Detecting Vehicle Speed Based on Video Using Computer Vision

M P Darshana Sampath Dahanayake
FGS/MDA/2021/030
University of Kelaniya, Sri lanka
dsdahanayake@yahoo.com

***Abstract*— Road safety is an important consideration in the transportation system. Approximately 1.3 million people die each year as a result of road traffic crashes. This is why speed limits are given for a particular road depending on the quality of the road and the how prone the road is to accidents. Speed Cameras are set up at intervals of the road in order to catch speed-limit violators. Just like any other technology, speed cameras have progressed over the years. This project will focus on making a speed camera without sensors and only with Image processing of videos**

***Keywords*— Speed detection, Computer Vision, Object detection, Image Subtraction, Open CV**

I. INTRODUCTION

Every year the lives of approximately 1.3 million people are cut short as a result of a road traffic crash. Between 20 and 50 million more people suffer non-fatal injuries, with many incurring a disability as a result of their injury. Road traffic injuries cause considerable economic losses to individuals, their families, and to nations as a whole. These losses arise from the cost of treatment as well as lost productivity for those killed or disabled by their injuries, and for family members who need to take time off work or school to care for the injured. Road traffic crashes cost most countries 3% of their gross domestic product.

In order to address issue and limit the speed of vehicle authorities deployed mobile units to measure the speed and take legal actions accordingly. Authorities uses Doppler speed radar equipment with traffic police or road safety officers. Other method is modern stationery speed camera stations deployed in Highways. Theses stationary speed cameras can send larger 24GHz Doppler tracker radar beam to multiple vehicle targets. It contains more sensors than handheld Doppler radar and capable of detecting speed of cluster of vehicles at the same time simultaneously. However. These equipment carries lot of cost and expenses to maintain them.

Present project prototypes the method to detect moving objects speed by using video surveillance camera data. The goal is to determine vehicle speed, count and save the image of the vehicles, which travels beyond the speed limit. This project aims to use computer vision, rather convolutional neural network to solve set task and then detecting vehicle positions and speed using background subtraction and mixture of Gaussian (MoG) method. This way we could limit the use of expensive stationery speed cameras and doppler radars. Drivers will be monitors seamlessly without human intervention only using computer vision and artificial intelligent.

II. PROBLEM STATEMENT

The objective of this project is to create a computer vision application using Image Processing in Python by using OpenCV and Tensorflow.

When it is comes to tracking the speed of vehicles on a segment of the road, vital steps of this project is:

- Vehicle Detection
- Speed estimation
- Capturing vehicle image

❖ Vehicle Detection

As the background of the vehicles is stationary (as the speed camera is stationary) image subtraction is used to detect moving vehicle.

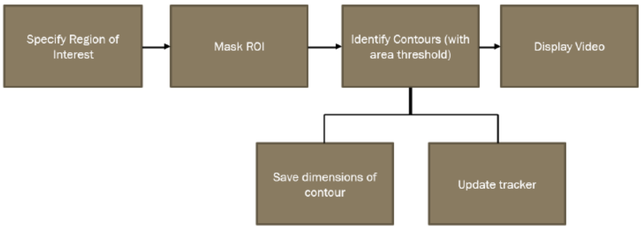

Figure 1: Vehicle Detection Block Diagram

❖ Speed Estimation

The speed of a vehicle can be estimated when a tracked vehicle covers a segment of the road.

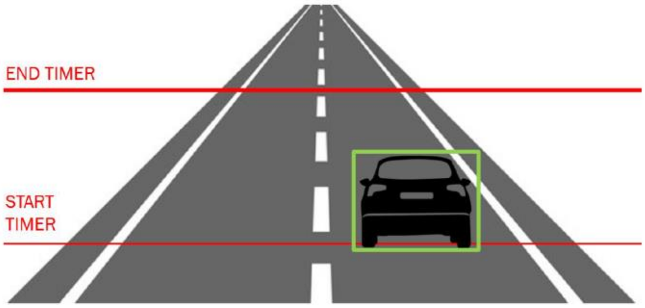

Figure 2: Speed Timer Diagram

❖ Capturing Vehicle Image

Based on Contour detection, the image of a particular ID is saved to a folder along with the speed. Picture is saved as soon as the speed is estimated.

III. PROJECT MODEL

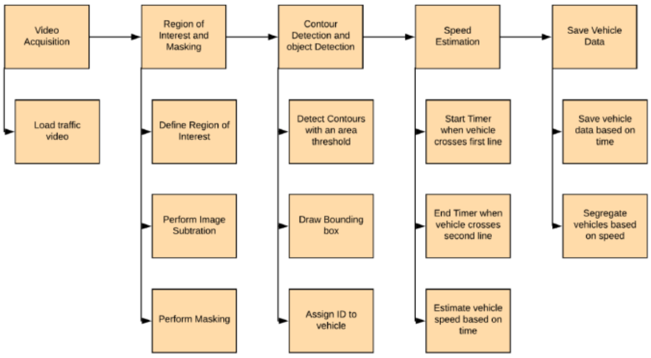

Figure 3: Project Model Block Diagram

IV. METHODS

Video Acquisition:

Figure 4: Sample Video Screenshot

The video was recorded in Paliyagoda-Katunayake highway Sri Lanka using a smart mobile phone. Video was taken with 720p video quality at 30 fps frame rate. Due to limited resourcing power video was taken at 352 x 640 resolution. However, the number plates of the vehicles are not clearly visible in the video.

Region of Interest and Masking:

Region of Interest (ROI) takes a smaller portion of the original video. On this ROI, Image subtraction is performed to detect a moving vehicle. (Image Subtraction helps find the difference between two frames). Masking is performed to make the moving vehicles appear white and the rest of the image black. Using this method image is converted in to binary value of every pixel and represent the it using white and black (1 and 0). Masking will sperate the foreground of the video (as white color) and background (black color).

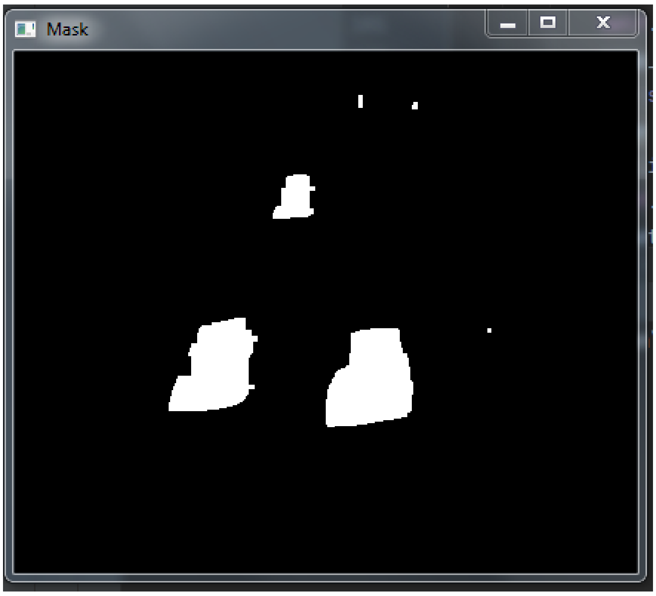

Figure 5: Masked Image

Contour Detection and Object Tracking:

Based on the area threshold of number of pixels, the contours are detected. The threshold is used to avoid detecting contours of smaller moving objects that are not vehicles. By experimenting I have found the threshold value as 725 and any object more than that value considered as a vehicle. The object is tracked based on the distance between two contours between frames. Mainly objects were detected calculating Euclidean distance of each frame and established a threshold for distance difference. Any Euclidean distance difference below the threshold, in this project below 100 considered as same object. Based on the video frame size or resolution, the threshold value could be varying. An ID is assigned to each contour once it identifies as a different object.

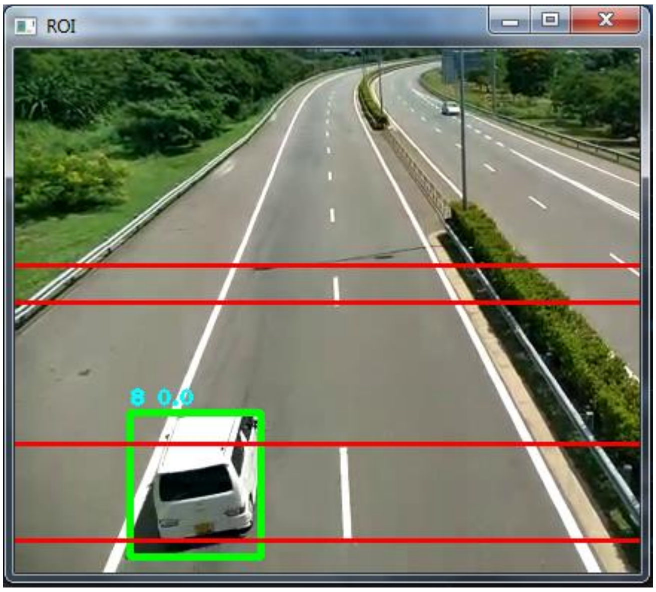

Figure 6: Contour Detection

Speed Estimation:

Time difference between the position of a vehicle is calculated and the speed is estimated based on a formula. The timer starts when the vehicle crosses the first line, and the timer ends when the vehicle crosses the second line. The speed is displayed on top of the bounding box only when the vehicle crosses both the lines.

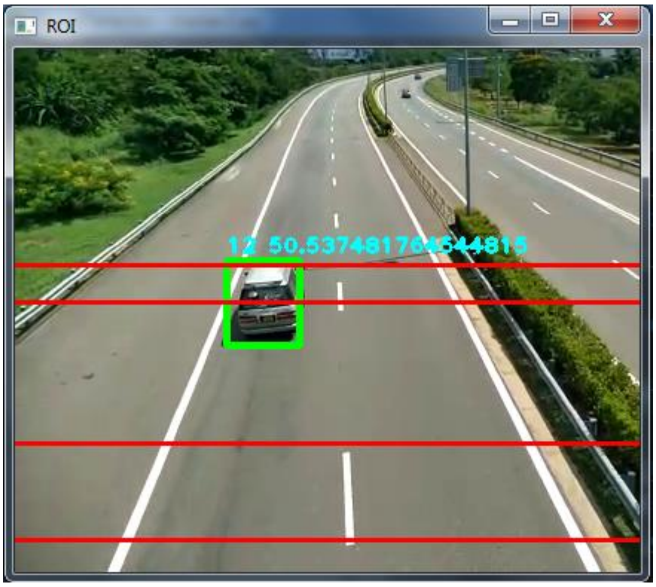

Figure 7: Speed Estimation

Save Vehicle Data:

This can be done after saving still images of the violators. Each vehicle is given a specific ID. The ID and speed details are saved to a text file

A good quality video is required to capture the number-plate of vehicles

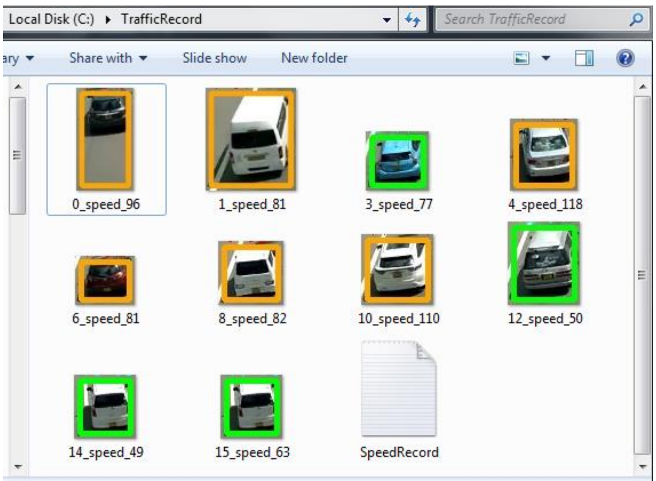

Figure 8: Saved Vehicle Pictures

Create Summary:

The vehicle data is saved in a text file. The vehicles that exceeded the speed limit are pointed. A summary of number of vehicles and the speed violators are displayed.

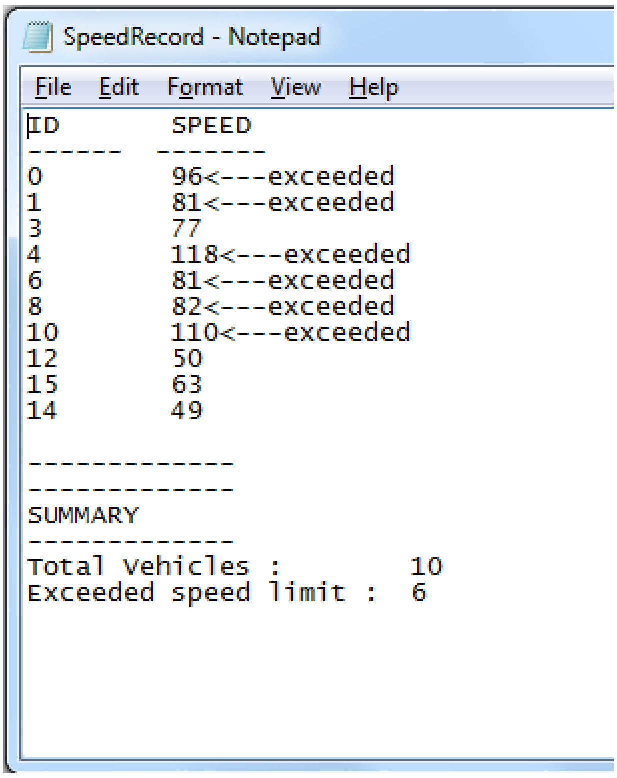

Figure 9: Saved Summary

V. Implementation

1) *Video Acquisition*

```
cap =
cv2.VideoCapture("Resources/traffic5.mp4")
```

2) *Region of Interest and Masking*

```
#KERNALS
kernalOp = np.ones((3,3),np.uint8)
kernalOp2 = np.ones((5,5),np.uint8)
kernalCl = np.ones((11,11),np.uint8)
fgbg=cv2.createBackgroundSubtractorMOG2(detec
tShadows=True)
kernal_e = np.ones((5,5),np.uint8)
```

```
#MASKING
fgmask = fgbg.apply(roi)
ret, imBin = cv2.threshold(fgmask, 200, 255,
cv2.THRESH_BINARY)
mask1 = cv2.morphologyEx(imBin,
cv2.MORPH_OPEN, kernalOp)
mask2 = cv2.morphologyEx(mask1,
cv2.MORPH_CLOSE, kernalCl)
e_img = cv2.erode(mask2, kernal_e)
```

3) *Contour Detection*

```
contours,_ =
cv2.findContours(e_img,cv2.RETR_TREE,cv2.CHAI
```

```
N_APPROX_SIMPLE)
detections = []

for cnt in contours:
    area = cv2.contourArea(cnt)
    #THRESHOLD
    if area > 725:
        x,y,w,h = cv2.boundingRect(cnt)
         cv2.rectangle(roi,(x,y),(x+w,y+h
         ),(0,255,0),3)
        detections.append([x,y,w,h])
```

4) Object Tracking

```
for rect in objects_rect:
    x, y, w, h = rect
    cx = (x + x + w) // 2
    cy = (y + y + h) // 2

    #CHECK IF OBJECT IS DETECTED ALREADY
    same_object_detected = False

    for id, pt in self.center_points.items():
        dist = math.hypot(cx - pt[0], cy -
         pt[1])

        if dist < 100:
            self.center_points[id] = (cx, cy)
            objects_bbs_ids.append([x, y, w,
             h, id])
            same_object_detected = True
```

5) Speed Estimation

a) Timer Start and stop

```
#START TIMER
if (y >= 265 and y <= 330):
    self.s1[0,id] = time.time()
    #print("Start time: ",int(self.s1[0,id]))

#STOP TIMER and FIND DIFFERENCE
if (y >= 155 and y <= 170):
    self.s2[0,id] = time.time()
    self.s[0,id] = self.s2[0,id] -
    self.s1[0,id]
```

b) Speed Formula

```
#SPEEED FUNCTION
def getsp(self,id):
    if (self.s[0,id]!=0):
        s = 115.7310 / self.s[0, id]
        #s = self.s[0, id]
    else:
        s = 0

    return int(s)
```

6) Drawing Rectangles and displaying on the screen

```
for box_id in boxes_ids:
    x,y,w,h,id = box_id


    if(tracker.getsp(id)<tracker.limit()):
        cv2.putText(roi,str(id)+"
"+str(tracker.getsp(id)),(x,y-5),
cv2.FONT_HERSHEY_PLAIN,1,(255,255,0),2)
        cv2.rectangle(roi, (x, y), (x + w, y + h),
(0, 255, 0), 3)
    else:
        cv2.putText(roi,str(id)+ "
"+str(tracker.getsp(id)),(x, y-
5),cv2.FONT_HERSHEY_PLAIN, 1,(0, 0, 255),2)
        cv2.rectangle(roi, (x, y), (x + w, y + h),
(0, 165, 255), 3)
```

7) Drawing Reference Lines

```
# DRAW LINES
cv2.line(roi, (0, 265), (460, 265), (0, 0,
255), 2)
cv2.line(roi, (0, 330), (460, 330), (0, 0,
255), 2)

cv2.line(roi, (0, 145), (460, 145), (0, 0,
255), 2)
cv2.line(roi, (0, 170), (460, 170), (0, 0,
255), 2)
```

8) Save Vehicle Images and speeds

```
#SAVE VEHICLE DATA
def capture(self,img,x,y,h,w,sp,id):
    if(self.capf[id]==0):
        self.capf[id] = 1
        self.f[id]=0
        crop_img = img[y-5:y + h+5, x-5:x + w+5]
        n = str(id)+"_speed_"+str(sp)
        file = 'C://TrafficRecord//' + n + '.jpg'
        cv2.imwrite(file, crop_img)
        self.count += 1
        filet = open("C://TrafficRecord//SpeedRecord.txt", "a")
        if(sp>limit):
            file2 = 'C://TrafficRecord//exceeded//' + n + '.jpg'
            cv2.imwrite(file2, crop_img)
            filet.write(str(id)+" \t "+str(sp)+"<---exceeded\n")
            self.exceeded+=1
        else:
            filet.write(str(id) + " \t " + str(sp) + "\n")
        filet.close()
```

9) Create Summary

```
#TEXT FILE SUMMARY
def end(self):
    file = open("C://TrafficRecord//SpeedRecord.txt", "a")
    file.write("\n-------------\n")
    file.write("-------------\n")
    file.write("SUMMARY\n")
    file.write("-------------\n")
    file.write("Total Vehicles :\t"+str(self.count)+"\n")
    file.write("Exceeded speed limit :\t"+str(self.exceeded))
    file.close()
```

VI. OUTPUT SCREENSHOTS

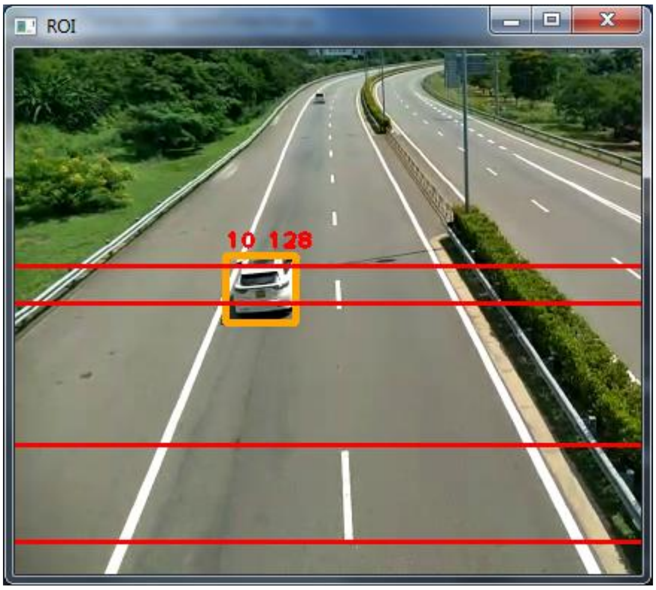

Figure 10: Speed Detector Main Output

VII. PREVIOUS STUDIES

- Title: Vehicle speed measurement model for video-based systems

  This paper uses segmentation of the road into strips for estimation of the speed of the vehicle. If we know the length of a segment of road and the time taken to cover it, we can determine the speed. Frame-rate can be used to assess the accuracy of speed calculation.

  A minimum speed of object is set up so as to not record unnecessary objects like bicycles and bird shadows.

- Determining vehicle speed based on video image subtraction

  This paper uses deep learning to identify vehicles based on image subtraction and contour detection.

VIII. RESULTS ANALYSIS

This project is successfully able to track vehicles and estimate their speed. Accuracy in detecting vehicles is 100% as long as there is no movement in the camera. Estimation of speed can have a difference of 0-2 km/hr depending of the program execution speed.

Multiple vehicles can be detected, and their speeds can be detected. However, if two vehicles are moving extremely close to each other, it may be detected as a single object.

This project requires the camera to be as still as possible, as movement is used to distinguish vehicles from the background.

IX. CONCLUSION

Road safety is an important factor for the police force. And as citizens it is our responsibility to follow rules and maintain safety on our roads. This project is able to estimate speed of vehicles and save vehicle data.

Future Improvements can be done by integrate a deep learning technique to identify the type of the vehicles using classification. Further this can be implemented with Raspberry pi 4 to make handheld device with live CCTV feed to detect the speed real time.

X. ACKNOWLEDGEMENT

I would like to thank Mr. Dinesh Asanka, Department of Industrial Management UOK, for his guidance and support throughout the project and for giving me the opportunity to explore deeper into Computer Vision concepts both theoretically and practically.

REFERENCES

[1] AlexanderGrents,VitaliiVarkentin and NikolayGoryaev, "Determining vehicle speed based on video using convolutional neural network" Science direct , Nov 2020.

[2] Alexander A SGunawan, Deasy Aprilia Tanjung, Fergyanto E.Gunawanb, " Detection of Vehicle Position and Speed using Camera Calibration and Image Projection Methods" Science direct , Oct 2021.

[3] Saleh Javadi, Mattias Dahl, Mats I Pettersson, "Vehicle speed measurement model for video-based systems" Science direct , Apr 2019.

[4] Tarun Kumar, Dharmender Singh Kushwaha, "An Efficient Approach for Detection and Speed Estimation of Moving Vehicles" Science direct , Aug 2016.

[5] YouTube – OpenCV – Murtaza's Workshop, Mar 2022.