



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

IJRSET

International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 12, Issue 6, June 2023

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.423

9940 572 462

6381 907 438

ijirset@gmail.com

www.ijirset.com

Traffic Volume Study by Using Python

Prof. Swapnil Mane, Sonal Jagtap, Jyoti Bhokre, Pallavi Maghade, Atul Patil

Assistant Professor (Research Guide), Department of Civil Engineering, Bharati Vidyapeeth's College of Engineering, Lavale, Pune, India

Research Scholar, Department of Civil Engineering, Bharati Vidyapeeth's College of Engineering, Lavale, Pune, India

Research Scholar, Department of Civil Engineering, Bharati Vidyapeeth's College of Engineering, Lavale, Pune, India

Research Scholar, Department of Civil Engineering, Bharati Vidyapeeth's College of Engineering, Lavale, Pune, India

Research Scholar, Department of Civil Engineering, Bharati Vidyapeeth's College of Engineering, Lavale, Pune, India

ABSTRACT: This paper focuses on problem of traffic. Traffic is a biggest problem in city now a days. Traffic congestion is one of the problems for cities around the world due to the rapid increasing of vehicles in urbanization. Traffic flow prediction is of a great importance for intelligent Transport System (ITS) which helps to optimize the traffic regulation of a transportation in the city. Now a days, several researches have been studied so far on traffic flow prediction, accurate prediction has not yet been exploited by most of existing studies due to the impact of inabilities to effectively deal with spatial temporal features of the times series data. Traffic information in transportation system will also be affected different factors. Hence we are discuss in this about traffic problem and try to provide solution with the help of object detection in open cv.

KEYWORDS: Vehicle detection, Traffic Analysis, Open Cv, Python, Car Cascade,

I. INTRODUCTION

Now a day's transportation is one of the most burning issues in every territory of the world. Every country is approaching differently according to their needs and solving their transportations problems within their capabilities. In designing buildings, we need to determine loads coming to the structure to calculate reinforcement to be provided for safe functioning of the structure. Here in transportation volume serves the same purpose. For planning, designing and operation of transportation system the first and foremost requirement is volume. Volume is simply the number of vehicles passing a section of a roadway. Expressing traffic volume as number of vehicles passing a given section of road or traffic lane per unit time will be inappropriate when several types of vehicles with widely varying static and dynamic characteristics are comprised in the traffic. Traffic jams on urban Network are increasing day by day, because the traffic demand increases, and the speed of the vehicles is drastically reduced thus causing longer vehicular queuing and more such cases substantially hamper the traffic flow by giving rise to holdup.

Such situations highlight towards the drawback such as

- Increase in pollution
- Wear and tear of vehicles
- Delays may result in late arrival

Traffic studies are required to transportation engineers for :

- planning and designing traffic facilities
- determining the need for traffic control devices
- studying the effectiveness of introduced schemes

- diagnosing given situations and finding appropriate solutions
- forecasting the effects of projected strategies
- calibrating and validating traffic models etc

Research Gap

With the help of previous work done we have finalized our methodology for our project. we will see how Vehicle Detection can be done using Python OpenCV via an Image file, webcam, or video file. Traffic volume study is play an important role in transportation engineering. Hence we have done traffic volume study with some advance technique. Traffic analysis will calculate for the number of vehicles in an area per some period of time and classify the vehicles. But the moving vehicles and their detection, tracking, and counting are very critical for flow of traffic in monitoring, planning, and controlling. Hence we decided the By analyzing the recorded video of traffic flow sequence from a camera, a video- based solution technology is applied in combination With this technology we applied Open CV for real time video applications. Such methods help us to detect, track, count and classify the moving vehicles.

The main goal of Vehicle detection and counting in traffic video projects is to develop and implement methodology for automatic detection and counting of moving vehicles on highways. Using object detection techniques, we can do extremely helpful high-value jobs such as surveillance, traffic control, criminal fighting, etc. Counting the number of objects, determining the relative size of the items, and determining the relative distance between the objects are all sub-tasks in object detection. These sub-tasks are crucial since they help solve some of the most difficult real-world challenges.

Let's have a look at some of the intriguing object detection use cases in real-world applications. we will develop a coding for a vehicle counting and detection system. It will be enough to work for both the images or the video, for the same, we will be using OpenCV for doing all the image processing operations and for classification the car and bus car cascade classifier for detecting and counting the cars and buses having said that you can also make your car cascade classifier.

Scope of Work

1. Helps traffic police:

A vehicle detection and counting system could be beneficial for the traffic police because everything they can monitor from one place only likes how many vehicles have crossed this toll and which vehicle.

2. Maintaining records:

It is challenging for some individuals to record all the vehicles with them because the cars are passing by in real-time. It's not like that one is watching the video, and they can pause it and have a note of it, so to remove this limitation, this application can be very well-versed to attain the time-saving quality and be automated.

3. Traffic surveillance control:

As this application can be planted anywhere as it only requires a camera or some wires (for establishing the connectivity with the central system) hence if the traffic is high at someplace, then

from that area, an officer can monitor it and forward the information to next toll officer so that they could be prepared beforehand

There is a vast scope for future improvement should be possible for example-

- 1) detection, tracking, counting and classification of moving vehicle should be possible on ongoing live videos.
- 2) Traffic counting, traffic control and classification
- 3) For extension of existing road
- 4) In addition, Speed detection of the vehicle
- 5) Detection of overspeeding vehicle, 3sitting riding on two wheeler
- 6) Management of traffic signals

In that way it may help to reduce the accidents and other safety measures can be taken to it by alarming. This may help to traffic improvement and for safety for passengers who travel on vehicles. From that region only we can calculate the vehicles passing through the tracker line so that high volume of vehicles on that lane can be diverted to other part of roads. Through this video it will be helpful to check the vehicle density and accidents occurred on that lane this system is helpful in all measures of the traffic improvement.

Implementation

The study focuses mainly on the traffic Volume to develop the predictive model. The increase or decrease in the traffic volume with the higher volatility makes it harder to predict however the machine learning models try to predict to some extent with a significant accuracy. Here the implementation is carried out using python.

II. LITERATURE REVIEW

Traffic volume studies are conducted to determine the number, movements, and classifications of roadway vehicles at a given location. These data can help identify critical flow time periods, determine the influence of large vehicles or pedestrians on vehicular traffic flow, or document traffic volume trends. The traffic engineer must acquire general knowledge of traffic volume characteristics in order to measure and understand the magnitude, composition and time and route distribution of volume for each area under his jurisdiction. The present study is essentially about the importance of traffic volume in traffic engineering of urban and suburban road links, in particular the literature on effect of traffic volume, speed-flow relationships, passenger car equivalents, peak hour factor, flow variations and traffic capacity and level of serviceability (LOS). But there are lot of studies which help to develop and modify the present study. The project presents an application of python used in this method Automated Machine Learning Pipeline for Traffic Count Prediction to traffic flow monitoring and road traffic analysis. The application is utilizing image-processing and vehicle recognition methods planned and personalized to the needs and constraints of road traffic analysis

Aim & Objective

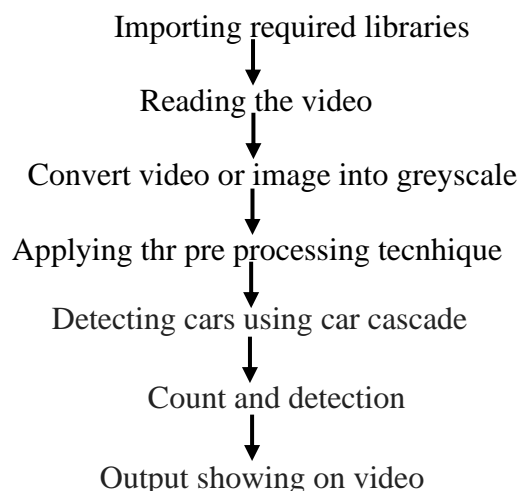
The present study is undertaken with the following objectives:

1. To compare the results with standard design service volumes and identify remedies.
2. To Planning, design and regulation of traffic.
3. To ensure the traffic controlling
4. To Establish properties and schedules for traffic improvements.

To analyze and optimize traffic systems, we first have to model a traffic system mathematically

Methodology

4.2) Flow chart of our methodology:-



First of all, you need to install OpenCV. We will do this tutorial using the completed Python programming language, so let's get started.-

Steps for Detection and Classification of Cars Using OpenCV

1. Import the relevant packages and start the network.
2. Retrieve frames from a video file.
3. Run the detection after pre-processing the frame.
4. Perform post-processing on the output data.
5. Count and track all cars on the route.
6. Save the completed data as a CSV file.

We required some libraries to import i.e

- 1) Open Cv
- 2) cv2.VideoCapture(video_path)

1) Open CV-

Open CV is an image processing library. It is designed to solve computer vision problems. OpenCV is a C/C++ library that is extended in Python.

2) NumPy-

The NumPy library is used to support multi-dimensional array, matrices, etc., in the Python programming language. It is an open-source numerical Python library.

NumPy provides:

1. Tools for integrating C/C++ and Fortran program
2. Sophisticated functions
3. A powerful N-dimensional array item.
4. Useful linear algebra, Fourier transform, and random number capabilities, and more.

Now create a file like **vehicle.py**

```
import cv2
import numpy as np
```

cv2.VideoCapture(video_path)-

If you use a video file as a source, then do the following

```
cap=cv2.VideoCapture('/home/python/OpenCV/vehiclesDetection/vehicles.mp4')
```

If using a webcam as a source, then. do the following

```
cap = cv2.VideoCapture(0)
#depends on your system 0, -1 or 1
```

Next, we will fetch the video from the internet that we will be working on. We will then resize the video and convert it into a NumPy array.

Example-

1) Reading video from these

```
#web camera
Cap=cv2.VideoCapture('video.mp4')
```

2) Grayscale

```
Grey=cv2.cvtColor(frame1,cv2.COLOR_BGR2GRAY)
Video.fromarray(grey)
```

3) Gaussian Blur

We will apply gaussianblur to remove the noise from the image and smoothing the image

```
blur = cv2.GaussianBlur(grey, (3, 3), 5)
Video.fromarray(blur)
```

4) Dilation

```
dilated = cv2.dilate(blur, np.ones((3, 3)))
Video.fromarray(dilated)
```

To fill the missing parts of the images whenever needed.

5) Detecting cars using car cascade



We need to detect multiple objects, i.e. cars, to use detectMultiScale.

```
car_cascade_src = 'cars.xml'
car_cascade = cv2.CascadeClassifier(car_cascade_src)
cars = car_cascade.detectMultiScale(closing, 1.1, 1)
```

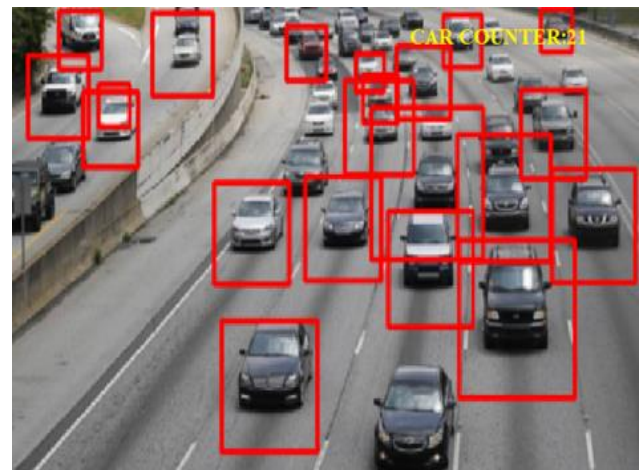
We will use the above-returned contours and draw a rectangle around detected cars. Here we will see that it will create the rectangle with a red/green boundary around every car it detects.

```
cnt = 0
for (x,y,w,h) in cars:
    cv2.rectangle(video_arr, (x,y), (x+w,y+h), (255,0,0), 2)
    cnt += 1
print(cnt, " cars found")
video.fromarray(video_arr)
```

```
1 import cv2
2 import numpy as np
3
4 #Web camera
5 cap = cv2.VideoCapture("video.mp4")
6 min_width_rect = 88
7 min_height_rect = 88
8 count_line_position = 550
9 #Initialize Subtractor
10 algo = cv2.bgsegm.createBackgroundSubtractorMOG()
11
12 def center_handle(x,y,w,h):
13     x1 = int(w/2)
14     y1 = int(h/2)
15     cx = x+x1
16     cy = y+y1
17     return cx,cy
18
19 detect = []
20 offset = 6
21 counter = 0
22
23 while True:
24     ret, frame1 = cap.read()
25     gray = cv2.cvtColor(frame1, cv2.COLOR_BGR2GRAY)
26     blur = cv2.GaussianBlur(gray, (5,5), 0)
27     #applying on algo
28     img_sub = algo.apply(blur)
29     dilated = cv2.dilate(img_sub, kernel, dst=None, anchor=None, iterations=1, borderType=cv2.BORDER_CONSTANT, borderValue=0)
30     kernel = cv2.getStructuringElement(cv2.MORPH_RECT, (5,5))
31     dilated = cv2.dilate(dilated, kernel)
32     dilated = cv2.morphologyEx(dilated, cv2.MORPH_CLOSE, kernel)
33     counterShape, h = cv2.findContours(dilated, cv2.RETR_TREE, cv2.CHAIN_APPROX_SIMPLE)
34     cv2.line(frame1, (25, count_line_position), (1200, count_line_position), (255, 255, 0), 2)
```

```
35
36 for (i,c) in enumerate(counterShape):
37     (x,y,w,h) = cv2.boundingRect(c)
38     validate_counter = (w>min_width_rect) and (h>min_height_rect)
39     if not validate_counter:
40         continue
41     cv2.rectangle(frame1, (x,y), (x+w,y+h), (0,255,0), 2)
42     cv2.putText(frame1, "Vehicle "+str(counter), (x,y-20), cv2.FONT_HERSHEY_SIMPLEX, 1, (255,255,0), 2)
43
44     center = center_handle(x,y,w,h)
45     detect.append(center)
46     cv2.circle(frame1, center, 4, (0,0,255), 1)
47
48 for (x,y) in detect:
49     if y<(count_line_position+offset) and y>(count_line_position-offset):
50         counter += 1
51         cv2.line(frame1, (25, count_line_position), (1200, count_line_position), (0,127,255), 1)
52         detect.remove((x,y))
53         print("Vehicle Counter: "+str(counter))
54         cv2.putText(frame1, "VEHICLE COUNTER: "+str(counter), (450,70), cv2.FONT_HERSHEY_SIMPLEX, 2, (0,0,255), 5)
55
56 # cv2.imshow("Detector", dilated)
57 cv2.imshow("video original", frame1)
58 if cv2.waitKey(1) == 13:
59     break
60 cv2.destroyAllWindows()
61 cap.release()
62
```

FINAL OUTPUT:-



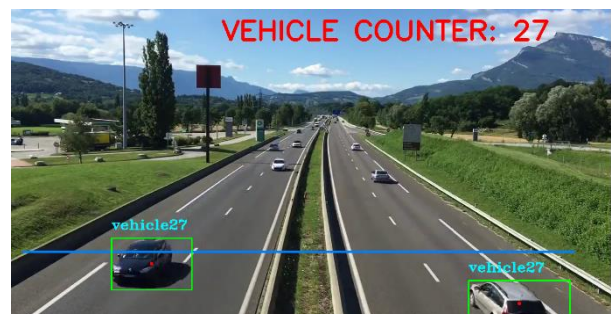


III. RESULTS AND DISCUSSIONS

Traffic volume study is carried out for vehicles and pedestrians separately or might be combined. It is carried out at number of selected positions along the road called as *traffic count stations*. We can classify the traffic according to this parameter.

SR.NO	Vehicle Category		Count	Total
01.	2 wheeler	Bicycles		
		Scooter/motorcycles		
02.	3 wheeler	Auto rikshaw		
03	4 wheeler	Car,van,jeep,		
		truck, tractor etc		
04.	Buses	Mini buses,PMT,ST		
05.	More Than 4 Wheel	6 WHEEL DRIVE, 8 WHEEL DRIVE		

Our expected output is classification of vehicles like bus, car, two wheeler, 3 wheeler etc.. But due to some technical issue we could not the conclude that final result



IV. CONCLUSION

Conclusion Based on our result, we obtained that the car/taxi are the majority of the vehicle at the every area. The second most road users will be the motorcycles, follow by pedal.

In this project, we have built an advanced vehicle detection and classification system using OpenCV. We can see that all the vehicles are successfully detected and have the counting numbers for different type of vehicle like car, buses, motorcycle, 3-wheeler. We have used the Visual studio and open cv to detect,counting and classify objects. And we learned about deep neural networks, file handling systems, and some advanced computer vision techniques.

Moreover, long period hours of observations should be performing in order to have a better understanding of the traffic volume at the junction. In conclusion, our objectives of the study are achieved. In this project, we have built an advanced vehicle detection and classification system using OpenCV. We can see that all the vehicles are successfully detected and have the counting numbers for different type of vehicle like car, buses, motorcycle, 3-wheeler. We have used the Visual studio and open cv to detect, counting and classify objects.



REFERENCES

1. A Review of Vehicle Detection Techniques for Intelligent Vehicles- Proff. Sriashika Addala
2. **Article-** Vision-based vehicle detection and counting system using deep learning in highway scenes
3. Machine Learning Based Traffic Volume Count Prediction- Proff, Chandrashekhar Raut
4. A technique on road traffic analysis using image processing- Priyanka Khanke, Prof. P. S. Kulkarni
5. Traffic analysis based on image processing- Harsha H. Sanap, P. A. Ghonge
6. Analysis & prediction on traffic volume in coimbatore bypass road Using python & tableau- Ms. R. Sushmitha, Mr. Anshulsaxena, Dr. K.M. Senthilkumar
7. <http://www.siamindia.com/>
8. <https://anaconda.org/anaconda/python>
9. <https://www.analyticsvidhya.com/>
10. <https://techvidvan.com/>
11. <https://geeksforgeekies.com>



Impact Factor: 8.423



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details