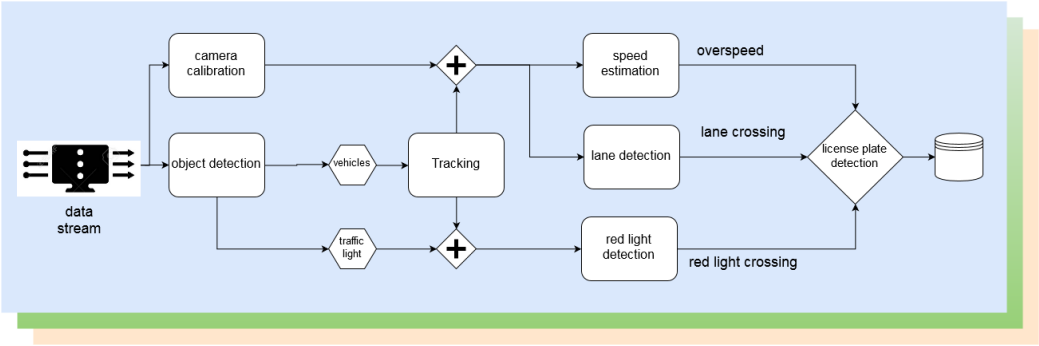

Fig. 2. Flow diagram of the proposed system

2. Methods and Implementation

The approach to traffic monitoring system is outlined in Figure 2. Details of different phrases are described in as follow.

2.1 Object detection

A pre-trained model of YOLOv3-416 detector has been used in order to simplify and reduce the training time for this task. We also manually selected 3000 frames from some of the videos which are recorded from several locations in Ho Chi Minh city. The training data are labeled in 4 categories, including cars, buses, trucks and motorcycles where each of them contains 5 to 40 objects. Besides vehicles, traffic lights are extracted from each scene for the red light detection task. We chose this architecture since it not only can achieves very fast object detection by only using a single feed-forward convolutional network to directly predict classes and bounding boxes of object but also it is easy to deploy.

All frames are extracted from the streaming link and then it passes to the detector. After that a Non-Maximum Suppression [16] algorithm is applied for subtracting the duplicated bounding boxes and making the detector more robust. Confidence scores are generated during the process, in our system 0.5 is the threshold for maintain enough amount of detections.

The tracking and speed estimation module will use the bounding boxes from YOLOv3. An example of our performance can be seen in Figure 1.

2.2 Object tracking

In this section we briefly describe the algorithm used to track vehicles. Simple online and real time tracking (SORT) is a real time tracking algorithm which has the accuracy comparable to the state of the art online trackers while supporting higher update rates. It associates objects efficiency for online and real time applications which is considerably suitable for our system. Although this framework only uses rudimentary combination of familiar techniques such as Kalman Filters [17] and Hungarian method for tracking components, it has the low number of lost targets in comparison to the other methods. A linear Gaussian state space model was carried out to approximate the dynamics of each target vehicle. The state of each target is modeled as:

$$[x, y, s, r, \dot{x}, \dot{y}, \dot{s}, \dot{r}] \quad (1)$$

Where $x$and $y$represent the horizontal and vertical pixel location of the center of the bottom two coordinates of the bounding box, while $s$and r represent the scale and the aspect ratio of the bounding box respectively. When a detection is associated to a target, the detected bounding box is used to update the target state where the velocity components are solved optimally via a Kalman filter framework. If no detection is associated to the tar-get, its state is simply predicted without correction using the linear velocity model. Additionally, the assignment cost matrix is computed as the intersection-over-union (IOU) distance between each detection and all predicted bounding boxes from the existing targets and then it is optimized by Hungarian algorithm.

2.3 Camera calibration

The main purpose of calculating camera parameters is to convert the coordinate of a point in image domain to the 3D world, as a result, the speed of vehicle can be obtained. A Diamond Space algorithm is proposed by Dubska et al [18] is deployed which based on the detection of two vanishing points on the ground plane. The model makes some basic assumptions for zero pixel skew, square shaped pixels and location of the principal point P in the center of the image that produce tolerable errors. It is proven [19] that from two orthogonal vanishing points we can get all the camera parameters. The vanishing point of the direction parallel to the vehicle's movement is denoted as the first vanishing point u and the one has the perpendicular direction to the movement of the vehicles and parallel with the road plane, v is the second point. Once we have the knowledge of these points the third vanishing points w, focal length$f$, the normal vector$n$, road planeρcan be extracted by the below equations. However, the road plane is computed only up to a scale, an arbitrary value $\delta = 1$is added to equation (7)

$$f = \sqrt{-u^T \cdot v} \tag{2}$$

$$\overline{u} = [u_x, u_y, f]^T \tag{3}$$

$$\overline{v} = [v_x, v_y, f]^T \tag{4}$$

$$\overline{w} = \overline{u} \times \overline{v} \tag{5}$$

$$n = \frac{\overline{w}}{\|\overline{w}\|} \tag{6}$$

$$\rho = [n^T, \delta]^T \tag{7}$$

Note that homogeneous 2D image coordinates are referenced by small letter $p$and the corresponding point on the image plane is also denoted as small letter but with the bar $\overline{p}$ and finally the 3D points are defined by capital letter P. With know road plane it is possible to compute 3D coordinates P of an arbitrary point $p$by projecting it onto road plane using the following equations:

$$\overline{p} = [p_x, p_y, 1]^T \tag{8}$$

$$P = -\frac{\delta}{[\overline{p}^T, 0] \cdot \rho} \tag{9}$$

It is also important to measure the scene scale λ to ompletely convert distance in image plane to 3D world and this is imply determined by utilizing Google Maps [20]. Figure 3 illustrates three orthogonal detected vanishing points as well as the horizontal line.

However, during process, we realize that the camera calibration only work well in some areas of the frame, and these areas depend on kind of camera and the camera angle while collecting data. Therefore, each video has different region of interest, we find those regions practically and make calculations within it.

2.4 Speed estimation

To estimate the velocity of tracked vehicle at frame Fwe find the center of the object from the predicted bounding box from SORT algorithm and the vehicle's center at frame$F - \tau$, where $\tau$is the tuning parameter for smoothing the noise produced by the frame and other element. Typically τfluctuates between 5 and 20 depending on different locations. The velocity is calculated by the below equation. Where P is the world coordinates obtained by (9).

$$v = \lambda \cdot \|P_t - P_{t-\tau}\| \tag{10}$$

Fig. 3. Illustration of three orthogonal vanishing points detected in Pham Van Dong Street. Red arrows are directed toward the first vanishing point and green ones to the second vanishing point. The third vanishing point is denoted by blue arrow. Also the horizontal line which connects the first and the second vanishing point is drawn by yellow color.

2.5 Lane departure warning

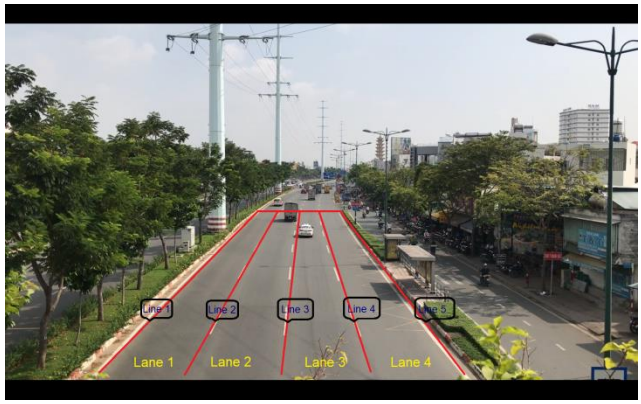

Fig. 4. Lane division in highway.

To implement lane departure warning, firstly we have to divide the road into several lanes and it is simply done by using Photoshop Portable to draw line (since a lane defined by two lines) as well as it coordinates in the image plane. It is important to note that we only consider a part of the road because we find that the camera calibration parameters are only accurate in the chosen area.

We secondly find all the equations for all the lines the plane, then for an arbitrary point in the image plane we plug its coordinate into each equation of the line pair. Next we multiply the result each other, if it is positive then the point lines between two lines, otherwise it lines outside. Take the plane in Figure 4 for example, we choose a point $P(x_1, y_2)$ (represents the center of a vehicle) and line 1 and line 2 have the following equations:

Line1: $$a_1 x + b_1 y + c_1 = 0 \tag{11}$$

Line2: $$a_2 x + b_2 y + c_2 = 0 \tag{12}$$

Then we define the dot product of $k = \overrightarrow{n_1} \cdot \overrightarrow{n_2}$ where $\overrightarrow{n_1}$ and $\overrightarrow{n_2}$are the normal vectors of line 1 and line 2 respectively.

We substitute the coordinate of P into the left hand side of equation (11) and (12) then multiply the result with k, if the result is negative then point P lines between line 1 and line 2 or vice versa. The same process is carried for the remaining

adjacent line pairs. After allocate which lane the vehicle in, it is easy to determine whether the car is changing lanes by assign ID for each lane.

2.6 Red light violation detection

We make use of the detection from section 2.2 and algorithm for lane detection in section 2.5 to solve this task. The flow diagram of the procedure is described in Figure 5.

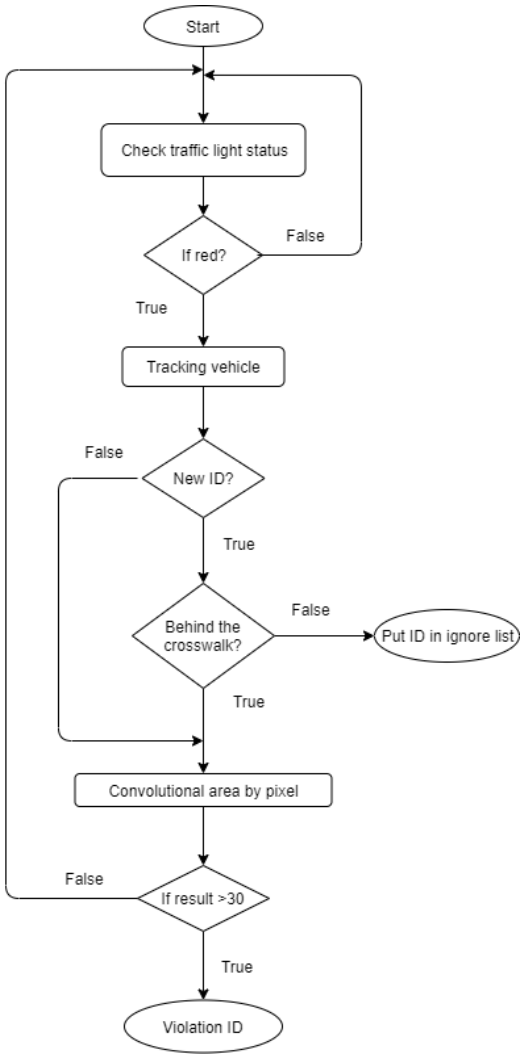

Fig. 5. Flowchart of the red light detection model.

Initially, we inherit the red light bounding box from section (2.1) then checking the traffic light status by using OpenCV library. Next if the traffic light is red, we start tracking the vehicles. Then the system decides whether the car has a new ID or not. Subsequently if an object has new ID, we need to determine its position which is behind the crosswalk or in front of it. Otherwise we move to the final step: to check the convolutional area of the tracked vehicle's bounding box with the violation area, with the result greater than 30% we can get the IDs of violated vehicles. Notice that the camera's angle in this track plays an essential role in the accuracy of the model.

2.7 License plate detection

For vehicle identifications purpose, the detection of license plates is critical. To take advantage of the rectangle and planar shape of license plates, a novel CNN network based on YOLO, SSD [21] and Spatial Transform Network (STN) [22] called Warped Planar Object Detection (WPOD-NET) [23] network is used. This network performs feed-forwarding through only 21 convolutional layers and 14 of them are residual blocks to conserve all the feature maps. Despite of the complexity, it is able to work in real time and robust to variety of distortions. WPOD itself is well trained on several dataset including USA, European, Taiwanese, Brazilian, LPs. Additionally we also add some data from Green Parking Ltd. in Ho Chi Minh city to increase the detection rate since we mostly use our system in Vietnamese vehicles.

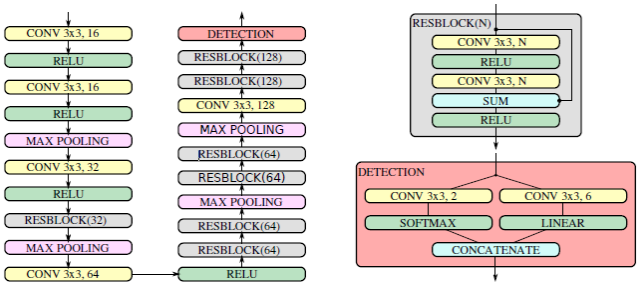

Fig. 6. Detailed WPOD-NET architecture.

3. Results and future works

Our application was implemented and tested in Python 3.6, C++ and Matlab, with the helps of Keras, Tensorflow, Numpy and OpenCV libraries. Our system estimated the speed of all vehicles on the thruways in all frame of 10 videos which was collected in Ho Chi Minh city. We observe that the system depends heavily on the calibration process and the angle of camera however in intern of mean speed value, our result is in a reasonable range. Based on our experiments, we find that the performance of lane and red light violation detection work stable in several conditions. Yet some improvements should be considered like tracking with deep convolution neural network features and more robust method to deal with the camera angle or fully automatic camera calibration strategy. The complete visualizations of our result are in the below figures.

Fig. 7. Result of lane departure warning system. The road was divided into four lanes with different colors.

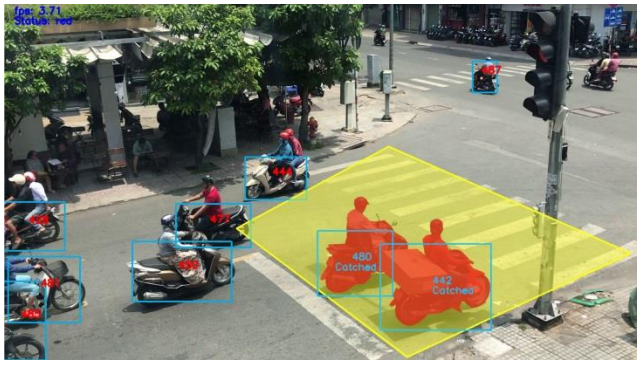

Fig. 8. Result of red light violation detection. The violated vehicles were painted in red color and the yellow area was the region of interest for detecting.

We exanimated the red light violation detection system in two different locations and it only had the accuracy of 40%. on the detection rate. The mistakes were caused because the tracking algorithm failed if there were other vehicles passing and block the camera view. In the future some adjustments should be made on the tracking mechanism to increase the accuracy.

Fig. 9. Results of overspeed detection. Assume that the speed limit was 60km/h. The violated vehicles were painted in red and the green ones were obeyed the speed limit.

We visualize the speed distribution of three different locations in Figure 9, two of them from Pham Van Dong street and the other one from Vo Van Kiet street. The average speed varies from 15 to 64 km/h.

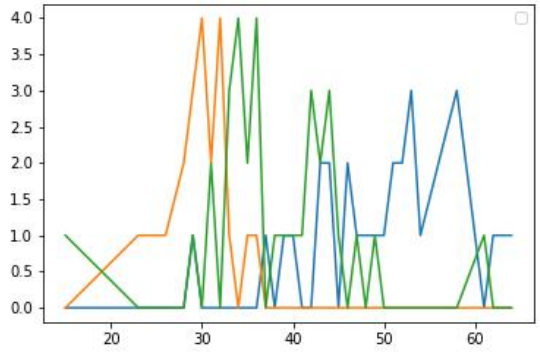

Fig. 10. Histogram of vehicle's velocity from three locations. The x-axis represents vehicle speed (km/hour) while the y-axis is the number of vehicles within the corresponding speed range.

4. Conclusions

Traffic flow analysis is important in many aspects of traffic operation and management, such as flow monitoring, incident detection, or delay cost estimation, etc. Some data collection strategies like radar sensor and inductive loop detector are not efficient in cost and surveillance cameras are only used for manually check. Utilizing the sources generated from these cameras is not only beneficial but also necessary in the age of AI technology.

This study aims to solve the problem of speed estimation, red light violation detection and lane departure warning. We proposed a pipeline with 5 main steps were carried out: 1) multi-object detection using YOLOv3 2) multi-object tracking based on Kalman filter 3) lane and red light detection 4) camera calibration for speed conversion 5) license plates detection by WPOD-Net and store to database.

This is a very good opportunity for us to understand the difficulty of the real-world problems. In the future, we believe we will keep working on the related key problems, such as multiple object tracking and vehicle re-identification, to improve the large scale surveillance video analysis.

Acknowledgement: This study was funded by office for International Study Programs – OISP, Ho Chi Minh City University of Technology, VNU-HCM.

5. References

[1] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep ConvolutionalNeural Networks," 2012.

[2] C. Szegedy, "Going deeper with convolutions," 2014.

[3] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks For Large Scale Image Recognition," 2015.

[4] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," 2015.

[5] R. Girshick, J. Donahue, T. Darrell and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," 2013.

[6] G. Girshick, "Fast R-CNN," 2015.

[7] S. Ren, K. He, R. Girshick and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," 2015.

[8] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once:Unified, Real-Time Object Detection," 2016.

[9] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," 2016.

[10] A. Bewley, Z. Ge, L. Ott, F. Ramos and B. Upcroft, "SIMPLE ONLINE AND REALTIME TRACKING," 2017.

[11] A. Milan, H. S. Rezatofighi, A. Dick, I. Reid and K. Schindler, "Online Multi-Target Tracking Using Recurrent Neural Networks," 2016.

[12] C.-T. Chu, J.-N. Hwang, H.-I. Pai and K.-M. Lan, "Tracking Human Under Occlusion Based on Adaptive Multiple Kernels With Projected Gradients," 2013.

[13] K.-H. Lee , J.-N. Hwang and S.-I. Chen, "Model-Based Vehicle Localization Based on 3-D Constrained Multiple-Kernel Tracking," 2015.

[14] Z. Tang, Y.-S. Lin, K.-H. Lee, J.-N. Hwang, J.-H. Chuang and Z. Fang, "Camera Self-Calibration from Tracking of Moving Persons," 2016.

[15] Z. Tang, J.-N. Hwang, Y.-S. Lin and J.-H. Chuang, "Multiple-kernel based vehicle tracking using 3D deformable model and camera self-calibration," 2016.

[16] R. Rothe, M. Guillaumin and L. V. Gool, "Non-Maximum Suppression for Object Detection by Passing Messages between Windows," 2014.

[17] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," 1960.

[18] M. Dubská and A. Herout, "Real Projective Plane Mapping for Detection of Orthogonal Vanishing Points," 2013.

[19] B. Caprile and V. Torre, "Using vanishing points for camera calibration," 1990.

[20] "Google Maps. [Online] Available: https://maps.google.com/".

[21] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu and A. C. Berg, "SSD: Single Shot MultiBox Detector," 2015.

[22] M. Jaderberg, K. Simonyan, A. Zisserman and K. Kavukcuoglu, "Spatial Transformer Networks," 2015.

[23] S. M. Silva and C. R. Jung, "License Plate Detection and Recognition in Unconstrained Scenarios," 2018.