

# Speed Detection Software

Himanshu Dwivedi  
Student,  
Computer Science and Engineering Department,  
BPIT Delhi

Arunim Garg  
Student,  
Computer Science and Engineering Department,  
BPIT Delhi

Sarthak Jain  
Student,  
Computer Science and Engineering Department,  
BPIT Delhi

Dr. Achal Kaushik  
Professor,  
Computer Science and Engineering Department,  
BPIT Delhi

## INTRODUCTION

In this research paper, we are going to introduce a very general problem which is a curse for all. The problem is related to the ongoing accidents phenomenon due to the carelessness and inability of humans in controlling the speed of vehicles[1]. Vehicles are one of the great inventions, we can say, which were introduced for the comfort of humans in travelling from one place to another. Due to this equipment only, it is possible for each human to go to very distant places in no time which was almost impossible before their existence. As a human nature, humans started to treat this auspicious gift not for their need but for their greed also. It also became a standard of respect among many of us to drive the vehicles out of controllable speed disregarding the harm it can cause to ourselves as well as to others. It doesn't cause the loss of only vehicles but also to properties, nature and even to human lives. Humans either by giving the priority to reach their destination on time or thinking to compete with others or show off to others they cross the limits of speed of their vehicles which could not be controlled by them at the instant any emergency occurs to stop the vehicle immediately.

To keep control on such nuisances by humans many rules and regulations came into existence all over the world varying from country to country. Separate departments of security forces were established with the only motive of keeping control on the speed of vehicles. Various categories of punishments as well as fines were decided on the breaking of the rules and regulations set by the government on drivers. Instead of all such efforts of government at different levels the problem could not be made to come to any appreciable level of comfort due to the various reasons like improper implementation of laws as well as inability of humans to detect overspeeding of vehicles on time.

To overcome this inability of humans to detect overspeeding of vehicles on time some hardwares were designed like RADAR and LIDAR called Doppler Shift Phenomenon[3][9][10]. These devices were allotted to authorities on duties especially so that they can estimate the bypassing vehicles in front of them without much delay and thus can take appropriate actions on time. Instead of such a good effort there is a big flaw that these devices are also dependent on humans for their working. Thus for example,

it may happen that humans can't start recording the speed of vehicles at the instant it came into existence and may not stop the recording at the same instant the vehicle passed away. Thus due to the delay or hurry introduced by humans may result in inappropriate data being collected.

Now there was the need to give full control to a software which is more reliable than humans. Many investigations and researches were started more than 10 years ago. The idea was to develop an algorithm to introduce the capability of controlling the speed of vehicles by the software itself which must not be dependent on humans for their working. At such a moment Machine Learning (more specifically Deep Learning) was implemented to achieve the specified target. First of all many hardwares with the required functionality being built in the same were introduced like RaspberryPi[4] and arduino 3[9][8][14]. Such devices are somewhat explained later in this research paper[1][4][6][7][8][9][14]. Then the problem arose of cost and material for such components and hardwares. Then some research was also done to get rid of such hardwares to make the service available to each person. Time to time many researches were done to introduce a new algorithm better than the previous one or to introduce changes in the existing algorithm to improve it more and remove some of its flaws. Many existing modules were incorporated in the new algorithms defined like openCV[1][14][12], KNN[12], RCNN[11][14], Haar Classifier [13]etc. In such algorithms the thing which was implemented to detect the speed of objects as a machine learning concept are mainly image processing of the videos being recorded. The rest is mainly normal speed calculation based on the distance covered and the time taken which can be taken constants in configuration to be already defined while designing the algorithm and implementation varying from place to place.

Now we are also about to take a step forward in these efforts done by various persons at different times. At this time our idea is to develop an algorithm which consists of calculating the speed of objects specifically vehicles instantaneously by dividing the video into different frames and calculating the result between every two consecutive frames. Most importantly we want to observe the behaviour of the object at the instants when it enters or leaves the region of camera. Alongwith we are also going to mix up different algorithms mentioned in papers into a single algorithm for detecting objects basically to compare

each one to a different one also to make each one to improve the other one.

## LITERATURE REVIEW

There are following research papers that have been published in the recent past:

[1] In this research paper the object is detected using the Caffe model for DNN while simultaneously the distance of obstacle is measured with the help of ultrasonic sensor and according to the distance of obstacle the motor movements are controlled with the help of PWM controller. Caffe is a deep learning framework developed by BAIR. If any obstacle is detected in that frame then the sensor will calculate the distance of the vehicle from the obstacle. The corresponding frames are converted into blob which is compared with the pre-trained model. If the distance calculated is less than or equals to 30cm then a low signal is sent and the vehicle is stopped.

[2] In this research paper the moving objects are detected using Gaussian Mixture Model, DBSCAN clustering and bounded in a box to keep track till the time it is visible in the camera. Then the object is tracked by applying Kalman Filter and Optical Flow method. Both these methods communicating with each other give rise to the Euclidean Manager concept applied in the paper. Then the speed of the car is calculating by calculating the moments of its pixels. The video is also converted to gray scale image as it is being mentioned as a noise in the research paper while detecting them. The algorithm is tested on cars with the speed of 15km/hr and 20 km/hr.

[3] In this research paper, the camera was calibrated based on geometrical equations. The algorithm needs only a single video camera and a Cor2Duo computer processor with Matlab software which is installed. The software system is composed of 6 subsystems namely, the camera calibration unit, the background update and removal unit, the vehicle detection unit, the speed measurement unit, the result analysis unit and the outstanding reference. Camera is set at a certain height above on a freeway. For extraction of foreground and background a method called combination of saturation and value or CVS is applied. The speed of the vehicle in each frame is calculated using the position of the vehicle in each frame. The blob centroid determines the distance of the vehicle moving in consecutive frames and therefore as the frame rate of captured moves is known and speed is calculated. The average error of the detected vehicle speed was  $\pm 7$  km/h for speed below 50km/h.

[4] This research paper is written for the purpose of analysing the performance of Raspberry Pi 2 in detecting the speed of objects, specifically vehicles. In this project the complete description of Raspberry Pi 2 is given that it is furnished with two USB 2.0 ports which are joined with LAN9512 combo center point/Ethernet Chip IC3, which is itself, a USB Gadget associated with the single upstream USB port on BCM2835. On the software side Raspbian OS

is used with OpenCV-python installed in it. On the algorithm part, after initializing the system the video is read first and then color conversion is applied on it to remove the RGB color considered as noise while detecting the pixel of the object. Then in each frame of the video the object is detected and corresponding speed is calculated at last. In the result section the performance of Raspberry Pi 2 is described as 320p, 540p, 720p captured images give almost the same result with the only difference in memory usage to some limit. It is concluded that it uses 25% of its CPU usage and 600MB/1000MB of memory to run the system successfully.

[5] In this research paper two techniques linear and discrete motion speed detection were employed. In the first, the vehicle speed was calculated as the ratio of real distance covered by the field of view (FOV) and the time duration between the vehicle entering and exiting the FOV. Time duration is determined as the time between the time stamps at the beginning and at the end of the FOV. In the second, vehicle speed is calculated at different time stamps within the FOV with respect to initial startup time stamp. Vertical distance between the camera and the vehicle and the vehicle travelling distance in the FOV are determined by using trigonometry. The error in speed compared to speedometer was  $\pm 1.2$ m/sec

[6] In this research paper, the system consists of vehicle detection nodes, the master node, and upper computer, the vehicle detection node mainly uses STM8L SCM with the geomagnetic sensor and the 433MHz wireless transceiver module. In this a new type of geomagnetic sensor (AMR) is selected to design a wireless vehicle detection system. The geomagnetic sensor is used because it can determine the presence of the vehicle in the field based on the change detected magnetic strength. Time is noted when a vehicle passes 2 nodes. final speed is calculated when distance is divided by the time difference. Test results show that the system possesses small size, low cost and high performance, and can actually be applied to outdoor parking spaces detection.

[7] In this research paper a proposal has been made to improve the existing car's speed detection system. In this project hardware as well as software are used. Camera quality has been given special privilege to detect the objects even in dim light. Storage scope has also been added with a network card so that if at any time the network gets lost while the system is being operated then after regaining the signal the data is again gathered from the storage and the process further continues from the point it was stopped. OCR to detect the VIN of a car has also been added in this project in case the speed gets out of the specified limit. For the storage purpose compression has also been applied on images and a web server is also being added to access the data remotely.

[8] In this research paper, the central and intelligent unit of the model is arduino. It includes the programming of Arduino according to layout whenever there is an over speed vehicle which cross the range of specific fixed RPM of that road then an alert in the form of message is send to the

controlling authorities so that they can take necessary action. When the vehicle starts, IR sensor detects the speed and microcontroller on the Arduino Nano processes the data and it will notify GSM module when the vehicle overspeeds to send an alert message to the concerned authority, By referring to both figures, the complete program can be constructed later in Arduino IDE software

[9] In this research paper the vehicle detection system has been implemented with Arduino. In this implementation Doppler shift phenomena has been implemented. In this phenomena electromagnetic waves have been sent and the response of the reflection of those waves is being recorded with corresponding timestamps. In the complete circuitry voltage regulator and amplifier have also been included to strengthen the waves. On getting and converting the speed of objects, specifically vehicles in MPH, it is checked if the speed is within the given specified limit or not. If any vehicle with speed greater than the given speed limit is detected then the message with the speed is displayed in LCD.

[10] In this research paper IOT based framework has designed an algorithm to detect the object and estimate its speed in which the object is tracked using Google Maps and GPS Module by sending its coordinates to the server which helps in determining the position of the object. Then the speed is calculated using a Radar based system.

[11] In this research paper CNN is applied to detect and track objects with good accuracy. In this work the video surveillance system is implemented which helps in providing much more information like Vehicle plate number, location, speed, classification and status of vehicles. Different images of the same area are processed one by one to identify the background. Thus in this way the vehicles are separated by implemented edge detection and finally vanishing point. Then only these areas of interest are being processed for speed estimation. The classification is useful to classify or identify five objects - Car, SUV, Bus, MotorCycle and Truck. The final model is tested with 80% data and the rest is validated based on test performed.

[12] In this research paper the object is tracked using KNN and openCV to estimate the speed of the vehicle. All the objects appearing in the scene are bound in a box. If the object is not detected accurately then it is bound in a red box after which it is converted into a green one. All the pixels of the object detected is recorded and based on the change in centroid location in different frames of the video the object is tracked. Here also the concept of white and black image is considered to calculate the centroid of the vehicle.

[13] In this proposal the concept of HAAR classifier is implemented for detection and speed estimation of the objects. In this concept the images are divided into positive and negative images. For the testing purpose the 2000 negative images and 1500 positive images are considered for testing purposes. This concept bounds the image in a box, calculates the sum of intensities of pixels in different boxes and sends the difference between them. Basically 3 Haar

like features have classified to detect edges, to detect lines and four rectangle features to detect slanted lines. After speed estimation performance matrix is used to calculate experiment results between actual positive, actual negative, predicted positive and predicted negative.

[14] In this research paper the combined use of Arduino Uno, OpenCV and R-CNN is proposed. In this method static frames are analysed and their pixels are differentiated from each other thus giving rise to four different methods for foreground detection basis on comparison with threshold value or adaptive background techniques. Fifth method of RCNN of dividing an image into 2000 regions and designing a Convolution network corresponding to each image is also defined and thus compare to other above defined methods.

[15] In this research paper all the proposed methods of speed estimation and object tracking have been classified into Background Subtraction Methods, Feature Based Methods and Frame Differencing and Motion Based Methods. Then different camera calibration methods have also been defined like Region-Based Tracking Methods, Contour Tracking Methods, 3D Model-Based Tracking Methods, Feature Based Tracking Methods and Color and Pattern-Based Methods.

#### PROBLEM STATEMENT

In most of the recent papers mentioned above there is much research done in order to improve the detection of object specifically vehicles. Many algorithms designed and improved for this purpose but when the matters come at speed calculation normally the concept what is applied is that through the complete coverage area in the view of camera or software only the average speed for that area or space is calculated at once. Thus it gives the idea that with what speed the car has passed the region coming into the view of the camera installed at various places. Now almost every person in the world knows about this effort but still they try to fool even this great technology by manipulating vehicle's speed at such places where they come under the view of the camera. Thus there is an urgent need to modify the algorithm such that it can analyse the complete activity of the driver. By activity it means that either the driver tries to increase the car's speed at different instants of time, or decrease the same at different instants of time or makes it constant. It is also possible that in the region covered under the view of the camera the driver keeps the speed below the threshold or safe value but changes abruptly at the starting or end of the view.

Thus keeping in mind the above problem we need to modify the algorithm in such a way that the speed estimation or detection is not average for the whole region but it detects the speed between every two consecutive video frames especially at the starting and end of the video frame where there are great chances of the driver to change vehicle's speed abruptly trying to fool the complete system.

Second thing which needs some attention and focus is that through a simple algorithm only the whole process either seems very complicated to proceed or unreliable to implement thus there is also a trial in this research paper to

extract the special functionality of some predefined algorithms respectively and mess them up in a single one to make it convenient to understand and also reliable to understand.

#### PROPOSED WORK

The basic idea of this proposal is to calculate the speed of vehicles in video with respect to frame. At each frame we will detect the position of the object and track it by comparing it to the position of the same in the previous frame. At the same time we will calculate the speed of the vehicle by estimating the rate of change of position in two corresponding frames.

First we will classify the video into different frames or I can say images. On each of those frames treated as images we will apply our complete processing of the algorithm. First we will convert the images into grayscale images i.e., we will remove the color property from the pixels of the images or video because this is considered as noise while processing any of their features.

Then we will use both RCNN and Fast RCNN algorithms one by one to see the difference in the output produced by the algorithms in our model. In Fast RCNN the complete image is processed at one go to detect the objects and create a convolution network map for the complete image. Then in the SVM classifier the detected objects are classified into different categories. In RCNN (Region Convolution Network) we divided each image into different regions to process separately. Thus according to the mentioned algorithm we limit the number of regions to 2000. The 2000 regions to be processed are selected on the basis that they have significance to be processed for the detection of objects based on certain criteria such as two separate regions should not be much similar to each other but they should show some difference to each other. Then each region is processed in a convolution neural network. This feature has certain pros alongwith the cons such that here each region is needed to be processed separately from each other and the processing of each region takes about 47 seconds thus the processing of 2000 separate regions will take approximately 94,000 seconds which is an appreciable amount of time. Thus to overcome this limitation the same author has introduced a new version of the algorithm called Faster RCNN (Region Convolutional Neural Network) in which instead of different regions of the image the complete image is processed itself at one go to generate a Convolution Neural Network Map with different objects being detected separated from the background. Thus we will apply the selected algorithms out of two on the first generated frame and treat it as out training data or frame for the rest of the generated frame.

Once the image has been detected and classified we will try to improve the image for better processing. Here comes the Gaussian Mixture Model which is used in unsupervised learning to improve the boundary region of the image being classified. In this model first the image has to be divided into the background and the different objects being detected. This work has already been done above with the help of the RCNN algorithm. Then we will analyse or process each pixel of the image under consideration. We will record the position of the pixel and analyse or try to calculate

the probability that to what extent the given region belongs to that region of image in which it has been classified or detected in the above algorithms i.e., in short to what extent our classification of different regions as background or different objects is accurate. Thus accordingly we will modify our regions of background or different classified objects. Corresponding to the calculation of the probability of a pixel belonging to the classified region the probability distribution graph or normal graph will also be designed to determine which will help in determining or modifying the accuracy in the detection of the objects in the image or video frames in the previous steps.

Once the image has been defined accurately within the defined boundaries using the Gaussian Mixture Model then we will apply the same result on the rest of the frames or images to extract the same objects in those frames also. Here the strategy of HAAR Cascade Classifier is applied which says to detect the object in the training set of data and then apply the result to the rest of the data which acts as testing data. In the training data the positive and negative images i.e, the object and the background are processed and the result is produced to process the rest of the frames with the same results to classify them as positive or negative images to detect the objects out of the given frames or images we can say.

Once the images have been completely processed according to the above mentioned algorithm and the objects have been extracted in each of them, the centroid of each object will be determined by calculating its coordinates using the coordinates of the pixels under the object being classified using the Gaussian Mixture Model above. For such purpose we will employ OpenCV in which the centroid of a blob (an object of irregular shape) can be calculated using the method given in the reference [16].

Now comes the role of a storage system in which the coordinates of coordinates being detected in each frame will be recorded with the corresponding timestamp in which the complete processing will take place or I can say the timestamp in which the centroid of the object in that frame has been determined. Now with the recorded data we have the position of the centroid of objects correspondingly the position of the objects as a whole recorded with the timestamp in our records for the different frames of video being processed. Thus taking any two frames under consideration we can determine the rate of change of position of the object with reference to the difference of the timestamp being recorded. Thus analysing the speed between every two consecutive frames we can prepare the complete record during the period in which the driver or vehicle was under the vision of camera or software. During this whole period we need to concentrate on the timestamp in which the driver enters the region under the camera or software vision as well as the timestamp in which it leaves the timestamp to analyse the actual behaviour of the driver being recorded to determine if he was actually driving with the recorded speed or was trying to fool of the complete software.

The project can later be extended further to train our system to actually determine the behaviour of driver during the time period during which the video was

recorded. In this way applying the training and testing model we can make the system so perfect that without the need of humans the system will itself be able to determine the actual speed of the driver not only during the period it would be in front of the camera or software but during the complete journey of the driver

**EXAMPLE**

1. First we will divide the complete video into a varying number of frames limiting their number keeping in mind that the change in frames' structure is significant in terms of object present.

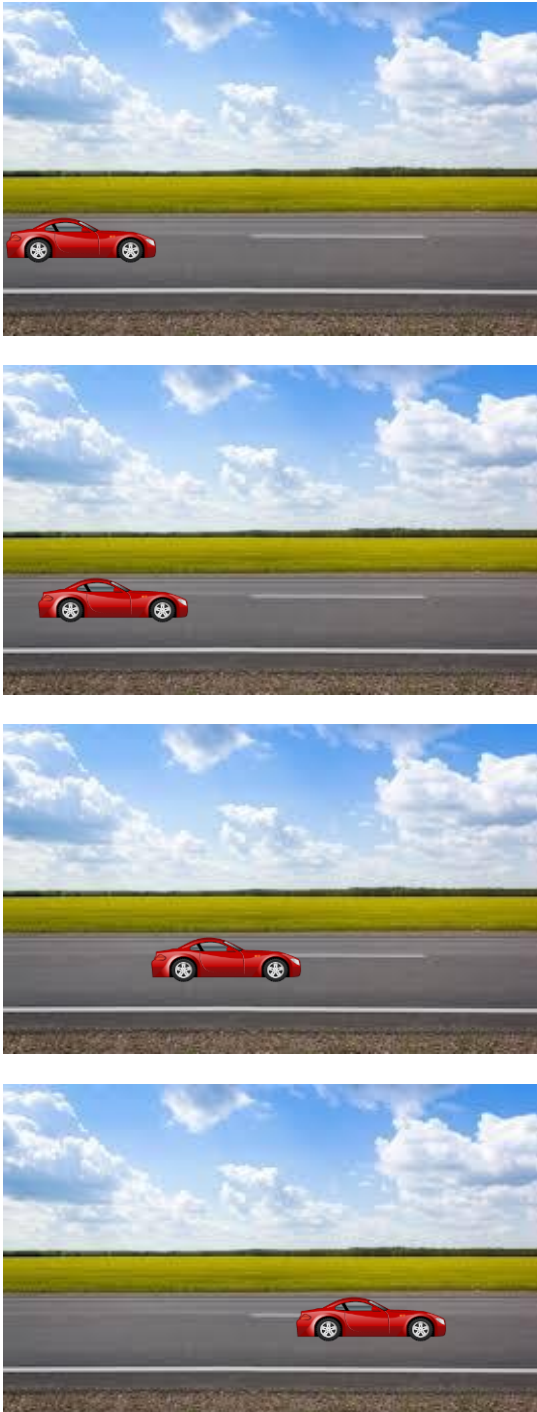


Fig 1. Dividing Video into frames

2. Then we will convert the complete video or images into the grayscale images to remove the color present in them and act as a noise in the detection of objects especially in terms of their pixels.



Fig 2. Converting image to grayscale

3. Now comes our R-CNN algorithm (or Faster RCNN based on future testing that one gives better results) to classify the object from the first frame which will act as training data for the classification of the same object in the rest of the frames later. On the detecting region we will apply SVM classifier to detect the category of objects as a car, human, animal or anything else.

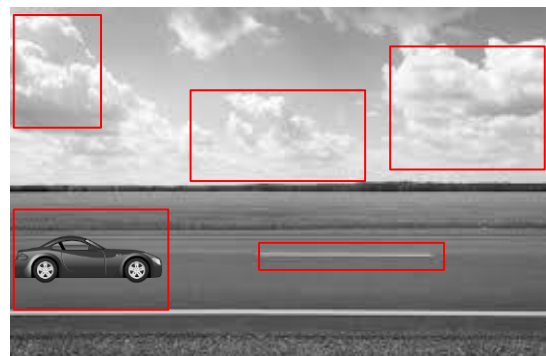


Fig 3. Detect object in image with RCNN

4. Now after the object is detected and classified the background extraction is applied in the manner that the region except the object detected is treated as background and called as black image and the object classified is treated as white image.



Fig 4.. Extracting object from image using background extraction

5. Then once the object region is classified in each frame we will apply GMM (Gaussian Mixture Model ) to all the detected regions of objects to accurately define the boundary of the object detected in each frame.



Fig 5.. Apply Gaussian Model Mixture to accurately define the boundary of the object being detected

6. Then as the HAAR classifier cascade says we can apply our training data to the rest of the frames to classify the same detected object in them.

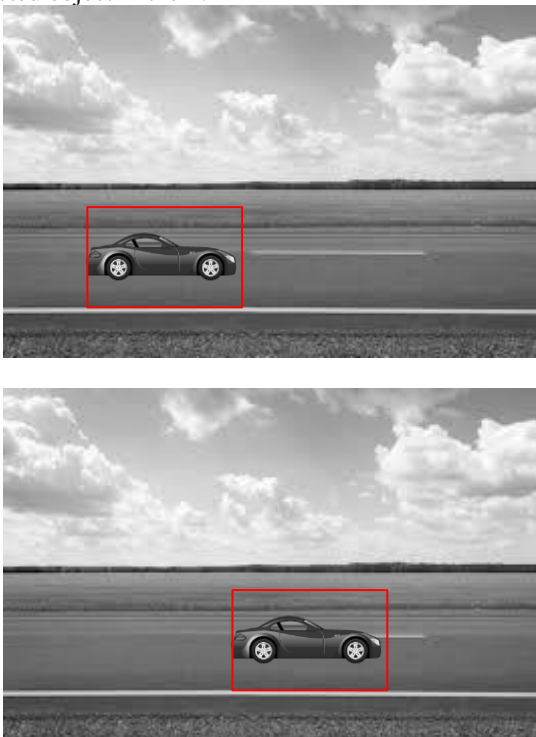


Fig 6. Apply the concept of HAAR cascade classifier to apply the result of first frame to the rest of the frames

7. Then after this once the object is completely defined in each frame or image we will record the position of all the boundary pixels of the object detected and then correspondingly calculate the centroid of these pixels in each frame or image and record the position of the same with respect to the position of the complete object in each frame



Fig 7. Calculate the coordinates of the centroid with the pixels' coordinates under the region of object being classified

8. Now comes the role of database or file whatever convenient to use to record the timestamp and position of the object in each frame. We will try to connect the database to our LAN system so that the data will always be saved with us on regaining the connection if any network fails. Now as we data of object' centroid location with the timestamp recorded.

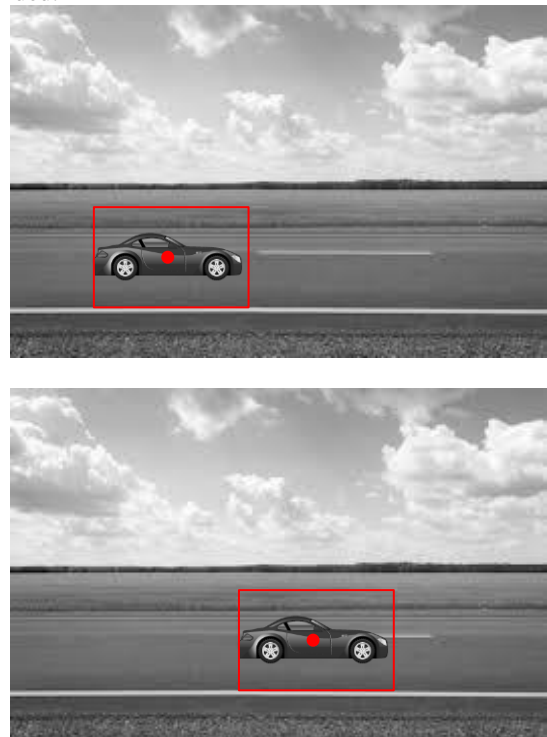


Fig 8. Calculate centroid of object in each frame and record the corresponding result with timestamp for each frame

9. We can now calculate the difference in position of an object's centroid or we can say the object as a whole of which we are taking the centroid as reference and calculate their difference. Then taking it with respect to time we can approximate the change in speed at different instants of time as if the speed is increasing or decreasing during different instants of time.

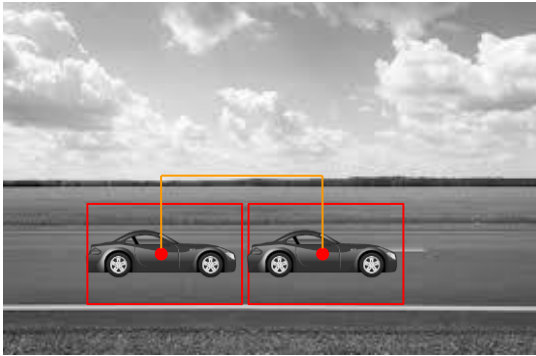


Fig 9. Calculate the rate of change of centroid or object with respect to time between any two frames

In this way we can record or observe the behaviour of the driver during the whole period of time it is under the vision of the camera or software. During this period our main focus will be on the instants when the objects are entering or leaving the area of region under the camera's software's eye so that it can predict if the driver was genuine in the image or video being recorded if he/she was trying to make fool of the system. Thus in the future this proposal can be extended further to more accurately detect and observe the behaviour of the driver and whatever object is moving to predict its actual activities with respect to speed with which it is actually moving.

**COMPARISON TABLE**

Earlier Proposed	Proposing By Us
1. In the existing research mainly the average speed has been directly calculated by noticing the time period for which the object remains in the vision of the camera.	In our proposal we are going to propose to calculate instantaneous speed of vehicles by recording the position of objects in each frame and then calculating the rate of change in their position between consecutive frames.
2. In the previous research different algorithms are introduced keeping one aspect and trying to modify their accuracy like RCCN then fast RCNN.	In our proposal we are introducing to test all the different aspects absent in one algorithm but present in another one which can be used to test the previous one.
3. Only those facts are considered while detecting the object which the object shows to the software by inspecting the time period for which the object is visible at one go.	We are proposing to give special attention to the region where the driver enters or leaves the region under the camera's view so that some change in the object speed can be predicted for better determining its actual speed.

**MATHEMATICAL FORMULATION**

1. As already specified that first the object will be detected in the RCNN model using SVM classifier after which the accuracy of the boundaries of the object will be accurately defined by GMM with some equations defined below:

**Gaussian Mixture Model[1]**

At any time t the pixel history (x0, y0) is determined as:

$$X1.....Xt = \{V(x0, y0, i): 1 \leq i \leq t\}$$

This is based on K Gaussian Mixture Model where K is dependent on Computer Performance: The probability of observing current pixel Value is given by :

$$P(Xt) = \sum k \omega_i, t \eta(Xt \mu_i, t, \Sigma_i, t)$$

Where  $\omega_i, t$  = an estimate of the weight of ith Gaussian mixture at time t

$\mu_i, t$  = mean value

$\Sigma$  = covariance matrix

$\eta$  = Gaussian Probability Density

Function

( $\eta$  is given as

$$\eta(Xt \mu_i, t, \Sigma_i, t) = \frac{1}{(2/\pi)^{n/2} \Sigma_i^{0.5}} \exp - \frac{1}{2} (Xt - \mu_i, t) \Sigma_i, t (Xt - \mu_i, t)$$

Where  $\Sigma_k, t = \sigma^2 t I$

From this formula the probability of a pixel belonging to the classified region is determined and corresponding graphs will be plotted.

2. After accurately determining the boundaries we will apply the algorithm of HAAR Cascade Classifier to the rest of the frames. After applying the same we will determine the precision and accuracy of our object detection using the formulas defined in HAAR Cascade classifier mentioned below:

**Haar Cascade Classifier[2]**

As in the haar cascade classifier we are implementing the result of first image on the rest of the images so for that to determine the accuracy here are some predefined formulas

TP = True Positive(Data Points correctly classified as belongs to the object)

TN = True Negative(Data Points correctly classified not belonging to the object)

FP = False Positive(Data points incorrectly classified as belongs to the object)





vision of the camera or trying to make a fool of the system. Thus training the system in this manner for some consecutive years the system could be developed to take its own decision and record the data corresponding to different drivers or vehicles being passed regarding the actual or approximate speed with which they would be driven. The system till some time can be fed with the data being recorded treated as training data on which it would be made to carry on complete analysis to extract the features or characteristics of the driver each time being recorded or analysed with its corresponding speed and then the same data being developed would be applied in the future recorded videos which will be treated as testing data. In this way, at that time the system will require no intervention of humans. With its own intelligence it will be able to determine or prepare the complete database with the information corresponding to each driver it will make an analysis and then can send instant signal or any notification if any vehicle being captured or predicted as getting overspeeded. Due to such a capability of the software the hassle to install unlimited cameras or softwares at different locations in any specific area will be reduced to install a few only at required locations thus saving money and efforts.

#### REFERENCES

- [1] Collision Avoidance Based on Obstacle Detection Using OpenCV by DR. S.S. Lokhande, Sonal Darade, Pranjali Deshmukh, Neha Joshi, Dept. Of Electronics and Communication, Sinhgad College Of Engineering, Pune, India
- [2] Vehicle Speed Detection from Camera Stream Using Image Processing Methods by Jozef Gerat, Dominik Sopiak, Milos Oravec, Jarmila Pavlovicova, Faculty of Electrical Engineering and Information Technology/Slovak University Of Technology in Bratislava, Ilkovicova 3, 812 19, Bratislava, Slovak Republic
- [3] Vehicle Speed Detection in video image sequences using CVS method, Arash Gholami Rad, Department of civil engineering, University of Malaya. 50603 Kuala Lumpur, Malaysia, Abbas Dehghani Department of Electrical Engineering, Sadjad Institute of Higher Education, Mashhad, Iran, and Mohamed Rehan Karim Department of Computer Science, UCLA, 420 Westwood Plaza, Los Angeles, CA 90095, USA
- [4] Vehicle Size Comparison for Embedded Vehicle Speed Detection & Travel Time Estimation System by Using Raspberry Pi, I.Was Zaidy, A.Alias, R.Ngadiran, R.B.Ahmad, M.I.Jais, D.Shuhaizar affiliations Embedded Network and Advance Computing (ENAC) School of Computer and Communication Engineering Universiti Malaysia Perlis Pauh Putra, Perlis, Malaysia
- [5] Vehicle Speed Detection using Camera and Image Processing software, Hakan Koyuncu Computer Engineering Department, Istanbul Gelisim University, Baki Koyuncu, Electrical and Electronics Engineering Dept., Istanbul Gelisim University Corresponding Author; Hakan Koyuncu
- [6] Design Of Vehicle Detection System based on Magnetic Sensor, Yang Xu and Xiaorong Zhou, School Of Automation, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
- [7] Enhancement Of Vehicle Speed Detection System for Avoidance of Accident, Abhrajit Chattopadhyay, Gunda Ravichandra, Vivek, Sudhanshu Solankim, P Suganya SRM Institute Of Science and Technology, Chennai, Tamil Nadu
- [8] A Compact-Low Cost, Intelligent Vehicle over Speed detection and Reporting System Employing Arduino And GSM M-Tech, UCIM, Panjab University, Chandigarh, India, Gurpreet Singh, Poonam Kumari, Assistant Professor, UCIM Panjab University, Chandigarh, India, HPS Kang Associate professor, UCIM, Punjab University, Chandigarh, India
- [9] Implementation of Doppler Radar-Based Vehicle Speed Detection System, May Zin Tun Lecturer, Department of Electronic Engineering, Technological University, Mandalay, Myanmar, Kay Thwe Zin Lecturer, Department of Electrical Engineering, Technological University, Mawlamyine, Myanmar
- [10] IoT based framework for vehicle overspeed detection, Mohammad Ahmar Khan, Sarfraz Fayaz Khan, Assistant Professor, Department of Management Information Systems, College of Commerce and Business Administration, Dhofar University, Salalah, Sultanate Of Oman
- [11] Multiple Vehicle Tracking and Classification System with a Convolution Neural Network, HungJun Kim
- [12] Real Time Detection Of Vehicle Speed Based on Video Image, Genyuan Cheng, Tangshan Municipal Transportation Bureau, Hebei Province Tangshan, 063000 China, Yubin Guo School of Traffic and Transportation, Beijing Jiaotong university, Beijing, 100044, China, Xiaochun Cheng, Beijing National Day School, Beijing, 100039, China, Dongliang Wang, Tangshan No 1 Middle School, Hebei Province Tangshan, 063000, China, Jiandong Zhao, Key Laboratory of Transport Industry of Big Data Application Technologies for Comprehensive Transport, School of Traffic and Transportation, Beijing Jiaotong University, Beijing, 100044 China
- [13] Detection Of Traffic Violations using Moving Object and Speed Detection, Neha Prakash, Chirag Uday Kamath Student, Gururaja H S Assistant Professor, Department of Information Science and Engineering, B.M.S. College of Engineering, Bengaluru, India (Visvesvaraya Technological University, Bengaluru, Karnataka, India)
- [14] Campus Traffic Reinforcement using IOT and OpenCV, Raynal D' Cunha, Department of Computer Engineering, Don Bosco Institute Of Technology, Mumbai India, Bebetto Francis, Department Of Computer Engineering, Don Bosco Institute Of Technology, Mumbai India, Richard Britto Department of Computer Engineering Don Bosco Institute of Technology, Mumbai India, Phiroj Shaikh Department of Computer Engineering Don Bosco Institute Of Technology, Mumbai India
- [15] Vehicle Detection And Tracking Techniques: A Concise Review, Raad Ahmed Hadi Department of Computer Science, University Technology Malaysia, Skudai, Malaysia, Ghazali Sulong Department of Software and Networking Engineering, College of Engineering, Iraqi University, Baghdad, Iraq, Loay Edwar George, Department of Computer Science, College of Science, Baghdad University, Baghdad Iraq