



**INT 247**

**CA-1**

**ON**

## **Vehicle Detection and Speed Tracking**

Under the Guidance of-  
**Upinder Kaur**

Submitted by-  
**Rishikesh Singh**  
(11902664)

# INTRODUCTION

Detection of vehicle and tracking of speed is the crucial part of town planning. In the last decade, vision-based traffic monitoring system has received considerable attention. This can be done with the help of vehicle detection and speed monitoring. The monitoring system gives various information about, vehicle count, traffic congestion and speed of the vehicle. One of the root cause of road accidents is speed. Extracting frames from the video and comparing the speed between two given points can be used to determine whether the car is moving above the permissible limit or not. There are many algorithms available for extraction of vehicles from the background. Traditionally radar systems were used for such applications but had some limitations. So to overcome the limitations in existing methods, various techniques have been developed for vehicle speed determination using image processing.[7] But the main factors that would affect these image processing algorithms is, waving of tree branches, camera noise, illuminations. The goal of this current research is to develop an automatic vehicle counting system along with the detection of speed, which can process videos recorded from stationary cameras over roads e.g. CCTV cameras installed near traffic intersections / junctions and counting the number of vehicles passing a spot in a particular time for further collection of vehicle / traffic data.

Vehicle Speed surveillance is a predominant factor in enforcing traffic laws. Traditionally vehicle speed surveillance was done using a radar technology which consists of a radar gun and a radar detector. Radar is an acronym that stands for Radio Detection and Ranging. Radar systems create radio waves, a form of electromagnetic energy that can be directed out into the air where the signals produced travel at the speed of light – roughly 186,000 miles per second, or  $3.08 \times 10^8$  meters per second. The transmission of these signals and the collection of returned energy that bounce off of objects in the path of the radar's transmission (called returned pulses) is what allows radar to be used to detect objects and range them, meaning establish their position and distance relative to the radar system's location. When a radar is used to detect the speed of an object (for example, when a police officer with a stationary radar gun is detecting the rate at which a car is moving), it does so by taking advantage of a phenomenon that occurs whereby the frequency of the radio wave for the return signal is altered because of the car's motion relative to the radar. If the car is moving toward the radar device, the return signal radio wave frequency increases. The radar gun can then use this change in frequency to determine the speed at which the car is moving. This principle, which establishes that the difference between the frequency of the emitted pulse and the frequency of the return pulse varies with the relative motion of the source to the object, is called the Doppler effect. So, while the distance of an object can be established by the amount of time that it takes to detect the return pulse, the speed of an object can be detected by establishing the change in the pulse characteristics between transmitted and received echo. This provides a velocity along the direction in which the radar is pointing, termed the radial velocity. One point to note is that the pulse characteristic changes used to establish the speed of a moving object like a car will depend on the relative position of the car to the radar. The measured speed will be accurate if the car moves directly towards the radar. But if the car's motion is at an angle relative to the radar gun's line of sight, the speed being measured will be a component of the actual speed of the vehicle. This principle is known as the cosine error effect. Because of these errors, the United States of America's law keeps a buffer of 8km which is caused because of the above-mentioned error. Also, radar technology can track one vehicle at a time. The paper deals with vehicle detection and speed tracking which is explained further.

## 1) Feature based modeling

This technique of identifying image displacements which are easiest to interpret is feature based modeling. The technique helps in identifying edges, corners and other structures in an image which are restricted properly in a two-dimensional plane and trace these objects as they transit between multiple frames. This technique comprises of two stages; finding the features in multiple

images and matching these features between the frames:

Stage 1: In this step, the features are found in a series of two or more images. If carried out perfectly, with no overhead cost; it may work efficiently with less overload and reduce the extraneous information to be processed

Stage 2: Features found in stage 1 are matched between the frames. Under most common scenarios, two frames are used and two sets of features are matched to a resultant single set of motion vectors. These features in one frame are used as seed points which use other techniques to determine the flow.

Despite this, both these stages of feature-based modelling possess drawbacks. In the stage of detecting features, it is necessary that features are located with precision and good reliability. This is proved to be of immense significance and research is performed on feature detectors. This feature holds an ambiguity of possible matches to occur as well; unless it is priorly known that image displacement less than the distance between features. [3]

### 1) **Frame Differencing and motion-based methods:**

Frame differencing is a method of finding the difference between two consecutive images from a sequence of images to segregate the moving object (vehicle) from the background. If there is a change in pixel values, it implies that there was a change in position in the two image frames. The motion rectification step of detecting a vehicle in a trail of images by alienating the moving objects, also known as blobs based on its speed, movement and orientation.

It is recommended to use an intraframe, interframe and tracking levels as frameworks to identify and control the motion of vehicles in frame. Using quantitative evaluation this paper illustrated that interframe and intraframe can be used to control and handle partially detected images and tracking level can be used to handle full blocked images efficiently. An approach to calculate frame difference can be given as: Difference between two consecutive frames

### **Speed Determination:**

The vehicles with a particular Id is observed for a series of sequential frames. The number of frames in which the car appears is noted.

Total Frames Covered= frame n – frame 0

Where, frame 0 is the first frame when object is entered in Region of interest and frame n is last frame when object passed away from Region of interest and the real-world distance is mapped on the image. The count of total number of frames is then multiplied with duration of one frame which is calculated from frame rate of video. Similarly, total

time taken by vehicle to travel and distance is fixed and is mapped from real-world into image. The thresholding phase. Opening closing dilations are performed where opening and closing is used to remove the detected foreground holes. Dilation consists of interaction of structuring element and foreground pixels. The structuring element is a small binary image.[10]After the whole process, the selected object pixels are applied for connected component analysis. Connected component is applied to binary and grey scale image and analysis is used to identify connected pixel region by scanning an image pixel-by-pixel. It has various connectivity i.e.8-pixel connectivity or 4-pixel connectivity.

This approach is not effective in windy conditions as this technique detects motion caused by movement in air. The possibility of camera not remaining fixed in its position due to air cannot be neglected which results in motion and formation of holes in the binary image.

## A. Gaussian Mixture model:

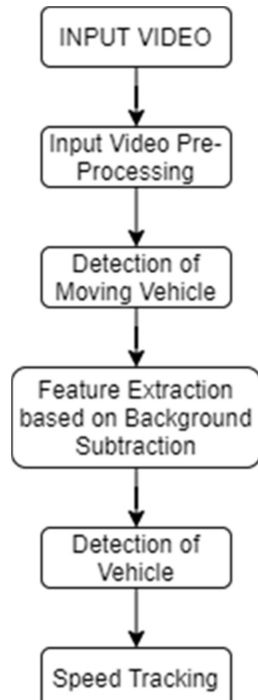
Gaussian mixture model is a probabilistic function used for representation of normally distributed data points in a complete set of data points. These models do not require a prior knowledge of the data point is from which subpopulation cluster which allows the model to learn in an unsupervised manner. Gaussian mixture models are typically used for extracting features in tracking of numerous objects which uses the number of mixture components and their means to estimate the location of an object at every frame in the series of images or video. [6] The primary aim using this approach is to detect the vehicle and tracing algorithm which can be used to monitor the traffic. This model uses a customary observation pattern change for each pixel in the image matrix. Further, Mahalanobis distance of the Gaussian is calculated depending on customary observation change factor, intensity of colour and determining the Gaussian component mean.

## B. Blob Detection Algorithm:

Blob detection algorithm is a technique which can track the motion of non-stationary objects in the frame. A blob is defined as a collection of pixels which are identified as an object. This algorithm determines the location of the blob in consecutive frames of images. Pixels with similar intensity values or colour codes are clubbed to determine the blob. The algorithm is capable to detect multiple blobs in the same image and differentiate their speed and motion. This method has to estimate factors like size, location and colour to determine if the new blob shows resemblance to the previous blob such that the blob has the same label name.

### A. Speed Tracking:

The presented methodology is used for determining the speed of a moving vehicle towards the camera situated at a considerable distance by tracking the motion of vehicle through series of images. The proposed methodology consists of steps as shown in the figure below

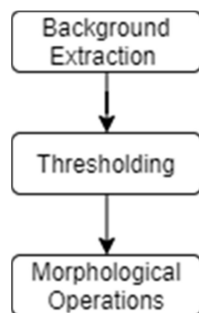


Background Subtraction algorithm is used which subtracts the background from the primary feature/image. An average of all frames is obtained consisting of only the main feature/image hence subtracting the background. Later, the output obtained is applied for Thresholding and Morphological Operation. Detection of the object and centroid is done with the help of Connected Component Method. Centroid is obtained for all frames. Velocity of the vehicle is calculated using the distance travelled by vehicle and frame rate of the input video. The various parameters such as number of frames, frame rate, colour format,

frame size are extracted.

### **✚ Detection of Moving Vehicle:**

The main challenge face during vehicle detection and speed tracking is detecting the main object (In our case the vehicle). To detect moving object there are various approaches such as temporal differencing method, optical flow algorithm, background subtraction algorithm. [7] In temporal difference method, background image is extracted from two adjacent frames with the only drawback of video being slow. The second, Optical flow algorithm detects the object independently based on motion of camera but gets complex for real time applications. In background subtraction absolute difference between background model and each instantaneous frame is taken to detect moving object. Background model is an image which has no moving object in it. In this we would be using Background Subtraction Algorithm which specifically consists of 3 stages.



#### **Background Extraction:**

The video which is recorded on highway consists of object along with the background, it is very difficult to capture image without the object, thus, for getting such image, background extraction method is used. In this average of all frames is taken and the object is subtracted out leaving the background alone. This extracted image is known as ROI (Region of Interest). Now, each of the frames are converted from RGB to a gray-scale image and each individual frame is multiplied with the ROI(Region of Interest) obtained. Because of this other unwanted noise of waving trees and vehicles is avoided which helps in increasing accuracy. The absolute difference of each instantaneous frame and background model after multiplying both with extracted ROI has taken to detect only moving vehicles.

#### **Thresholding**

Image segmentation is done using thresholding in which gray-scale images are converted into binary images. Threshold value is selected which is very important. To separate foreground vehicle from static background thresholding is used here.

$$g(x, y) = 0 \text{ for } f(x, y) < T$$
$$g(x, y) = 1 \text{ for } f(x, y) \geq T$$

Where  $g(x, y)$  is threshold image,  $T$  is the selected threshold value;  $f(x, y)$  is instantaneous frame. In this work, we got vehicle as object and some noise.

#### **a) Morphological Operations:**

Morphological Operations are used to remove noise from the imperfect segmentations and are well suited for binary images. This is performed on output image obtained from

### ✚ Feature Extraction based on Background Subtraction:

The features in feature extraction are nothing else but independent characteristics of vehicle such as speed, color, shape, centroid, edges etc. The result of connected component analysis is used and a bounding box has been drawn around vehicle. In this work the centroid and histogram of vehicles surrounded by bounding box are selected as features.

### ✚ Detection of Vehicle:

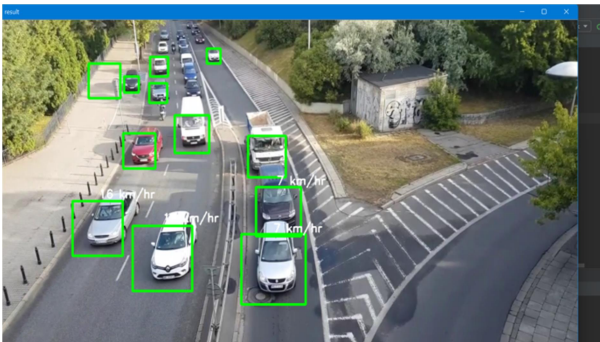
Vehicle detection process is based on the process of feature detection. The features which are extracted are tracked over sequential frames.[Mohit] Matching algorithm is used to determine whether it is the same object or a different one. Mahalanobis distance is used in object matching algorithm. During the past decade, Mahalanobis distance learning has attracted a lot of interest. The Mahalanobis distance between two d-dimensional numerical vectors  $x$  and  $x'$  is defined by

$$d2(x, x') = (x - x')^T M (x - x'), [7]$$

where  $M$  is a  $d \times d$  dimension matrix. Similarity and dissimilarity between two groups is found using Mahalanobis distance. When covariance matrix is same as identity matrix then Mahalanobis distance is same as Euclidian distance. In object matching, Mahalanobis distance between features of object in the previous frame and instantaneous frame is determined. A threshold value is set which is compared with calculated distance. If the distance is less than the threshold value, then the object in the previous frame and instantaneous frame is same. According to this, match id has given for each object and is tracked over for sequential frames.

$$\text{Speed} = \text{Distance} / (\text{TF} * \text{Frame rate})$$

Thus, from distance and travelled time of detected vehicle, speed of that vehicle is determined from above formulae.



## IMPLEMENTATION

```
import cv2
import dlib
import time
import math

carCascade = cv2.CascadeClassifier('myhaar.xml')
video = cv2.VideoCapture('cars.mp4')

WIDTH = 1240
HEIGHT = 740

def estimateSpeed(location1, location2):
    d_pixels = math.sqrt(math.pow(location2[0] - location1[0], 2) + math.pow(location2[1] - location1[1], 2))
    ppm = 8.8
```

```

d_meters = d_pixels / ppm
fps = 18
speed = d_meters * fps * 3.6
return speed

def trackMultipleObjects():
    rectangleColor = (0, 255, 0)
    frameCounter = 0
    currentCarID = 0
    carTracker = {}
    carLocation1 = {}
    carLocation2 = {}
    speed = [None] * 1000
    while True:
        start_time = time.time()
        rc, image = video.read()
        if type(image) == type(None):
            break

        image = cv2.resize(image, (WIDTH, HEIGHT))
        resultImage = image.copy()

        frameCounter = frameCounter + 1
        carIDToDelete = []
        for carID in carTracker.keys():
            trackingQuality = carTracker[carID].update(image)

            if trackingQuality < 7:
                carIDToDelete.append(carID)

        for carID in carIDToDelete:
            carTracker.pop(carID, None)
            carLocation1.pop(carID, None)
            carLocation2.pop(carID, None)
        if not (frameCounter % 10):
            gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
            cars = carCascade.detectMultiScale(gray, 1.1, 13, 18, (24, 24))

```

```

for (_x, _y, _w, _h) in cars:
    x = int(_x)
    y = int(_y)
    w = int(_w)
    h = int(_h)
    x_bar = x + 0.5 * w
    y_bar = y + 0.5 * h
    matchCarID = None

    for carID in carTracker.keys():
        trackedPosition = carTracker[carID].get_position()

        t_x = int(trackedPosition.left())
        t_y = int(trackedPosition.top())
        t_w = int(trackedPosition.width())
        t_h = int(trackedPosition.height())
        t_x_bar = t_x + 0.5 * t_w
        t_y_bar = t_y + 0.5 * t_h

```

```
if ((t_x <= x_bar <= (t_x + t_w)) and (t_y <= y_bar <= (t_y + t_h)) and (x <= t_x_bar <= (x + w)) and (y <= t_y_bar <= (y + h))):  
    matchCarID = carID  
  
if matchCarID is None:  
    print ('Creating new tracker ' + str(currentCarID))  
  
    tracker = dlib.correlation_tracker()  
    tracker.start_track(image, dlib.rectangle(x, y, x + w, y + h))  
  
    carTracker[currentCarID] = tracker  
    carLocation1[currentCarID] = [x, y, w, h]  
  
    currentCarID = currentCarID + 1
```

```
for carID in carTracker.keys():  
    trackedPosition = carTracker[carID].get_position()  
    t_x = int(trackedPosition.left())  
    t_y = int(trackedPosition.top())
```



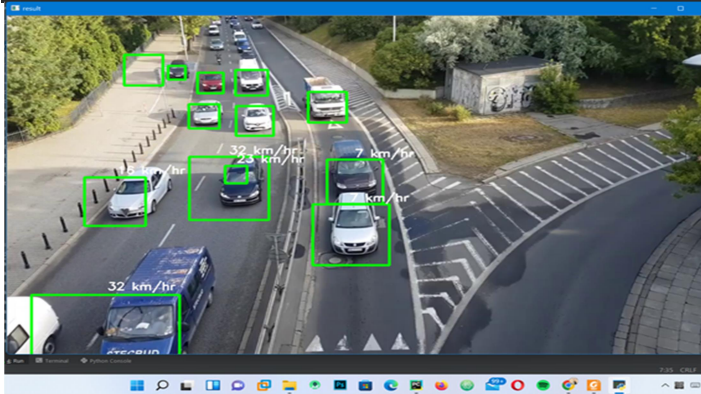
```

# print 'previous location: ' + str(carLocation1[i]) + ', current location: ' + str(carLocation2[i])
carLocation1[i] = [x2, y2, w2, h2]

# print 'new previous location: ' + str(carLocation1[i])
if [x1, y1, w1, h1] != [x2, y2, w2, h2]:
    if (speed[i] == None or speed[i] == 0) and y1 >= 275 and y1 <= 285:
        speed[i] = estimateSpeed([x1, y1, w1, h1], [x2, y2, w2, h2])

    if speed[i] != None and y1 >= 180:
        cv2.putText(resultImage, str(int(speed[i])) + " km/hr", (int(x1 + w1/2), int(y1-
5)), cv2.FONT_HERSHEY_SIMPLEX, 0.75, (255, 255, 255), 2)
    cv2.imshow('result', resultImage)
    if cv2.waitKey(33) == 27:
        break
cv2.destroyAllWindows()
if __name__ == '__main__':
    trackMultipleObjects()

```



## CONCLUSION

Road safety and reducing accidents is a very crucial issue and must be considered at utmost priority. One must abide the rules of maintaining appropriate speed guidelines. Technological tools and tracking devices which help in monitoring the motion and speed of vehicles can help reduce the number of accidents on roads as well as trace the origins of the mishap. In this paper, we have discussed the challenges and obstacles faced while implementing a system which detects a vehicle and monitors its speed and motion. The separation of foreground and background objects and commonly preferred approaches to solve this issue. In addition, to this we have also suggested a possible formulation which can be used to detect the motion of vehicle. Furthermore, the paper also talks about the speed tracking algorithm and tried to elucidate the working of these algorithm and mathematics involved behind it. To support our thesis, we have also mentioned snippets from the system we designed for vehicle detection. Several nations are already using such systems to detect the speed and direction of vehicle. Moreover, some systems have advanced to the capacity of detecting the number plates of vehicles which are blurred for normal cameras and uses image processing algorithms to sharpen the image and extract the number plate which makes it even easier to locate the vehicle. Also, the speed breakers can be designed in such a way that they only rise when the vehicles speed is

above the permissible limit.

We have used opencv and haar cascade classifiers for object detection. Haar cascade is a approach based on machine learning where a cascade function is trained from a series of images which includes positive and negative. After the training it is used to detect objects in other images/videos.

We have thus analyzed various methods for speed tracking and vehicle detection and implemented a optimum solution.

