GALGOTIAS UNIVERSITY
(Established under Galgotias University Uttar Pradesh Act No. 14 of 2011)

VEHICLE DETECTION FOR

INTELLIGENT VEHICLE SURVILLENCE

A Report For The Evaluation 3 Of Project 2

submitted by

PRIYANKA SHIVHARE

(1613101525/16SCSE101187)

in partial fulfillment for the award of the degree

of

Bachelor Of Technology

IN

COMPUTER SCIENCE

SCHOOL OF COMPUTING SCIENCE AND ENGINEERING

Under the supervision of

S.JERALD NIRMAL KUMAR,

ASSISTANT PROFESSOR

May 2020

Table of contents

Abstract:

Indian roads are getting congested day by day. The main reason is increase in the no. of vehicles per year which leads to increase in road network. According to the data, in January 2019, more than a million and half vehicles were bought and registered which estimate that at an average 51000 new vehicles bought across India per day. Increase in road congestion results in crime, accident, obstruct emergency healthcare services, etc. Road length can be a solution but it is time taking and costly. Vehicle detection and counting is one of the best solution for traffic congestion. Vehicle detection and counting will help in reducing crimes, accident, reducing delay of emergency healthcare services, etc.

Vehicle detection and counting is very helpful in intelligent transport system. Vehicle detection and counting will be done by image processing and video processing in opencv python. Due to background movement (i.e. moving tress) and change in environment (i.e. rain, wind, etc) ,it will be very challenging to detect moving vehicles. We will be saving the data of vehicle detection and counting in google cloud (i.e. fcm firebase cloud messaging). We can easily retrieve the data from fcm.

Keywords:

Road congestion, vehicle detection, vehicle counting, google firebase, image processing using opencv.

INTRODUCTION:

Today traffic survillence has become the major topic for research because has major impact on everyone life. Day by day ,the number of vehicles are increasing. This is becoming the main cause of traffic jam and other problems caused in the traffic. Widening the roads is a simple solution , but it is very expensive. Today we have achieved lots of progress in image processing and video processing. We can implement vehicle detection and counting by image processing and video processing for intelligent transport system and that also at extremely low cost. We just need video data of the traffic which we can easily get from video camera. Video camera are easy to install and at the same time low cost.

We can detect and count vehicles in many ways, but here we will be using opencv in python to detect and count vehicles. This system involves detecting moving vehicles and counting them. We can use this system for vehicle speed detection,the frequency of vehicles ,detecting and alerting the chances for future traffic congestion, detecting ambulance. This system will consider a background image , and on the basis of background image it will detect the moving vehicle, and on the basis of moving vehicle data ,it will collect vehicle counting data.

In this process, the challenging part is background obstacles. These obstacles can be dusty/windy weather, moving trees, rain, moving camera, strong winds, sun rise ,sunset, shadow movement etc. Due to these obstacles, background is also detected with the moving object and the collected data becomes less accurate. To prevent such problem, we apply algorithm where we are setting a frequency value. If the moving object movement frequency value is more than constant frequency , then it will be considered as moving object or else it will be considered as background movement. The constant frequency will be changed after some interval of time depending on background conditions, i.e. in heavy rainfall and strong winds, the background movement will increase, and so the frequency value and vice versa.

We are using google cloud, so we don’t have to worry about the storage. We can store the data and retrieve directly from google firebase.

Literature survey:

In past few years, the detection of Objects in real time and Image processing has become an active area of research and several new approaches have been proposed. Several researchers have conducted many studies about Object detection1. S.V. Viraktamath, Mukund Katti, Aditya Khatawkar& Pavan Kulkarni has conducted a study of openCV and also have published an IEEE paper for Face Detection and Tracking using OpenCV. Their work is related with converting web cam captured 2D Images and convert them into 3D Images related to human faces by constructing 3D Geometry data outputs . Ashish Pant, Arjun Arora, Sunnet Kumar and Prof. R.P. Arora form DIT Dehradun have researched about Image Processing and encrypting an Image in order to transfer safely over the networks. They entitled their work as Sophisticated Image Encryption Using OpenCV Kevinhughes, an elite individual in Opencv area has written a number of blogs containing projects tutorials in this area and steps for installing various software. Serge Belongie and Jitendra Malik, members of IEEE have done a was study in the field of Shape Matching and Object Matching Based on their shapes, differentiating two object based on the difference in their shapes. Orlando J. Tobias, and Rui Seara, Member, IEEE, have put their great efforts studding the ways and techniques for Image Segmentation and histogram Thresholding.

Problem statement:

The main problem in the object detection is the background motion. Background movement is sometimes more ,sometime less ,due to this we are not able to set the background image. And same is in the object counting. Shadow in the background image is becoming an obstacle in object counting. Heavy rainfall and strong winds are also a problem, because due to heavy rainfall and strong wind the camera is moving. The input video is blurred. And our system works only on those videos in which camera is not moving. Also dust particles in the video are decreasing the clearity of the video.

Proposed model:

We will be using different method for vehicle detection and vehicle counting.

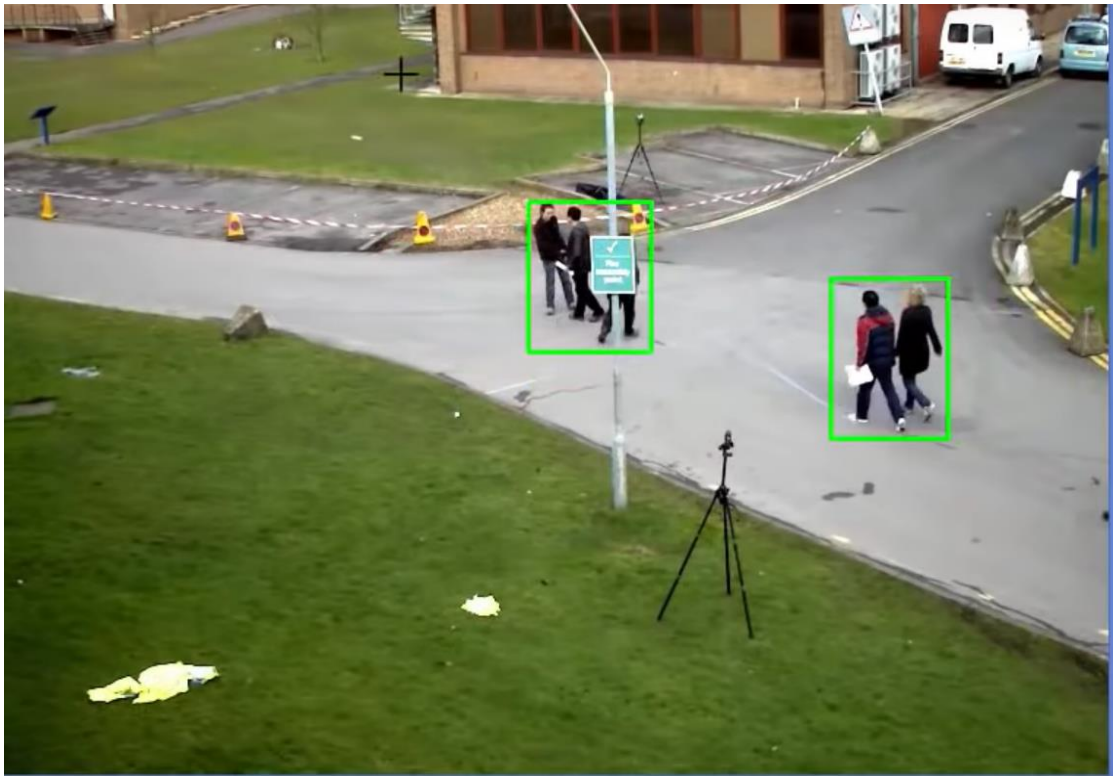
1. Vehicle detection: We will be contour in opencv for vehicle detection. It will join all continuous point along the boundary which are having same colour or intensity. We will use binary image for finding contour. We will be apply threshold or canny edge detection to find contour of the image.
2. Vehicle counting: We will be using tenserflow and opencv for counting the vehicle counting.

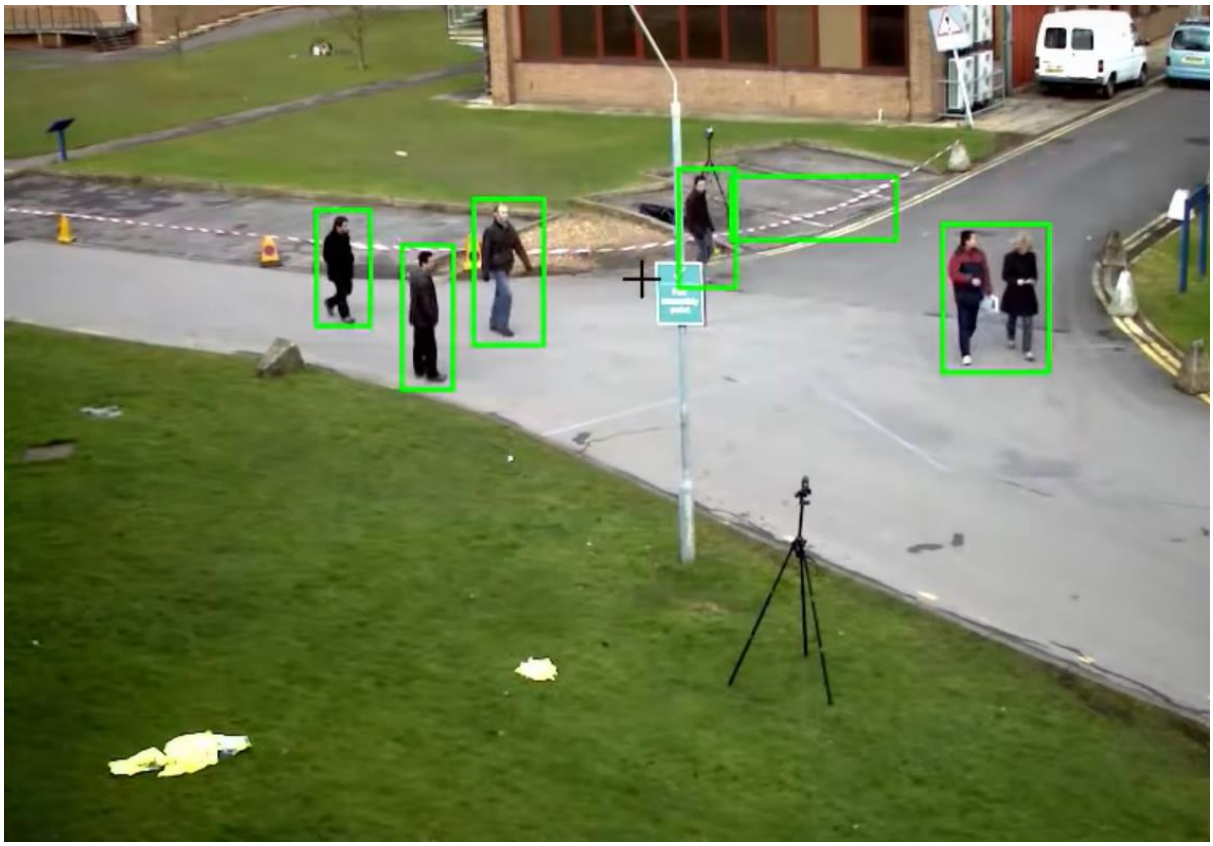
Implementation:

Vehicle detection:

```
Import numpy as np
Import cv2
import pyrebase


config = {
  "apiKey": "AIzaSyCrDjbyixHFN7BcC_hY_bWToNz7Jg_vZMQ",
  "authDomain": "my-first-project-512eb.firebaseapp.com",
  "databaseURL": "https://my-first-project-512eb.firebaseio.com",
  "projectId": "my-first-project-512eb",
  "storageBucket": "my-first-project-512eb.appspot.com",
  "messagingSenderId": "1046966982590",
  "appId": "1:1046966982590:web:3c03ddcbcdf4a188b98bef",
  "measurementId": "G-JEWBBSK2TY"
}

firebase=pyrebase.initialize_app(config)
storage=firebase.storage()
storage.child("video/video.mp4").download("video.mp4")
cap = cv2.VideoCapture("videooo.mp4")
frame_width=int( cap.get(cv2.CAP_PROP_FRAME_WIDTH))


frame_height=int( cap.get( cv2.CAP_PROP_FRAME_HEIGHT))


fourcc= cv2.VideoWriter_fourcc('X','V','I','D')


out = cv2.VideoWriter("output.avi", fourcc, 5.0, (1280,720))


ret, frame1 =cap.read()
ret, frame2 =cap.read()
print(frame1.shape)
whilecap.isOpened():
   diff = cv2.absdiff(frame1, frame2)
gray= cv2.cvtColor(diff, cv2.COLOR_BGR2GRAY)
   blur = cv2.GaussianBlur(gray, (5,5), 0)
   _, thresh = cv2.threshold(blur, 20, 255, cv2.THRESH_BINARY)
   dilated = cv2.dilate(thresh, None, iterations=3)
```

```
    contours, _ = cv2.findContours(dilated, cv2.RETR_TREE,
cv2.CHAIN_APPROX_SIMPLE)


for contour in contours:
        (x, y, w, h) = cv2.boundingRect(contour)


if cv2.contourArea(contour) <900:
continue
        cv2.rectangle(frame1, (x, y), (x+w, y+h), (0, 255, 0), 2)
        cv2.putText(frame1, "Status: {}".format('Movement'), (10, 20),
cv2.FONT_HERSHEY_SIMPLEX,
1, (0, 0, 255), 3)
#cv2.drawContours(frame1, contours, -1, (0, 255, 0), 2)


    image = cv2.resize(frame1, (1280,720))
out.write(image)
    cv2.imshow("feed", frame1)
    frame1 = frame2
    ret, frame2 =cap.read()


if cv2.waitKey(40) ==27:
break


cv2.destroyAllWindows()
cap.release()
out.release()
```

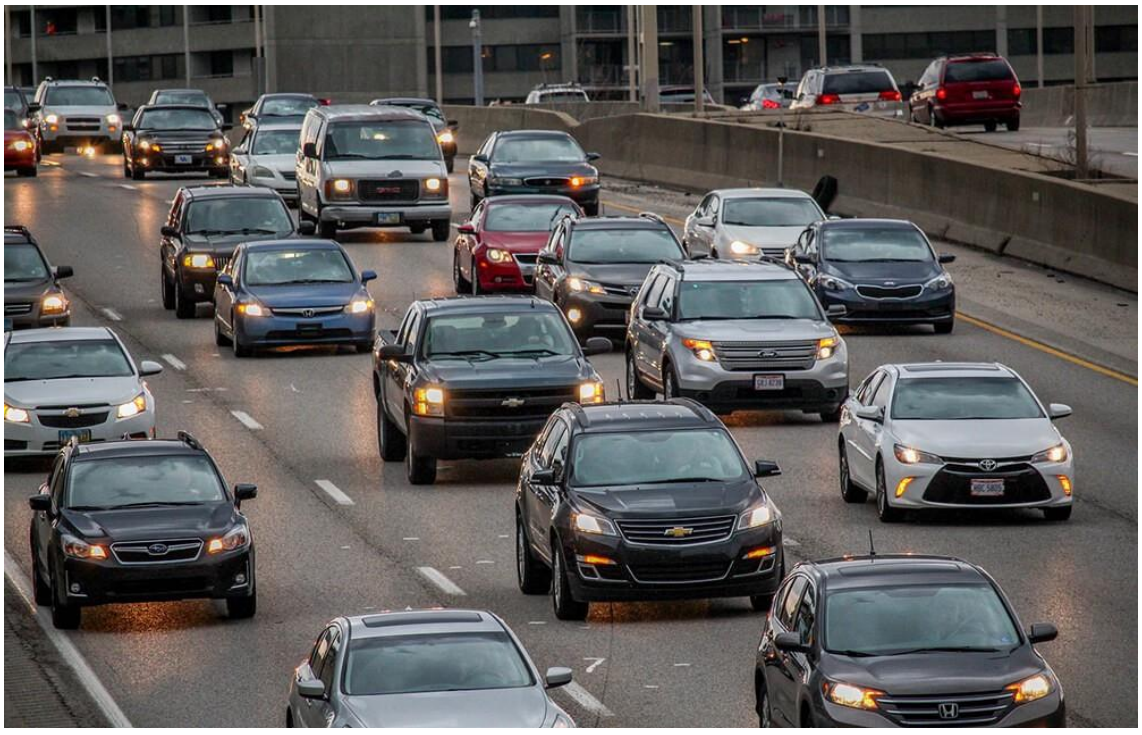


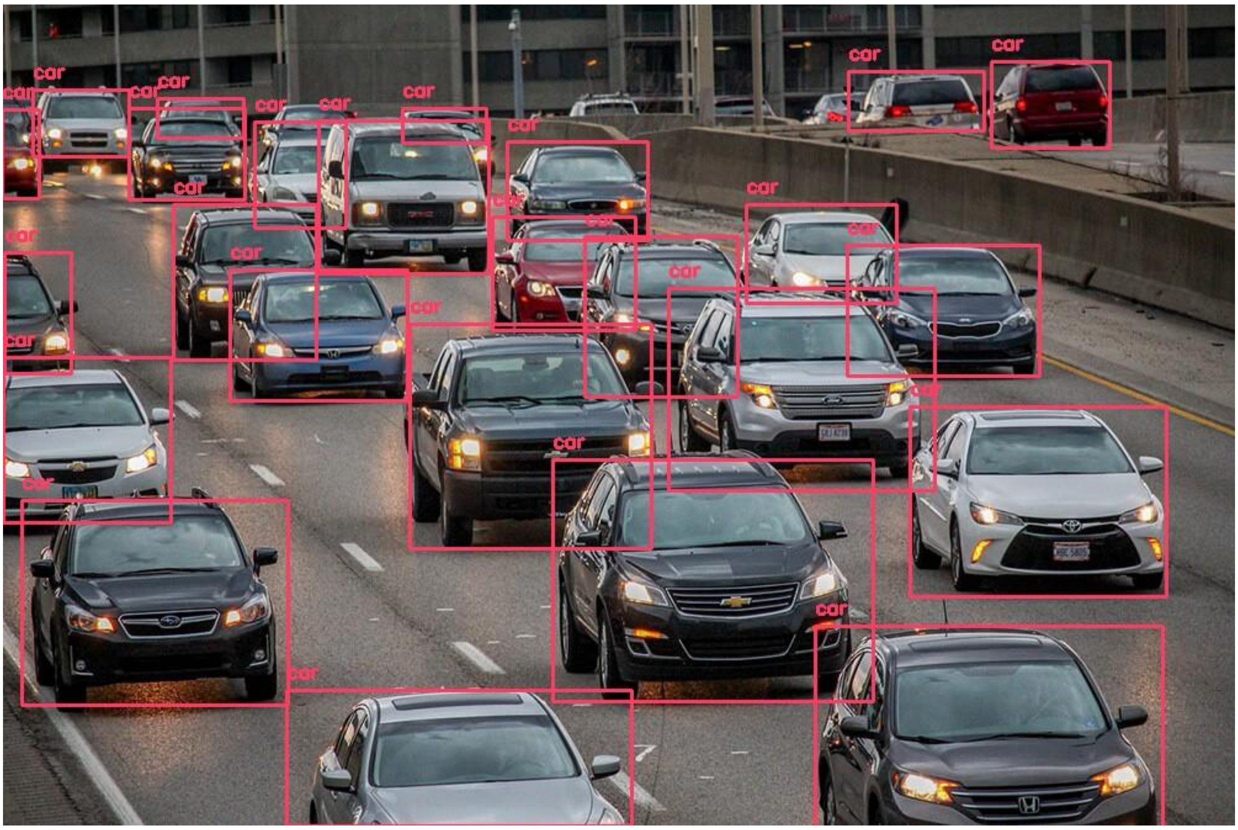


2. **Vehicle counting:**

```
import cv2
import matplotlib.pyplot as plt
import cvlib as cv
from cvlib.object_detection import draw_bbox
import pyrebase


config = {
  "apiKey": "AIzaSyCrDjbyixHFN7BcC_hY_bWToNz7Jg_vZMQ",
  "authDomain": "my-first-project-512eb.firebaseapp.com",
  "databaseURL": "https://my-first-project-512eb.firebaseio.com",
  "projectId": "my-first-project-512eb",
  "storageBucket": "my-first-project-512eb.appspot.com",
  "messagingSenderId": "1046966982590",
  "appId": "1:1046966982590:web:3c03ddcbcdf4a188b98bef",
  "measurementId": "G-JEWBBSK2TY"
}

firebase=pyrebase.initialize_app(config)
storage=firebase.storage()
storage.child("video/video.mp4").download("video.mp4")
im = cv2.imread('cars_4.jpeg')bbox, label, conf = cv.detect_common_objects(im)output_image =
draw_bbox(im, bbox, label, conf)plt.imshow(output_image)
plt.show()print('Number of vehicles in the image is '+ str(label.count('car')))
```

Result:





Output: Number of vehicle in the image is 3