Detection Model in Cvat using AWS ec2 instance", Region 10 Conference (TENCON) 2020 IEEE, pp. 708-713, 2020.

“Bodh”, BPIT’s International Journal of Technology & Management

ISSN: 2454-8421, Volume 7, Issue 2, July- Dec 2021, Page 11-18

Dynamic Operational Time Traffic Light System

Drishti Agarwal[1], Preeti Nagrath[2], Rachna Jain[3], Narina Thakur[4]

[1, 2] Department of Computer Science and Engineering Bharati Vidyapeeth’s College of Engineering

New Delhi, India drishtis.agarwal@gmail.com
[3] *Department of Information Technology, Bhagwan Parshuram Institute of Technology New Delhi, India*
4 Department of Computer Science and Engineering Bhagwan Parshuram Institute of Technology New Delhi, India
[1]drishtis.agarwal@gmail.com, [2]preetinagrath1@gmail.com, [3]rachnabtec@gmail.com, [4]narinat@gmail.com

Abstract - The process of detecting, tracking, and counting moving vehicles is crucial for traffic flow monitoring, control, and planning. In comparison to other solutions, video-applicable solutions are simple to set up and do not obstruct traffic flow. This paper provides a video-based system with the help of advanced methods implemented in python language and OpenCV computer vision kits, indicating that the outlined approach can effectively perform detection, tracking, counting of moving vehicles, and calculate the operational time for traffic lights by examining the traffic flow video series recorded from a video-based source (camera).

Keywords--- Tensor, **OpenCV, traffic, operational time, detect, vehicle, count.**

I. INTRODUCTION

With the growing number of automobiles, expressways, highways, and roadways are becoming increasingly congested. Smart Transportation Systems are being developed around the world to gather, process, and organize data from diverse sources to improve the efficiency and reliability of transportation flows and render them increasingly effective, smarter, and safe. As a result, the process of detecting, following, and counting the number of passing vehicles is becoming increasingly crucial for traffic control. Infrared detectors, inductive loop detectors, radar detectors, and video-based systems are used in the traditional method of vehicle detection. Surveillance cameras are highly influenced by the external environment, such as harsh weather conditions, shadows, sun- light, and so on, therefore video-implemented systems have an advantage. Moreover, video-based surveillance systems can provide many benefits such as easy setup, conveniently modified, do not disturb the traffic flow etc. As a result, they have attracted researchers all over the globe [1]. Road congestion is a significant concern that impacts many regions throughout the world. Numerous factors contribute to the heavy traffic situation.

The population of citizens settling in cities has increased significantly, resulting in a surge in the number of automobiles. Roadway infrastructure, on the other hand, has developed slowly and is now insufficient. As a result, there is a mismatch between the number of cars and the number of roads, leading to traffic, particularly in large cities. Inadequacies in public transit networks contribute to the same issue. Another reason is that the traffic management system is flawed and does not provide real-time traffic updates. If the traffic issue is not addressed, it appears to become worse in the future. Intelligent transportation systems are adaptive, intending to increase vehicle flow (the number of vehicles passing in a given time-frame) and decrease the average wait time interval for all cars in a system-controlled area. As a result, the purpose of this model is to build and implement a traffic control system based on real-time data sources using appropriate and favorable algorithms. The real-time vehicle data is acquired from traffic cameras installed at a four-way intersection. For this purpose, we have used a powerful tool OpenCV that is Open Source Computer Vision Library in Python. By using OpenCV, we were able to develop real-time computer vision applications. Road congestion issues have also been solved using computer vision. Take, for example, a video stream of the road that may be analyzed to identify and measure the total number of vehicles on the road. Computer vision can also be used to calculate additional information such as a vehicle's speed or traffic density. This will also benefit two categories of people: commuters and traffic management personnel. Moreover, if commuters are already aware of real-time traffic data, one can utilize it to determine the optimal mode of transportation and avoid traffic congestion. Furthermore, traffic authorities can include information on traffic density flow into their traffic control systems, resulting in improved traffic control. On real-time traffic film, numerous approaches for vehicle detection and counting have been developed. We must first detect the blobs in the movie before counting them. This is done by a series of processing on every frame of the video such that the blobs can be detected precisely.

The whole process of detection involves 3 major steps: Background Subtraction followed by vehicle detection and then finally counting them. Firstly, we take the grayscale background image of the road and calculate the grayscale difference between the current grayscale frame and the average of the scene so that the background is masked out and the blobs are detected. Then the frame is applied to various filters such as Gaussian Blur to smoothen the image so that detecting whole blobs as one unit becomes easy. Secondly, by performing a binarization process to obtain a foreground area, and then with the help of morphological operations to eliminate noise and shadow. Once the blobs are detected the next thing is to count them in such a way that no blobs are counted more than once. It is ensured by a calculating distance of blobs in consecutive frames and if it turns out to be less than a threshold value it’s considered as the same vehicle. Once all the vehicles in the lane are calculated for each road in the junction and lastly based on traffic density the operational time gets calculated for each traffic light. The paper is organized as follows:

I. Related Work
II. Data Collection
III. Methodology
IV. Experimental Results
V. Conclusion and Future scope.

II. RELATED WORK

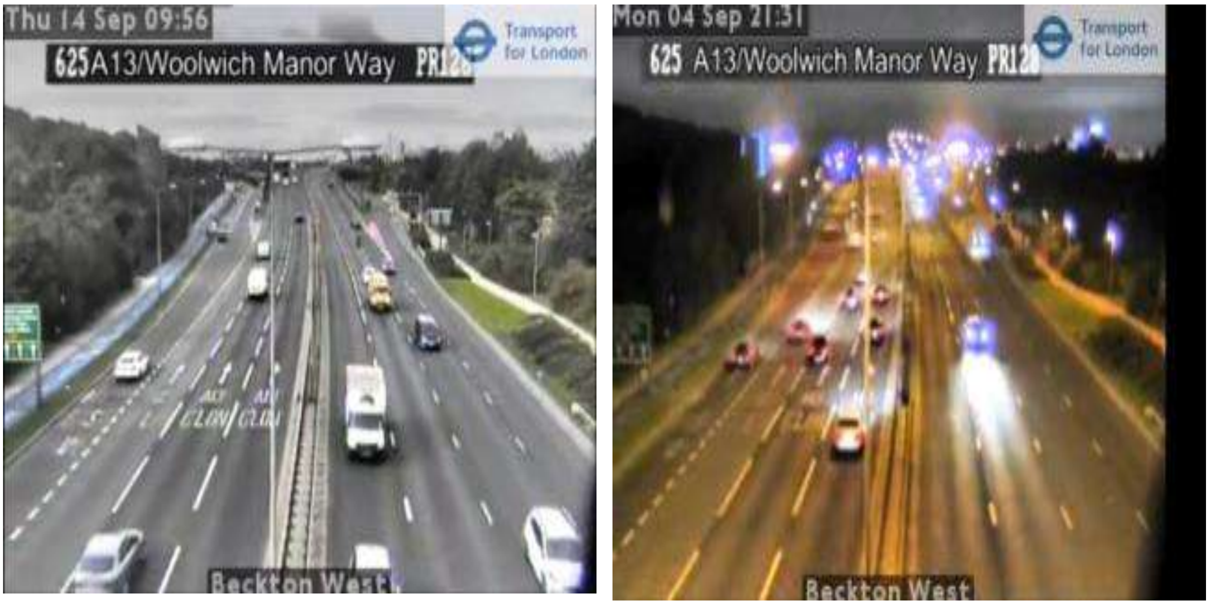
In September 2012, Vivek Tyagi, Senior Member, IEEE with his team published their research. The weightings of the mixture in the accumulated signal vary based on the traffic density conditions [6], in which the identified occupancy is based on the noise from vehicles, which contains numerous noise signals. They collected the spectral components of the roadside acoustic data utilizing Gaussian mixture concepts and modeled the class conditional probability distributions of these feature vectors conditioned on broad traffic density states: jammed, medium-flow, and free-flow traffic (GMMs). The GMM parameters are created by manually labelling 2.5 hours of roadside acoustic data traffic density levels. When 20 to 30 seconds of audio input proof is supplied, a Bayes’ model is used to classify traffic density groups based on these dispersions, with approximately 95 percent accuracy. A radial basis function (RBF) kernel classifier based on a support vector machine (SVM) was often used. [7]. CHEN Wenjie, CHEN Lifeng, and CHEN Zhanglong of Fudan University [8] submitted their work on developing a dynamic traffic signal model based on traffic detection utilising sensors. The traffic density was calculated using a range of sensors, thus it was a traditional and costly technique to deal with the problem. In 2007, Erhan Bas¸ published a detection technique for counting the volume of traffic in surveillance video. They proposed an innovative transportation video analysis method that takes into account the architecture of the scene and employs adaptive bounding box dimensions to identify and detect vehicles based on their approximate distance from the camera. Throughout the rest of the study, they recommend algorithms for vehicle detection and tracking and present results obtained by applying these algorithms to various video footage gathered at various time periods at one place in Istanbul [9].

Luis Unzueta and colleagues submitted a paper in 2012. They developed an adaptive multi-cue segmentation method which distinguishes between pixels associated with mobile and stopped cars. First, an adaptive threshold is applied to a mixture of brightness and chrominance discrepancy charts between the learned backdrops and the present frame. It tends to add additional information derived from gradient distinctions to aid in the classification of shadowy automobiles of rendered shadows, as well as the removal of roadside headlamp reflections. The segmentation is then approached in two steps, combining the ease of a linear 2-D Kalman filter with the difficulty of a 3-D volume approximation using Markov chain Monte Carlo (MCMC) methodologies. Experiments show that technology can number and categorize objects in real- time with performance comparable to inductive loop detectors in a variety of environments [10].

III. DATA COLLECTION

The data collected can be from any video source such as webcam, CCTV-cam, mobile cam etc. The model can work on live video feed from the traffic junction. For demonstration purposes the video data is collected from: https://s3-eu-west- 1.amazonaws.com/jamcams.tfl.gov.uk some video samples are as in Fig.1.

Fig.1. (a) Day time CCTV footage (b) Night time CCTV footage



IV. METHODOLOGY

This Section is all about the detection and counting of vehicles in the video footage. This section is further divided into three subsections giving the details of each step.

A. Foreground Extraction Or Background Subtraction

Foreground extraction is the process using which we can extract the foreground components of any frame. In particular, a background image of the road, which contains no vehicle, and the current frame in the video are converted from color (RGB) to HSV image. The recorded frame’s valuation is then subjected to a non-linear bilateral filter. It converts each pixel’s intensity to grayscale by replacing it with a weighted average of intensity values from neighbouring pixels. The gray concentration of the backdrop picture is then deducted from that of the current frame for each pixel (x, y). The absolute result is saved in the same position in another image, which is referred to as a different image.

B. Vehicle Counting

Once the traffic density is counted using OpenCV, the final thing is to calculate the operation traffic light duration and compare the results with the static duration of traffic lights. Before jumping to calculation part, we made some assumptions:

- No. of Vehicles in each road is a random value between 25-200.
- Rate at which vehicle leaves the traffic signal: Heavy Traffic: 25 vehicle/second, Medium Traffic: 20 vehicle/second, Low Traffic: 15 vehicle/second
- Static timed traffic signal duration is 6 seconds. For each cycle, the traffic signal with highest operational time will glow first then next turn will be in clockwise direction.

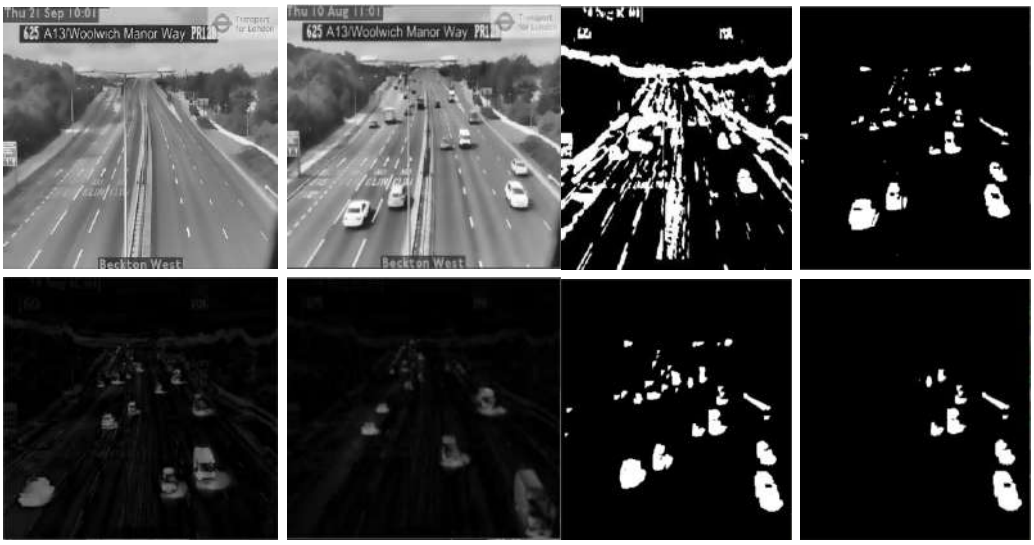


Fig.2. (a) Image of the road with no vehicles (b) After applying Bilateral Filter (c) Foreground Extraction (d) Foreground Extraction after Gaussian blur

After each Fig. 3 (a) Mean thresholding on starting frames (b) Mean thresholding on frames after sometime (c) Merging blobs after removal of noise and hole filling (d) Setting up region of interest complete cycle, the traffic signals will reset themselves and start operating in the similar fashion before.

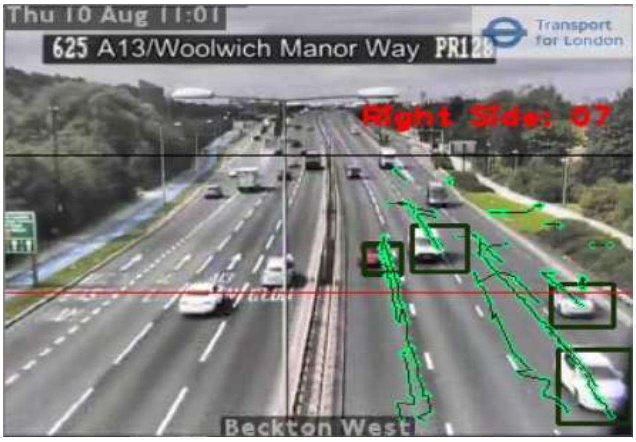


Fig.4. Final output screen counting and tracking each vehicle in Region of Interest (ROI).

C. Time Calculation

For the operation of traffic lights, we need to define some variables to store the required information needed for proper working of traffic lights. Using the python library numpy, we will create an array by the name of 'total' which will contain the total number of vehicles present in each lane for the respective traffic light, then we will create another array by the name of 'rate' which will contain the rate of vehicles arriving per second, after that another array by the name of 'dept' will contain the departure rate for every lane and two more arrays will be created by the name of 'old vehicle' contains the total number of vehicle from the beginning for each lane and 'new vehicle' which will contain the number of vehicles remaining after each second during operational time of each traffic light . Another variable 'active' is defined which will tell us the highest operational time required by a particular traffic light. Some functions we have to define in order to implement the working of traffic lights in an orderly manner.

1. **time caḷ** : It will calculate the operational time for each traffic light, that is the time interval till which each traffic light will be green during their respective turn and store it in an array by the name of 'time'.
2. **time dec:** Responsible for decreasing the operational time, that is for how much duration a particular traffic light will be switched to 'green' signal by unity.

3. **car dec:** It will calculate the number of vehicles re- maining after each second for the respective traffic light which had been set to indicate 'green' signal.
4. **light sat:** It is used to change the signal of a particular traffic light according to its operation.
5. **light operation:** It is used to operate all the traffic lights in an appropriate manner, that is by diverting the traffic flow in descending order. In order to maintain the stability along all the paths. This function is also responsible for calling the above functions by fulfilling their conditions

By implementing nested for loop, in order to call the light operation function and by updating the value of 'active'. when operational time of one traffic light has been completed by deleting the highest operational time present in 'time' till all elements are used. The execution of all the traffic lights will take place in such a way that traffic lights with the highest traffic density will be diverted first, then second highest traffic lights will be diverted and so on. Till all traffic lights have diverted the traffic flow and the cycle resets itself to be implemented again.

V. EXPERIMENTAL RESULTS

To achieve the desired result from the proposed model, first we have to take into consideration the required information for its implementation. As a result, we have taken into consideration a set of assumed data which is as follows. Initially, all the traffic lights display's 'RED' lights, from the beginning, we have taken the maximum number of vehicles in a frame to be 200 and minimum number of vehicles in a frame to be 25. In order to have efficient detection of vehicles, we divided the traffic density into three parts namely Low, Medium and Heavy.

The respective number of vehicles required to differentiate between each type of traffic is as follows, Greater than 140 (Heavy), Less than 140 but greater than 70 (Medium) and Less than 70 (Low).The departure rate that is number of vehicles entering in a particular frame per second for each type of traffic is assumed to be 25 vehicles/sec for Heavy traffic, 20 vehicles/sec for Medium traffic and 15 vehicles/sec for Low traffic. Hence, the operational time in dynamic system is calculated with the help of above information.

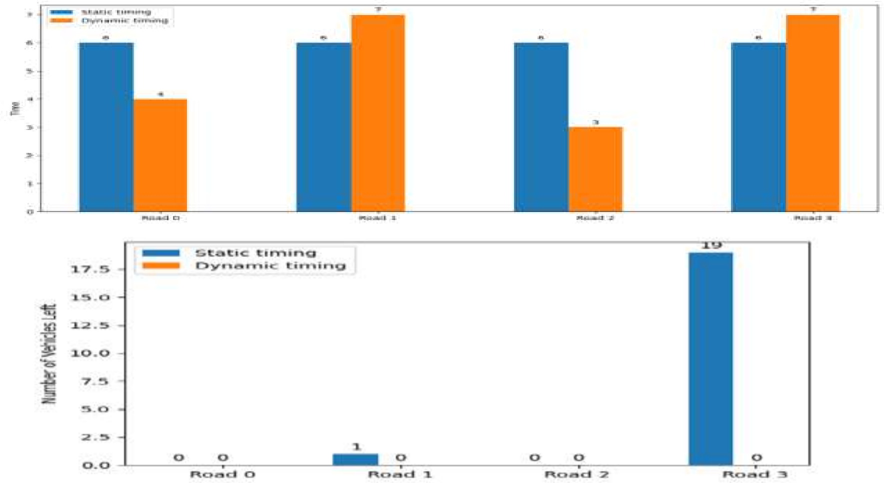


Fig.5. (a) Comparison between static and dynamic operational time (b) Comparison between total numbers of vehicles left out after implementation of static and dynamic operational time system

Finally, in order to compare our results with the static system, we have assumed the operational time (Time interval till which 'GREEN' light is displayed) of each traffic light to be 6 seconds and departure rate is same as in case of dynamic system. Hence, the experimental results displayed above are calculated, as in Fig 5(a) and Fig5 (b).

VI. CONCLUSION AND FUTURE SCOPE

In this paper, we proposed a real time object detection approach for calculation of vehicles, and then estimated the required time for the traffic flow. After that, we used the estimated time and operated the traffic lights accordingly depending upon the traffic density in a sequential manner. Image Processing played an important role in extracting the information from images. After the processing approach, we implemented the object detection techniques using OpenCV like frame differencing to notice the changes in coordinates of the vehicles under consideration in under consecutive frames and using thresholding technique to separate the vehicles and the unwanted objects from the images in order to increase the efficiency of the detection of vehicles and thus increased the accuracy for counting process of vehicles and as explained in detail in section 4. After completion of the detection process, we counted the total number of vehicles in our frame using user defined functions in python. Once the counting was completed, the operation of traffic lights was handled by the user defined functions and thus traffic lights were operated accordingly depending upon the live traffic density in an orderly manner. In the present day scenario, implementation of such a model requires a lot of different types of physical components like Arduino, camera, light bulbs, storage unit, durable structure build etc. Moreover, a GPU is also required for processing of images in real time for implementation of object detection process. A storage unit is also required, since the count of vehicles has to be stored in order to calculate the implementation time for each traffic light. An interconnected wired system has to be implemented in order to operate each traffic light accordingly. A reliable power source is also required for reasonable and error free working of the system. The requirement of a power source can be fulfilled in one way by using a solar cell and a rechargeable battery installed along the system which can be used during day as well as at night. At last, a powerful computer system is also required in order to handle execution of all the processes in a reasonable time frame so that the operation of traffic lights can be in sync with each other and thus the proposed system can work efficiently and at a reasonable fast pace in order to cope up the increasing traffic density without dealing with any problem. By fulfilling all the stated requirements, the proposed system can work efficiently in any environment and thus eliminating the problem of unsupervised traffic flow in the world.

REFERENCES

[1] Li, D., Liang, B., Zhang, W. (2014, April). Real-time moving vehicle detection, tracking, and counting system implemented with OpenCV. In 2014 4th IEEE International Conference on Information Science and Technology (pp. 631-634). IEEE.

[2] Seenouvong, N., Watchareeruetai, U., Nuthong, C., Khongsomboon, K., Ohnishi, N. (2016). A computer vision based vehicle detection and counting system. 2016 8th International Conference

[3] Y. Liu, G. Li, S. Hu and T. Ye, "Real-time detection of traffic flow com- bining virtual detection-line and contour feature," Proceedings of 2011 International Conference on Transportation, Mechanical, and Electrical Engineering (TMEE), pp. 408–413, December 2011.

[4] S. A. Amman and M. Das, "An efficient technique for modeling and synthesis of automotive engine sounds," IEEE Trans. Ind. Electron., vol. 48, no. 1, pp. 225–234, Feb. 2001.

[5] Tyagi, V., Kalyanaraman, S., Krishnapuram, R. (2012). Vehicular traffic density state estimation based on cumulative road acoustics. IEEE Transactions on Intelligent Transportation Systems, 13(3), 1156-1166.

[6] Chen Wenjie, Chen Lifeng, Chen Zhanglong and Tu Shiliang, "A realtime dynamic traffic control system based on wireless sen- sor network," 2005 International Conference on Parallel Process- ing Workshops (ICPPW'05), Oslo, Norway, 2005, pp. 258-264, doi: 10.1109/ICPPW.2005.16.

[7] E. Bas, A. M. Tekalp and F. S. Salman, "Automatic Vehicle Counting from Video for Traffic Flow Analysis," 2007 IEEE In- telligent Vehicles Symposium, Istanbul, 2007, pp. 392-397, doi: 10.1109/IVS.2007.4290146.

[8] L. Unzueta, M. Nieto, A. Cortes, J. Barandiaran, O. Otaegui and

[9] P. Sanchez, "Adaptive Multicue Background Subtraction for Robust Vehicle Counting and Classification," in IEEE Transactions on Intelligent Transportation Systems, vol. 13, no. 2, pp. 527-540, June 2012, doi: 10.1109/TITS.2011.2174358

[10] Betke, M., Haritaoglu, E., Davis, L. S. (2000). Real-time multiple vehicle detection and tracking from a moving vehicle. Machine vision and applications, 12(2), 69-83.

[11] Kowsari, T., Beauchemin, S. S., Cho, J. (2011, October). Real- time vehicle detection and tracking using stereo vision and multi-view AdaBoost. In 2011 14th International IEEE conference on intelligent transportation systems (ITSC) (pp. 1255-1260). IEEE.

[12] Jazayeri, A., Cai, H., Zheng, J. Y., Tuceryan, M. (2011). Vehicle detection and tracking in car video based on motion model. IEEE Transactions on Intelligent Transportation Systems, 12(2), 583-595.

[13] Liu, W., Wen, X., Duan, B., Yuan, H., Wang, N. (2007, June). Rear vehicle detection and tracking for lane change assist. In 2007 IEEE intelligent vehicles symposium (pp. 252-257). IEEE.

[14] Chen, Y. L., Wu, B. F., Fan, C. J. (2009, October). Real-time vision- based multiple vehicle detection and tracking for nighttime traffic surveillance. In 2009 IEEE International Conference on Systems, Man and Cybernetics (pp. 3352-3358). IEEE.Xiao, J., Cheng, H., Sawhney, H., Han, F. (2010, June). Vehicle detection and tracking in wide field-of-view aerial video. In 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (pp. 679-684). IEEE.

[15] Bedruz, R. A., Sybingco, E., Bandala, A., Quiros, A. R., Uy, A. C., Dadios, E. (2017, December). Real-time vehicle detection and tracking using a mean-shift based blob analysis and tracking approach. In 2017IEEE 9th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM) (pp. 1-5). IEEE.

[16] Chellappa, R., Qian, G., Zheng, Q. (2004, May). Vehicle detection and tracking using acoustic and video sensors. In 2004 IEEE International Conference on Acoustics, Speech, and Signal Processing (Vol. 3, pp. iii-793). IEEE.

[17] Di Stefano, L., Viarani, E. (1999). Vehicle detection and tracking using the block matching algorithm. Proc. of 3rd IMACS/IEEE, 1, 4491-4496.

[18] Kim, Z., Malik, J. (2003, October). Fast vehicle detection with proba- bilistic feature grouping and its application to vehicle tracking. In null (p. 524). IEEE.

[19] Caraffi, C., Voj´ıˇr, T., Trefny´, J., Sˇochman, J., Matas, J. (2012, September). A system for real-time detection and tracking of vehicles from a single car-mounted camera. In 2012 15th international IEEE conference on intelligent transportation systems (pp. 975-982). IEEE.

[20] Huang, L. (2010, March). Real-time multi-vehicle detection and sub-feature based tracking for traffic surveillance systems. In 2010 2nd International Asia Conference on Informatics in Control, Automation and Robotics , CAR 2010, Vol. 2, pp. 324-328, IEEE.