# Real-time Vehicle Detection and Tracking System using Cascade Classifier and Background Subtractor

Priyanshi Verma[1], Kajal Verma[1], Aprajita Singh[1], Arya Kumar Sundaram[1], Vijendra Singh Bramhe[1]

[1]Department of Computer Science and Engineering, School of Engineering & Technology, Sharda University, Greater Noida, Uttar Pradesh, 201310, India

**Abstract.** The utilization of video surveillance in the domain of Inspection and Supervision for security concern is extensive. The areas such as navigation in Military, Smart Transportation, etc. is aided by video Surveillance System. Vehicle counting systems, intelligent parking systems along with Autonomous Driving Assistant Systems (ADAS), there is a high demand for Real-time Vehicle Detection System for detecting any mobile architecture. The importance of processing a real-time stream of frames of a video stands out to be the major outlook for creating a lightweight and efficient model for giving potent results. For this purpose, we need an algorithm which can be easily configured according to the hardware and perform well, even in low processing power thereby reducing inventory costs. In this paper, we have scrutinized the Cascade Classifier and the Background Subtractor MOG2 modus operandi's for assuaging the competency of each model on different devices based on Frames Per Second (FPS) value and the amount of processing required to run in real-time. The comparison of these models has also been carried out with MobileNet and Xailient architectures for validation. The results gave us an idea of the need of including a complexity perspective into an algorithm for its actuation for proving its need in real-time.

**Keywords :** Cascade Classifier, Background Subtractor MOG2, Computer Vision (CV), MobileNet, Single Shot Object Detection (SSD), Xailient.

## 1 Introduction

With the increase in the number of humans existing over the crust, the increment in the ownership of vehicles has been escalated to an extreme extent. As per the Statista Data, the number of registered vehicles in India was reported to be 295 million until 2019 [1]. It elucidated a compound growth of 10% from 2007 to 2019 in India itself [2]. Due to this, we can often see that the roads are jammed due to extensive traffic, increase in accidents, increase in security breaches, etc. Thus, it becomes extremely important for ensuring the safety of human beings and aid the traffic issues, to develop a surveillance system for real time analysis of these vehicular prospect. Due to the increase in number of mobile structures on road, heavy obstruction is faced by common people due to traffic. This inhibits the prevention of a life in an ambulance, extrapolates the tendency of increased noise and air pollution, disrupts the daily routine of an individual which thereby affects their mental health, etc [3]. To aid this shortcomings, varied researches have been carried out for aiding Vehicular Surveillance System in an efficient manner with accurate outcomes [4].

For scanning a vehicle on a road, we need to detect it and further we can track it if certain anomalous outlook is exhibited. Thus, detectors such as You Only Look Once (YOLO), [5] RetinaNet [6], etc has been incorporated for Vehicle Detection widely. But the major constraint revolves around the hardware requirements [7] of these models. The increment in computational complexity hinders the overall performance of any workflow and when we consider doing it real-time, the approach revolving around the resources needs to be aggregated for concluding any modus operandi. In this articulation, we collated Cascade Classifier and Background Subtractor approach for detecting vehicles in a video stream in real-time and also tracking those vehicles through counting by initiating vehicle counter. Both these detectors have been fine tuned for obtaining higher efficacy. We've also integrated real-time Frames Per Second (FPS) [8] value for understanding the performance of model on varied hardware systems developing the low-specs and high-specs ones for clear explanation. Furthermore, for validating the efficacy of these models to be integrated into real-time, we also colligate them with different models such as MobileNet Single Shot Detection (SSD) [9] and Xailient [9] for deep explication. This analysis gave us an insight of the efficacy of each model and their potency in Vehicle Detection prospect and also an ideation about the impact of each model over varied hardware systems and their stability in terms of efficiency on those systems.

The outlook of the paper is structured as related work in Section-II, modus operandi utilized for analysis in Section-III, actuation of the workflow with varied tools in Section-IV, experimental results and analysis in Section-V, and ultimately concluding it in Section-VI.

## 2 Related work

Initiating for the Vehicle Detection prospect we scrutinized over diverse approaches, to ultimately formulate the potency along-with the shortcomings in these approaches. Vehicle Speed Calculation system utilizing Sparse Random Projection over Raspberry Pi was instantiated gaining an effective frame rate of 7.08 fps which was improved from 3.29 fps without optimization. The process needed more optimization for even better outcomes [10]. Cheap Embedded System for Counting of Vehicle and Classifying it for Management of traffic was elucidated expelling the efficacy of 95.35% with the need of more training samples for better results [11]. For aiding the safety of Self-Driving vehicles, a vision-based approach has been conferred with the integration of OpenCV python library for Object Detection on the go. The efficacy turned out to be decent giving an edge for integrating the given method in real-time [12].

The incorporation of Artificial Neural Network (ANN) for counting people inside public transport was exhibited for complying Transit Law in Ecuador using OpenCV gaining the effectiveness of 90% with the need of improvement for tackling obstacles for approximation. Moreover, perpetual driving aiding system was introduced using

OpenCV for detecting lanes, detection of blind spot, etc. The model was implemented over Raspberry Pi for gauging accuracy [13]. A vehicle counting system was also effectuated utilizing background subtraction method for specifically complying to The Sidoarjo Toll Road gaining an efficacy of 92.3% in the morning and 77.3% in the evening, giving an insight of the importance of camera position and the environmental outlook for accurate results [14]. The same method of Background Subtraction for Vehicle Detection but in real-time was also elucidated, formulating an Intelligent Transportation Systems (ITS) for traffic monitoring which gained an average accuracy rate of 71.38% with the shortcoming of proper tuning [15]. Furthermore, a real-time traffic data was collected for vehicle detection, counting, classification and speed measurement through adaptive video-based approach gaining an accuracy of 82% for real-time video and 81% for recorded video having the limitation of video processing in real-time [16]. Integration of Haar Algorithm for Object Detection was scrutinized for analysing its potency through Arduino Board in real-time [17]. You Only Look Once (YOLOv3) algorithm has also been consummated for performance analysis of Object Detection and Tracking outlook gaining an average accuracy of 92% over KITTI dataset. YOLOv3 performs well over images but when induced for real-time analysis requires a good system for its application [18]. The CAMSHIFT algorithm has also been effectuated for vehicle detection in real-time where it performed well but with the issues of including shadows in its detections [19].

Thus, commemorating all the minutes of these approaches we found the major limitation in the resource's perspective as well as the hindrance in the accuracy due to it. As a result, we came to a conclusion of actuating the modus operandi and thereby analyzing those results in real-time to provide an insight about the reason of shortcomings in the previous approaches.

## 3 Methodology Used for Analysis

The literature survey related to vehicle detection phenomena gave us an overview of best approaches and the most common ones for juxtaposing them simultaneously. Thus, we constricted ourselves to Cascade Classifier and Background Subtractor MOG2 for our explication. Let's look upon each approach algorithm in detail.

### 3.1 Haar Cascade Classifier

Haar Cascade Classifier is entrenched over 'Haar Wavelets' [9]. It can basically be coined as a series of rescaled 'square-shaped' functions which is collated to formulate a family of wavelet or create its basis. Haar Algorithm can be elucidated in four phases:

- The calculation of Haar Features
- The creation of integral images
- The utilization of Adaboost
- Finally, the Implementation of Cascading Classifiers

Basically, it can be subsumed as a Haar Wavelet operation for analysing pixels present in an image in the form of squares through a function. Training Data is utilized (if the dataset is provided) for obtaining a higher order of efficacy with the collation of machine learning technique. The features present in the dataset is detected and computed through a concept known as an Integral Image. The Adaboost algorithm which is utilized for enhancing the learning perspective of a model through the detection of minute features and thereby making the output more efficient is used in Haar Cascades. The Cascade Classifiers are stored as an .xml files for any object detection. Figure.1 demonstrates the diverse and unique Haar Features for Feature Extraction.

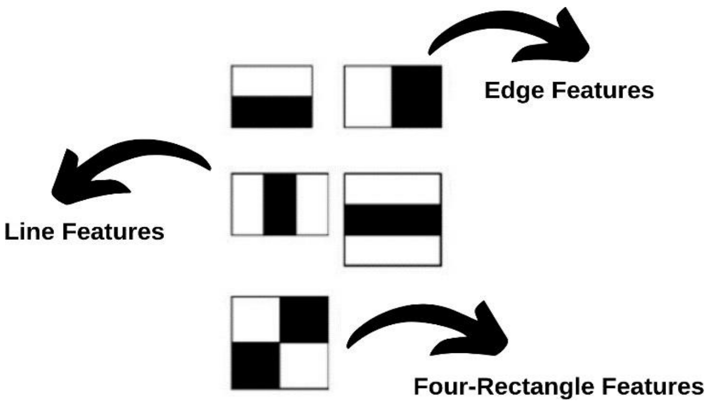


**Fig. 1.** Demonstration of variety of Haar Features.

### 3.2 Background Subtraction

One of the methods for detecting objects in motion is Background Subtraction [15]. Sometimes this method is also coined as Foreground Detection. The process of detecting foreground from a background is one of the famous techniques in the field of Image Processing and Computer Vision. The modus operandi of Background Subtraction is incorporated for detection of moving entities over videos captured through static cameras. This approach is implemented on the basis of the difference of background reference and the frame.

Furthermore, with the background subtractor, morphology is also induced, which is also an image processing outlook that works on image segmentation for escalating segmentation accuracy. The morphological modus operandi is basically used for binary images, grey images and images with varied intensity values. The most widely used morphological techniques are Dilation and Erosion. Dilation majorly focuses on accentuating the objects through addition of layers, whereas Erosion turns out to be opposite of Dilation, i.e., it aids the erosion of edges of any entity [16].

Moreover, an edge detector exhibits an edge formulation in binary image elucidation, where the pixels at the edges are represented in white and the remaining area is converted to black. Thus, for better separation between these edges it’s necessary to have a bind between two edges. Contours is coined as the sequence of pixel information forming the encapsulating boundary of a region. The utilization of contours for vehicle detection stands out to be one of the most prominent modules. Thus, the collation of all these aspect aids in consummating the modus operandi for Vehicle Detection using Background Subtractor. These techniques have been induced in real-time for testing their applicability and proving their potency. Thus, implementation over these prospects was initiated. Figure.2 represents the Background Subtractor overview.

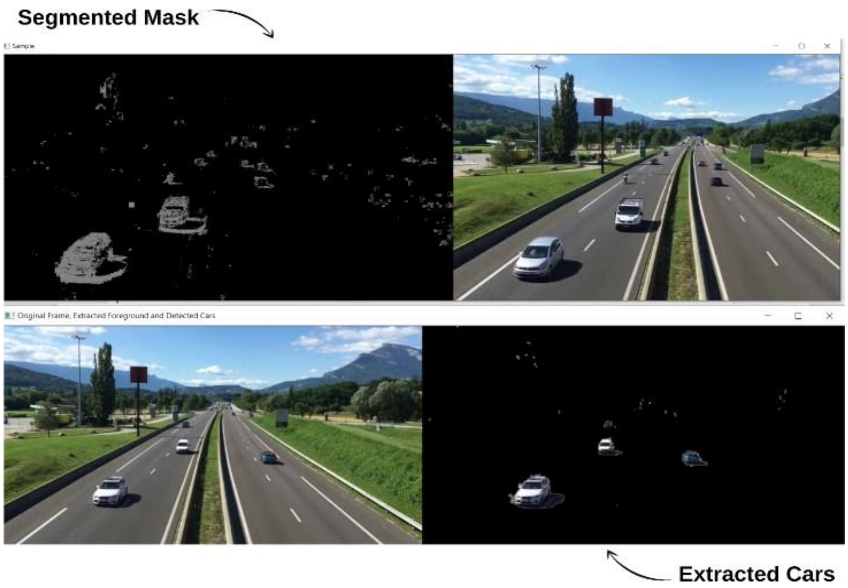


**Fig. 2.** Demonstration of Background Subtractor Output.

## 4 Implementation and Tools

Initiating with the process, we first got our hands on with the Cascade Classifier and moved on to the Background Subtractor Algorithm. The modus operandi for the proposed outlook has been mentioned below.

### 4.1 Dataset Utilized

We collated certain videos where we could see suitable number of cars that would be on a normal road. As we're dealing with Frame Rate, we tried to keep videos having higher speed when compared to other. We also wanted a video which could challenge the model for predicting accurate results. As a result, after filtering through our requirements we selected a video downloaded from pexels.com which was freely available to utilize. Figure.3 demonstrates a static image of the chosen video.

As we're dealing with vision-based modus operandi, we incorporated Computer Vision OpenCV [20] python library for implementation of both the algorithms.

**Fig. 3.** Demonstration of Dataset Utilized.

### 4.2 Algorithms Implemented

After deciding the Dataset and the computation framework, we stumbled upon the algorithmic outlook. Figure.4 graphically represents the implementation overview of Cascade Classifier for the proposed system. Similarly, Figure.5 demonstrates the actuation of Background Subtractor MOG2 for the exhibited formulation.

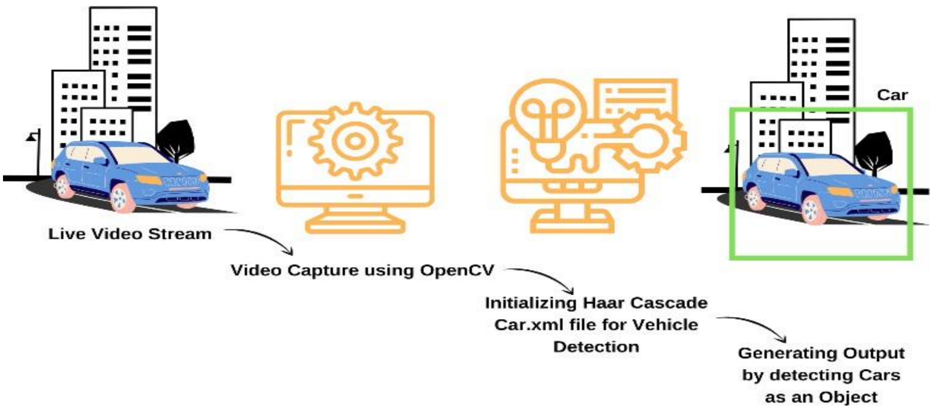


**Fig. 4.** Implementation Flow of Cascade Classifier

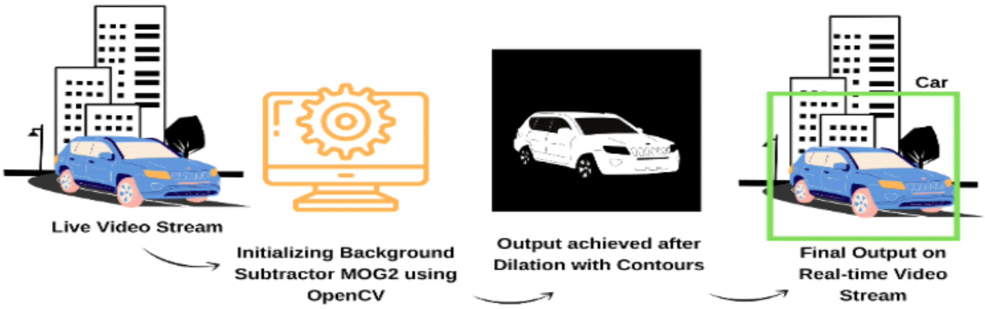


**Fig. 5.** Implementation Flow of Background Subtractor MOG2.

For cascade classifier we utilized Cars.xml Haar Cascade File available over GitHub repository for analysis through OpenCV. Furthermore, we integrated Background Subtractor MOG2 which is an improved version of Gaussian Mixture Model i.e., Background Subtractor MOG [1]. The major difference between both the prospects is, MOG2 has an option of choosing whether to include shadows for detection or not, whereas MOG doesn't have it.

Also, MOG2 provides efficient adaptability for diverse scenes due to alteration in illumination. Due to all these improvements, we utilized MOG2 over MOG for our explication. We also inculcated Frames Per Second (FPS) value in real-time for both the models to gauge the efficacy of the model, through encapsulating processing anticipation.

### 4.3 Resources Utilized

As we're colligating the computational complexity outlook for inferring an elucidation, we aggregated two systems with both lower processing power and high processing power, but both explicated over CPU processing itself and no GPU was utilized. i] The Low-End System: iball CompBook Pentium Quad Core Processor with Windows 10 Operating System. ii] The High-End System: Honor Magicbook 15 Ryzen-5 Quad Core 3500U with Windows 10 operating system.

The High-End here doesn't mean an extremely expensive system. We considered only the power of processors which can be consummated through their generations for inferring conclusion. The core remains the same i.e., both the systems are Quad Core Processors.

### 4.4 Metrics Used for Analysis

Frames per Second (FPS) Value has been inferred for gauging the potency of model over varied systems. The FPS value is calculated as: Demonstrated in Equation.1 and Equation.2. Also, the accuracy of both the algorithms has also been calculated, taking Vehicle Count (VC) as the reference point. Equation.3 denotes the accuracy calculation.

We ran the same code over both the systems and the result we got was worth the scrutiny as we got wonderful insights which was never juxtaposed in previous researches and the modus operandi proved to be an explication that was much needed. The experimental outcomes are elucidated in further section with real-time representation of Output with Frames Per Second (FPS) value and also the Vehicle Count.

## 5 Experimental Results and Analysis

As we implemented our proposed system in real-time, we got phenomenal insights. Figure.6 displays the output results for the Cascade Classifier with Frame Rate and the Vehicle Count in real-time. Similarly, Figure.7 represents the output results for the Background Subtractor MOG2 with Frame Rate and the Vehicle Count. The output is illustrated from the High-End device and the accuracy of these algorithms stood out to be best till now.

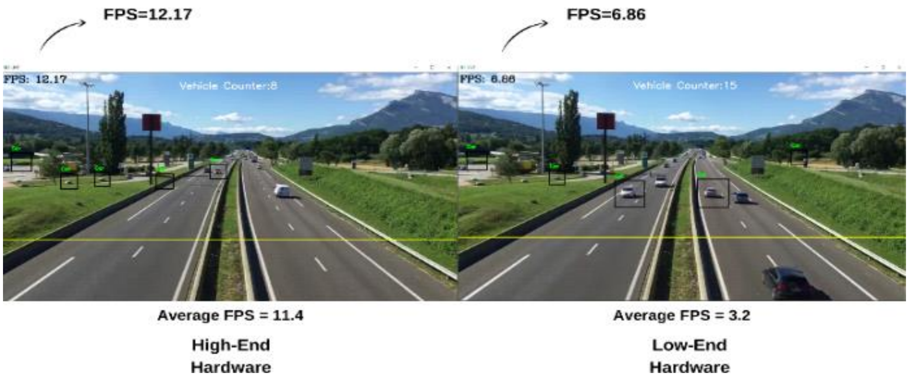


**Fig. 6.** Real-time Output for Cascade Classifier with FPS value and Vehicle Counter.

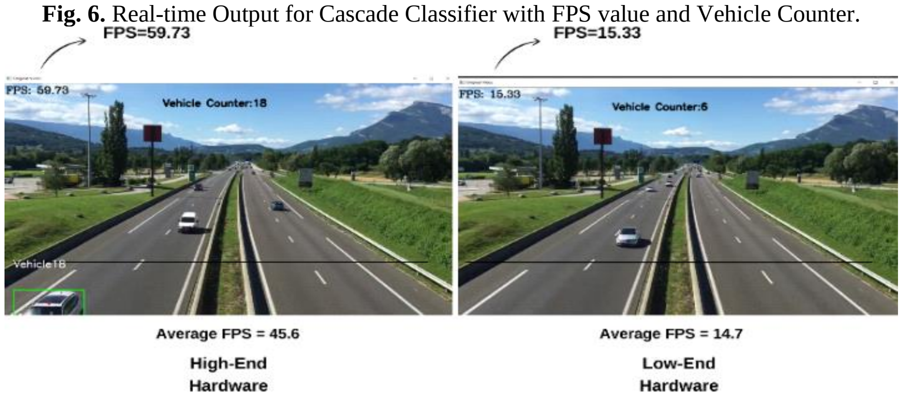


**Fig. 7.** Real-time Output for Background Subtractor MOG2 with FPS and Vehicle Counter.

After visualizing the results, we inferred certain overview of the model's performance and analogized the accuracies of the models through Equation.3. The accuracy was calculated through high-end system as the efficacy is based on number of correct predictions and not on Frames Per Second (FPS) value. Table-1 demonstrates the accuracy for both the models.

**Table 1.** Accuracy Illustration

| SN | Models | Accuracy( in %) |
|---|---|---|
| 1. | Casscade Classifier | 92 |
| 2. | Background Subtractor MOG2 | 97 |

Table-2 and Table-3 demonstrates the FPS comparison between all the approaches and also on both the hardware's for explication.

**Table 2.** FPS Value Comparison on High-end Hardware for different models.

| Models | Hardware Utilized | Cores Used | FPS Value | CPU Utilization(%) |
|---|---|---|---|---|
| MobileNet SSD [22] | Honor Magicbook 15 | 4 | 23.3 | 80 |
| Xailient [22] | Honor Magicbook 15 | 1 | 27.3 | 60 |
| Cascade Classifier | Honor Magicbook 15 | 4 | 11.4 | 85 |
| Background Subtractor MOG2 | Honor Magicbook 15 | 4 | 45.6 | 72 |

**Table 3.** FPS Value Comparison on Low-End Hardware for different models.

| Models | Hardware Utilized | Cores Used | FPS Value | CPU Utilization(%) |
|---|---|---|---|---|
| MobileNet SSD [22] | Iball CompBook | 4 | 0.8 | 95 |
| Xailient [22] | Iball CompBook | 1 | 8.2 | 75 |
| Cascade Classifier | Iball CompBook | 4 | 3.2 | 94 |
| Background Subtractor MOG2 | Iball CompBook | 4 | 14.7 | 85 |

## 6 Conclusion and Future Scope

The above explication resonates the efficacy of Background Subtractor MOG2 over Cascade Classifier in terms of accuracy for Vehicle Detection and Counting. As the feature extraction is escalated through Adaboost addition, the results for Background Subtractor MOG2 allows it to be a prominent solution for real-time Vehicle Detection for Traffic Monitoring Systems. The Cascade Classifier lags in terms of Vehicle Detecting and Tracking and also detects false entities. Furthermore, when the comparison is done upon utilization of resources Background Subtractor MOG2 and Xailient turns out to be the most efficient one. Through this explication, we can infer that a model possessing the capability of producing potent results along with the ability to consume less power can be brought to existence with fine tuning i.e., Background Subtractor MOG2.

Moreover, the elucidation can be done for a Traffic Surveillance System in real-time by building a prototype. Also, more tuning can be done for optimizing these algorithms for producing higher efficacy.

## REFERENCES


1. S. Choudhury, S. P. Chattopadhyay and T. K. Hazra, "Vehicle detection and counting using haar feature-based classifier," 2017 8th Annual Industrial Automation and Electromechanical Engineering Conference (IEMECON), 2017, pp. 106-109.
2. R. M. Waregaonkar, S. M. Deokar, S. S. Patankar and J. V. Kulkarni, "Development of prototype for vehicle speed measurement," 2017 2nd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 2017, pp. 945-949.
3. S. K. Mankani, N. S. Kumar, P. R. Dongrekar, S. Sajjanar, Mohana and H. V. R. Aradhya, "Real-time implementation of object detection and tracking on DSP for video surveillance applications," 2016 IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), 2016, pp. 1965-1969.
4. A. P. Kulkarni and V. P. Baligar, "Real Time Vehicle Detection, Tracking and Counting Using Raspberry-Pi," 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), 2020, pp. 603-607.
5. C. R. Rashmi and C. P. Shantala, "Vehicle Density Analysis and Classification using YOLOv3 for Smart Cities," 2020 4th International Conference on Electronics, Communication and Aerospace Technology (ICECA), 2020, pp. 980-986.
6. V. S. Sindhu, "Vehicle Identification from Traffic Video Surveillance Using YOLOv4," 2021 5th International Conference on Intelligent Computing and Control Systems (ICICCS), 2021, pp. 1768-1775.
7. F. H. Shubho, F. Iftekhar, E. Hossain and S. Siddique, "Real-time traffic monitoring and traffic offense detection using YOLOv4 and OpenCV DNN," TENCON 2021 - 2021 IEEE Region 10 Conference (TENCON), 2021, pp. 46-51.
8. Chato, D. J. M. Chipantasi, N. Velasco, S. Rea, V. Hallo and P. Constante, "Image processing and artificial neural network for counting people inside public transport," 2018 IEEE Third Ecuador Technical Chapters Meeting (ETCM), 2018, pp. 1-5.
9. https://xailient.com/real-time-vehicle-detection-with-mobilenet-ssd-and-xailient/
10. C. C. Paglinawan, A. N. Yumang, L. C. M. Andrada, E. C. Garcia and J. M. F. Hernandez, "Optimization of Vehicle Speed Calculation on Raspberry Pi Using Sparse Random Projection," 2018 IEEE 10th International Conference on

Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM), 2018, pp. 1-6.
11. L. C. Pico and D. S. Benítez, "A Low-Cost Real-Time Embedded Vehicle Counting and Classification System for Traffic Management Applications," 2018 IEEE Colombian Conference on Communications and Computing (COLCOM), 2018, pp. 1-6.
12. D. Dong, X. Li and X. Sun, "A Vision-Based Method for Improving the Safety of Self-Driving," 2018 12th International Conference on Reliability, Maintainability, and Safety (ICRMS), 2018, pp. 167-171.
13. S. Raviteja and R. Shanmugha sundaram, "Advanced Driver Assitance System (ADAS)," 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), 2018, pp. 737-740.
14. H. Yuliandoko, M. D. Ayatullah and F. E. Purwita, "Automatic Vehicle Counting Using Raspberry pi and Background Subtractions Method in The Sidoarjo Toll Road," 2019 2nd International Conference of Computer and Informatics Engineering (IC2IE), 2019, pp. 83-86.
15. D. R. Ignatius Moses Setiadi, R. R. Fratama, N. D. Ayu Partiningsih, E. H. Rachmawanto, C. A. Sari and P. N. Andono, "Real-Time Multiple Vehicle Counter using Background Subtraction for Traffic Monitoring System," 2019 International Seminar on Application for Technology of Information and Communication (iSemantic), 2019, pp. 1-5.
16. A. Ghosh, M. S. Sabuj, H. H. Sonet, S. Shatabda and D. M. Farid, "An Adaptive Video-based Vehicle Detection, Classification, Counting, and Speed-measurement System for Real-time Traffic Data Collection," 2019 IEEE Region 10 Symposium (TENSYMP), 2019, pp. 541-546.
17. L. Koraqi and F. Idrizi, "“Detection, identification and tracking of objects during the motion”," 2019 3rd International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT), 2019, pp. 1-6.
18. N. Jain, S. Yerragolla, T. Guha and Mohana, "Performance Analysis of Object Detection and Tracking Algorithms for Traffic Surveillance Applications using Neural Networks," 2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2019, pp. 690-696.
19. W. Hou, D. Xia and H. Jung, "Video Road vehicle detection and tracking based on OpenCV," 2020 International Conference on Information Science and Education (ICISE-IE), 2020, pp. 315-318.
20. A. C. Cormoş, R. Andrei Gheorghiu, V. A. STAN and I. Spirea Dănăilă, "Use of TensorFlow and OpenCV to detect vehicles," 2020 12th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), 2020, pp. 1-4.