



# INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X

Impact Factor: 6.078

(Volume 7, Issue 3 - V7I3-2200)

Available online at: <https://www.ijariit.com>

## Multiple vehicle speed detection on camera feed

Manasa Battula

[manasabattula12@gmail.com](mailto:manasabattula12@gmail.com)

Andhra University College of  
Engineering for Women,  
Visakhapatnam, Andhra Pradesh

Dr. K. Soumya

[soumyacf@andhrauniversity.edu.in](mailto:soumyacf@andhrauniversity.edu.in)

Andhra University College of  
Engineering for Women,  
Visakhapatnam, Andhra Pradesh

Chandini Beeram

[beeramchandinireddy@gmail.com](mailto:beeramchandinireddy@gmail.com)

Andhra University College of  
Engineering for Women,  
Visakhapatnam, Andhra Pradesh

Sarayu Bendalam

[sarayubendalam@gmail.com](mailto:sarayubendalam@gmail.com)

Andhra University College of  
Engineering for Women,  
Visakhapatnam, Andhra Pradesh

Mounika Bhimavarapu

[mounika13231@gmail.com](mailto:mounika13231@gmail.com)

Andhra University College of  
Engineering for Women,  
Visakhapatnam, Andhra Pradesh

### ABSTRACT

Technology is evolving day by day which leads to automation in many aspects. This also applies to traffic concerned issues. We are having saturation of the transportation infrastructure due to the growing number of vehicles over the past few decades. It affects our lives particularly in the urban areas, while people need to move rapidly between different places. This results in traffic congestion, accidents, transportation delays and larger vehicle pollution emissions. One of the major issues for accidents is driving vehicles at high speed. There is a need to control this issue which raises the question 'How?' Several solutions were introduced to reduce this problem. "The number of pedestrians killed in road crashes due to over speed increased from 22,656 in 2018, to 25,858 in 2019 i.e., an increase of about 14.13 percent." – News 18. This paper presents a way to control traffic in an efficient way using surveillance cameras. This proposed model can detect the speed of any number of vehicles using surveillance cameras. This model captures the images of the offenders and stores them in a folder. This model detects the speed with accuracy and correctness.

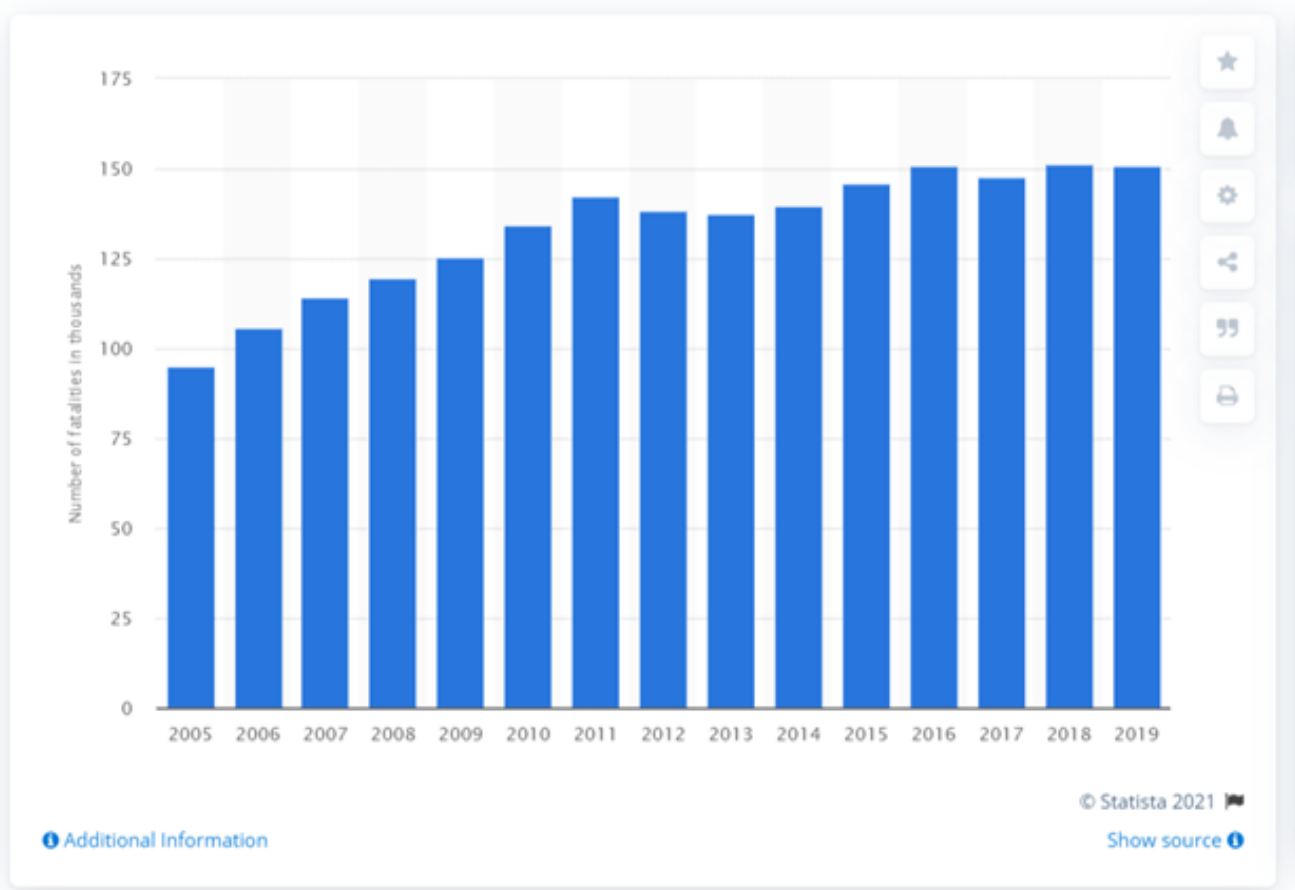
**Keywords**— Video Acquisition, Background subtraction, OpenCV, Morphing Techniques, Gaussian mixture-based foreground/background (algorithm), Offenders detection, Edge detection

### 1. INTRODUCTION

Traffic is a common issue in big cities and towns. The main reason for this problem is rapid growth of population and the increasing number of vehicles. At the same time, Vehicle manufacturers are producing vehicles that can move faster and faster, by enhancing the vehicle engine and increasing the maximum speed of the vehicle. Due to this, the numbers of accidents happening on the roads are also increasing dramatically.

The population versus road growth is not particularly even. There are some drivers who are not willing to obey the traffic rules. Overtaking tendency also causes traffic jams. One of the major issues is driving vehicles at high speed. This issue generally leads to life-threatening accidents. The below figure 1 is the analysis of deaths caused by driving vehicles at high speed from 2005-2019:

However, we can solve this problem by adopting some measures. Traffic rules should be imposed strictly so that the drivers are bound to obey them. Required traffic police must be posted on important points. Unlicensed vehicles should be removed. After doing all these things, we can hope for a good traffic system for our comfortable movement. But not all the places can be met with the above remedy measures. Video surveillance is a very popular research topic that continuously tries to detect, recognize and track the required targets. The basic ideology for any tracking method is using a detection method in every single frame. Object tracking is the process of following one or more objects that are found on a detection process using a camera. Background is the most commonly used method for simple object trackers.



**Figure 1: Deaths caused due to road accidents across India from 2005-2019**

In general, detectors are placed on the surface of the road or speed cameras are used for speed detection of vehicles. However, the equipment required for this method is costly. So, there is a need of having a low-cost vehicle speed detector system. This project uses OpenCV software for implementation. This method is automated since the vehicles detected by cameras use a technique which is very easy to trace down the speeding vehicle. This system is affordable and eliminates man power.

A study by HINDRISE states, "As per the reports, 11% of the deaths due to accidents in the world happen in India. On a more comprehensive note, accidents caused by over speeding vehicles on roads have taken the lives of more than 97k people in India in the year 2018. The most prevalent violation in terms of road accidents is driving at high speed. There is a need for better surveillance and feasible traffic control measures with more accuracy".

## **2. BACKGROUND AND RELATED WORK**

In this section, we provide theoretical work related to understanding the problem and get a solution precisely. Due to the advancement of technology, the automobile manufacturer manufactures the vehicles with best quality and the growth rate in domestic automobile production across India was about 6.3 percent in the year 2019. The production of passenger vehicles in India saw a decline of almost 20 percent in the second half of 2019, with leading manufacturers having to reduce output in order to prevent inventory overloads. In 2017, there were more than three million passenger cars sold across the south Asian country. So that, the number of vehicles on the road is increasing dramatically.

Many approaches have been conducted on detecting the speed of the vehicle. One of those is that we are presenting a robust and real-time, vision-based approach to detect speed breakers in urban environments for autonomous vehicles. It is designed to detect the speed breaker using visual inputs obtained from a camera mounted on top of a vehicle. This method performs inverse perspective mapping to generate top view of the road and segment out regions of interest based on the difference of Gaussian and median filter images. Furthermore, the algorithm performs RANSAC line fitting to identify the possible speed breaker candidate region. This initial guessed region via RANSAC, is validated using a support vector machine. This method can detect different categories of speed breakers on cement, asphalt and interlock roads at various conditions and have achieved a recall of 0.98.

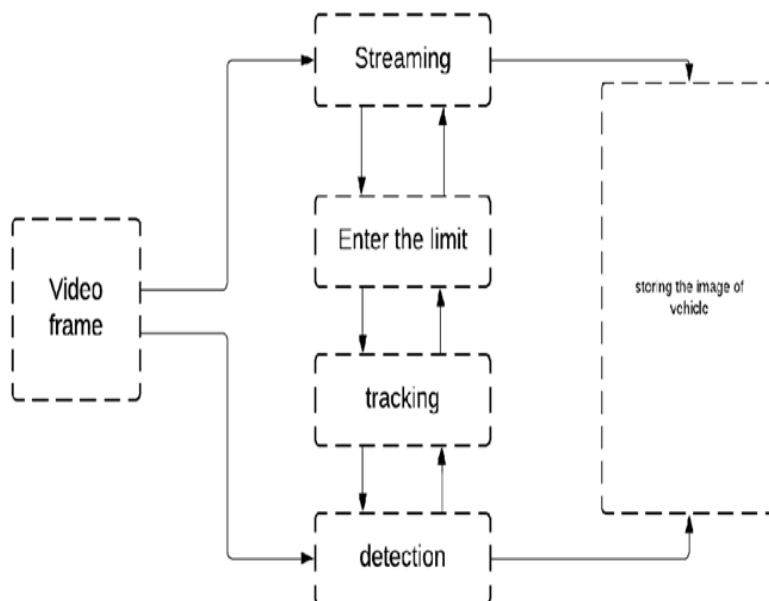
Another method is the radar system operation known as the Doppler shift phenomenon. The concept behind this system is a doppler shift that happens when for a moving vehicle the created sound is reflected off and the frequency of the returned sound is slightly changed. This method, with spatial equations and equipment, we can get the speed of a moving vehicle. However, this method still has several disadvantages such as the cosine error that happens when the direction of the radar gun is not on the direct path of the incoming vehicle and also the cost of equipment is one of the important reasons, and also shading and radio interference are two other influential factors that cause errors for speed detection and however the fact that radar sensors can track only one car at any time is another limitation of this method.

Interesting method is designed for an automated speed detection system which can detect the speed of vehicles and if over-speeding occurs, extract the license number of the vehicle and send it via email to Toll Plaza to charge the fine. This is designed for Punjab license plates which are currently implemented in Pakistan. Doppler effect phenomenon is used for the speed measurement. If over-speeding is detected, a camera automatically captures the picture of the vehicle and Digital Image Processing techniques are applied to extract the license number. MATLAB is used for image processing. Later, the extracted license plate number is sent via email to Toll Plaza. The results show that the proposed system successfully detects over speeding, extracts the license plate, has high performance and can be deployed on roads to check for over speeding.

Another existing model of Gaussian mixture model is Intelligent Transportation System. It is mostly used in the development of smart cities. Traffic engineering is the management of detecting vehicle speed on the highways. It is used to detect the speed of the moving vehicles using digital image processing. In this, the inputs are a sequence of frames, frame rate and ROI. Steps include First we separate foreground and background using Gaussian Mixture Model in each frame. For each frame, we calculate the centroid and location of the object. We can determine the speed by computing the movement of the centroid in a sequence of frames. To calculate speed, we only consider frames when the centroid is inside the predefined region of interest. After that, we transform the pixel displacement into a time unit of km/hour. System validation is done by comparing the speed calculated manually and obtained by the system. The results obtained for software testing can detect the speed of vehicles with the lowest accuracy is 77.41% and the highest accuracy is 97.52%. With the real speed of the vehicle, The detection results of testing by using real video footage on the road.

### 3. PROPOSED SYSTEM

The architecture of the proposed model is depicted in the figure below.



**Figure 2. Working module of speed detection using OpenCV**

From the above figure, the entire procedure that is followed in this project is displayed. This architecture includes the process of acquiring the video input from the camera/webcam using OpenCV library. The video input is then given to the model or classifier to identify the vehicles exceeding the speed limit. It captures the image and gives the speed of the vehicle. With the help of this technique, it is very easy to trace down the over speeding vehicle. It will store the image of the Offenders in the folder who have crossed the speed limit. This is how this system has been implemented to eliminate the manpower involved and make it fully automated.

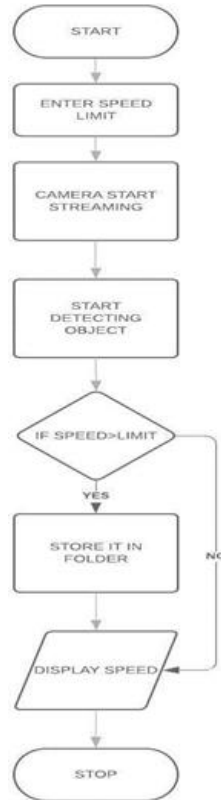
#### 3.1 Working

This program uses Open cv to detect and capture car with help of a traffic camera, once a vehicle passes through a certain distance its speed is calculated and displayed. This process takes place by road tracking which consists a technique known as line segmentation. Line segmentation is a crucial issue in trajectory planning which classifies different lanes and generate definite driving areas. Once the process is over, detection of vehicle in short image acquisition takes place.

Image acquisition is a digital image processing technique, which is defined as an action of retrieving an image from some source usually some hardware source. After the image is acquired, image subtraction has been done where the digital numeric value of one pixel or whole image is subtracted from another image. After completion of all the process, the speed of vehicle is displayed. If the speed of the vehicle is greater than the estimate limit then it will capture the image and it will store it in the folder.

#### 3.2 System Module

From fig 3.1, it can be observed that the system comprises of 5 main modules : 1.Road Tracking 2.Lane Segmentation 3.Image acquisition 4.Image subtraction 5. Offenders' detection. System flow can be represented as below.



**Figure 3: System Flow**

#### **4. EXPERIMENTAL SETUP**

This section includes all the elements used in the experiment (Multiple vehicle speed detection on camera feed) and the implementation of the proposed model.

##### **4.1 Experimental Methodology**

This subsection describes about the implementation process of each module of the system.

**4.1.1 Input speed and Video Acquisition:** Required speed limit is taken as input through `int(input('Enter The Speed Limit: '))`. We take video as input from the camera. OpenCV library plays a major role in obtaining and processing visuals that are captured simultaneously. OpenCV has the ability to process images and videos to identify different objects. There is no requirement to mention the resolution for the camera when OpenCV is used because the library itself opens the camera with a default resolution. The method of OpenCV we used is, `cv2.VideoCapture('test1.mp4')` and initialized it to `cap`.

**4.1.2 Road segmentation and image conversion:** After the video is acquired, our next step is to divide it into two lanes. For this we have taken two lane images through the method `cv2.imread("img.jpg")` and `cv2.imread("img1.jpg")` respectively. These images are converted from RGB to gray images using the method `cv2.cvtColor(mask2, cv2.COLOR_BGR2GRAY)` for both of the separately.

**4.1.3 Background subtraction:** We have done background subtraction using three different methods which are as follows:

- (a) `cv2.bgsegm.createBackgroundSubtractorMOG2()`- It is also a Gaussian Mixture-based Background/Foreground Segmentation Algorithm. The important feature of this algorithm is that it selects the exact number of gaussian distribution for each pixel. It detects and marks all the shadows in the image, but decreases the speed. Shadows will be marked in gray color.
- (b) `cv2.createBackgroundSubtractorMOG()`- It is a Gaussian Mixture-based Background/Foreground Segmentation Algorithm. It generally uses a method to model each background pixel by a mixture of K Gaussian distributions ( $K = 3$  to 5). The weights of the mixture generally represent the time proportions of those colors stay in the scene. The ones which stay longer and more static are the probable background colors.. While coding, we created a background object using the function, `cv.createBackgroundSubtractorMOG()`.
- (c) `cv2.bgsegm.createBackgroundSubtractorGMG()`- This algorithm combines statistical background image estimation and per-pixel Bayesian segmentation. It uses first 120 frames(by default) for background modelling. It provides probabilistic foreground segmentation algorithm that identifies possible foreground objects using Bayesian inference.

**4.1.4 Kernel Creation and Conversion :** We created a kernel using the method `cv2.getStructuringElement(cv2.MORPH_ELLIPSE, (3, 3))` in order to remove the noise in the foreground mask and converted the pixels under kernel to 1 and others to 0. This has been achieved through the method `np.ones((5, 1), np.uint8)`

**4.1.5 Capture video frames and their BGR reading:** We have captured the video frames from the video using the method `cap.read()` and BGR readings are captured through the method `cv2.split(frame)`.



**4.1.6 Contrast improvement:** We have used CLAHE which is a variant of Adaptive histogram equalization (AHE) and takes care of over-amplification of the contrast. CLAHE operates on small regions, called tiles, but not on the entire image at a time. The tiles which are neighbors to each other, are then combined using bilinear interpolation to remove the artificial boundaries. This algorithm improves the contrast of images. The method we used is `cv2.createCLAHE(clipLimit=2, tileGridSize=(1, 1))`.

**4.1.7 Defining boundaries where speed is captured:** We have drawn two lines at which speed of the vehicles will be displayed by the following methods `cv2.line(frame_og, (300, 513), (1900, 513), (0, 255, 0), 2)`, `cv2.line(frame_og, (300, 482), (1900, 482), (0, 255, 0), 2)`

**4.1.8 Other Image processing techniques:** `cv2.morphologyEx(foregroundMask, cv2.MORPH_OPEN, kernel)`- We used Morphological Opening to find things into which a specific structuring element can fit.  
`cv2.erode(foregroundMask, kernel, iterations=3)`- The basic idea of erosion is just like soil erosion only, it erodes away the boundaries of foreground object (Always try to keep foreground in white).. It helped us for removing small white noises.  
`cv2.morphologyEx(foregroundMask, cv2.MORPH_CLOSE, kernel, iterations=6)`- In mathematical morphology, the closing of a set (binary image) A by a structuring element B is the erosion of the dilation of that set, In image processing, closing is, together with opening, the basic workhorse of morphological noise removal. Opening removes small objects, while closing removes small holes.  
`cv2.dilate(foregroundMask, kernel_di, iterations=7)`- It is just opposite of erosion.. Generally, in cases like noise removal, first we do erosion and then dilation. Because, erosion removes white noises, but it also shrinks our object, we dilate it. Since noise is gone, they won't come back, but our object area increases. It helps us in joining broken parts of an object  
`cv2.medianBlur(foregroundMask, 5)`- The Median is an averaging method where the central element of the image is replaced by the median of all the pixels in the kernel area. This operation helps in processing the edges while removing the noise.  
`cv2.findContours(thresh1, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)`- The mode `cv2.RETR_TREE` helps in finding all the promising contour lines and reconstructs a complete hierarchy of nested contours.

**4.1.9 Frame division:** We detected the region of interest for vehicles and is shown as rectangle with a centroid in it by the following methods `cv2.rectangle(frame_og, (x - 10, y - 10), (x + w, y + h), (0, 255, 0), 2)`, `cv2.circle(frame_og, (cx, cy), 5, (0, 0, 255), -1)`

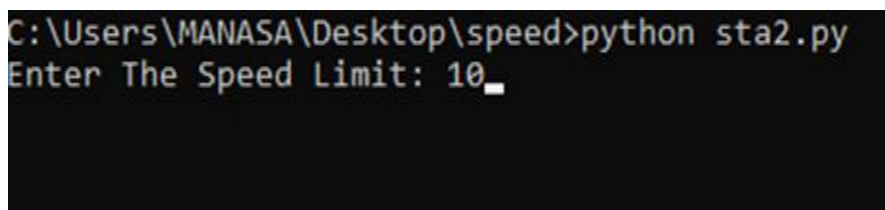
**4.1.10 Speed Calculation and display:** We calculated speed using the following formula  $speed1 = ((distance) / (36.6 * (seconds1))) * 3600 * 90$  and displayed it using `cv2.putText(frame_og, str(int(speed1)), (x1, y1), font, 2, (255, 255, 255), 8, cv2.LINE_AA)`. And the speed is calculated for both the lanes.

**4.1.11 Offenders detection:** Those vehicles which have crossed the given speed limit will be stored in respective folders called lane1 and lane 2. For this we have written `cv2.imwrite('Offenders/lane1/'+ 'Lane'+str(v)+''.jpeg', la)` for respective lanes.

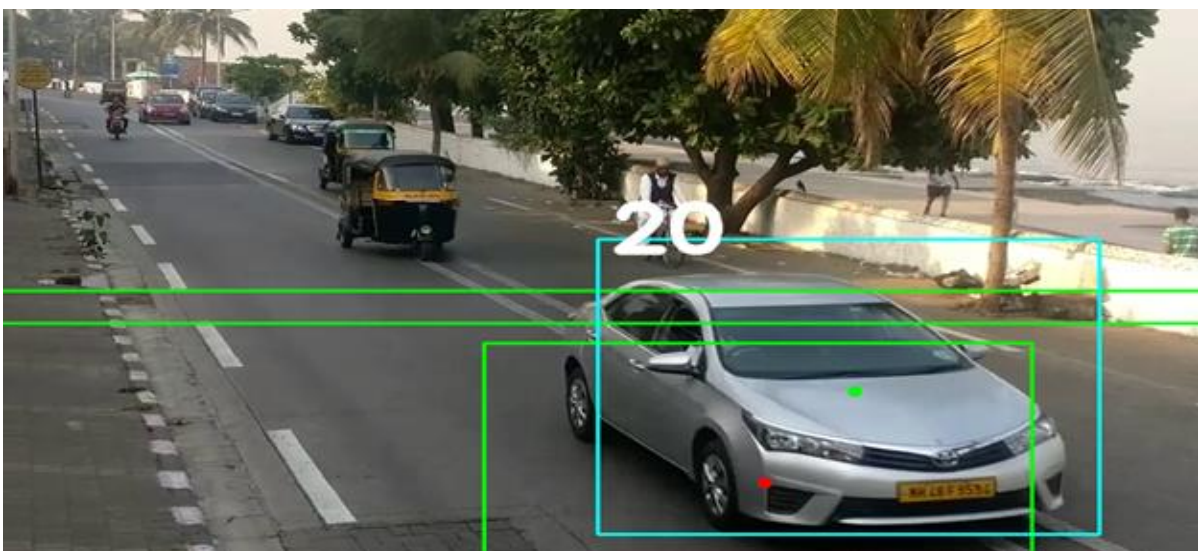
## 5. EXPERIMENTAL RESULTS

This section depicts the results obtained after conducting the experiment. We gave a certain speed limit, so that the images of those vehicles which cross the given limit will get captured and stored in the respective folders (Either in Lane1 or Lane 2)


### 5.1 Input



### 5.2 Speed detection






### 5.3 Storing over speeding vehicles



Offenders

Name	Date modified	Type	Size
lane1	26-06-2021 21:23	File folder	
lane2	21-06-2021 23:52	File folder	



## 6. CONCLUSION

Our designed speed detection system was capable of continuously monitoring the speed of the approaching vehicle. It worked well for all the vehicles. The output was more accurate with no other moving objects in the surrounding. The speed of each passing vehicle was displayed in the LCD display. The detected speeds were proportional to the ground truth speeds. This system is very useful to measure the low speeds accurately and without any expensive instruments. This helps in detecting those who violate traffic rules without any mere presence of traffic police. With a simple method of speed detection using a camera a useful software has been developed.

### 6.1 Future Scope

For the existing system, we can add other advanced features using Artificial Intelligence and Machine Learning. The speed detection using Open CV system is still a long way to go. The software system will not only be able to detect the vehicle speed but also will be able to capture the vehicle plate (number). We can use several Image processing techniques in order to get the clear view of the detected number plate. By this, penalty can be enforced on those who violates traffic rules. Considering the fact that importance of technology is increasing day by day, trustworthy speed detection systems can be deployed in few years.

## 7. REFERENCES

- [1]. Lee, S., Gwak, J., Jeon, M.: Vehicle Model Recognition in Video. International Journal of Signal Processing, Image Processing and Pattern Recognition.
- [2]. Yang, T., Kang, S.: Tracking for moving object
- [3]. J. Pelegri, J. Alberola, V. Llario, "Vehicle Detection and Car Speed Monitoring, The IEEE Annual Conference in the Industrial Electronics Society (IECON 02), pp. 1693-1695, November 2002.
- [4]. Z. Osman, S. Abou Chahine, "Speed Detection Scheme", The 14th International Conference on Microelectronics (ICM 2002), pp. 165-168, December 2002.
- [5]. S. Pumrin, D.J. Dailey, "Roadside Camera Motion Detection for Automated Speed Measurement", The IEEE 5th International Conference on Intelligent Transportation Systems, pp. 147-151, September 2002.
- [6]. <https://www.wikipedia.org>.
- [7]. <https://opencv.org>