

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/341335122>

Traffic Detection Using OpenCV

Preprint · May 2020

DOI: 10.22541/au.158931021.15827522

CITATIONS

0

READS

1,595

3 authors, including:



Siddharth Bose

Xavier Institute of Engineering

98 PUBLICATIONS 826 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



HALF WARM ASPHALT MIXTURE USING POLYMER MODIFIED BITUMEN EMULSIONS [View project](#)

TRAFFIC DETECTION USING OPENCV

Teena Varma^{#1} Cline Colaco^{#2} Mandar Bagwe^{#3} Siddharth Bose^{#4}

Department of Computer Engineering Xavier Institute of Engineering, Mumbai University

¹teena.v@xavier.ac.in

²clinecolaco@gmail.com

³bagwemandar7@gmail.com

⁴sid99bose@gmail.com

Abstract — Traffic jams have become one of the biggest problems any metropolitan city faces in today's time. This paper suggests implementing a smart traffic detector using OpenCV. The density of vehicles on the road keeps increasing to a higher amount these days. In traffic signal, people waste much time particularly during the peak hours of the day. In order to solve this problem of high traffic pressure, it is indispensable to solve traffic congestion. The frustration that is faced by people during traffic jams could also lead to mishaps such as accidents. Thus an idea of monitoring the traffic congestion using real-time image processing techniques and via Central Neural Networks, through this software has been proposed. The theme is to determine the traffic density on each side of the road by calculating the number of vehicles at the traffic signal zone. In this, an input image of traffic surveillance is shown to our trained machine which declares whether there is traffic or not by judging via the number of vehicles seen. After the image acquisition, the image undergoes various image pre-processing, image enhancement, and edge detection techniques. This project has been customized to be used in the future to control the traffic signals as well as monitor violators and avoid inconvenience and accidents as much as possible.

1. INTRODUCTION

Object recognition technology in the field of computer vision to find the objects in the image or video sequences and selects them. In modern life, we have to face many problems, one of which is traffic congestion becoming more dangerous day by day. The efficient use of this generation's technology in tackling one of the biggest problems commuters face daily would make a huge positive difference. Traffic congestion is a very difficult problem. Instead of constructing more and more bridges, highways, and flyovers to curb traffic and provide a smooth flow of vehicles, we can simply enhance and modernize the system to work at a highly better efficiency.

Another way to detect traffic is by motion detection, installing lasers on both sides of the road, etc., which is tedious and involves a large amount of hardware. The method that we are implementing uses image processing techniques and the help of OpenCV to count the number of vehicles on the road and estimate the density. The number of vehicles found can determine the amount of congestion and necessary steps to take accordingly. It organizes the traffic in a smart way, in this way you can organize the traffic without needing a person to do it. The image is acquired from file explorer and undergoes several image processing techniques like image segmentation, grayscale conversion, noise removal, image enhancement, edge detection, background subtraction, canny edge detection, masking, HOG transform, grid search, and traffic density count. Based on vehicle density, the estimation of the amount of traffic is calculated and then it gets stored in the database.

2. LITERATURE SURVEY

The density of vehicles on the road keeps on increasing day by day. The need to control traffic congestion has become more advantageous these days. Increase in the traffic congestion generally results in high transportation cost and also wastage of time particularly during the peak hours of the day, excessive fuel consumption, etc, so the effort has to be made to put an end to such problems that keep on increasing on a day to day basis in our country. Even though several techniques are already available for detecting the density of vehicles on the road like a fuzzy-based controller, Radio Frequency Identification (RFID) concepts which include an RFID reader and tag for detecting the traffic, they are all much costlier and are difficult to implement in real-time. Efficient and cost-effective methods have to be incorporated for controlling traffic congestion.

Several algorithms are available to calculate the traffic density. Some are as follows: model-based, feature-based, region-based, probabilistic based, active contour-based. But these algorithms failed to concentrate on computational complexities and time factors. Thus these algorithms are not implemented in real-time. Immediate response should be considered as an important factor while working with traffic-related problems.

3. PROBLEM STATEMENT

The current road surveillance cameras and their viewing software can simply capture images when traffic rules are violated and for obtaining number plates and identification of drivers.

Through this project, we will be able to successfully detect as well as further monitor the traffic according to the number of vehicles as well as keep in track the rest of the attributes present such as road signs, footpath, lanes and pedestrians.

This will give the necessary boost required in the existing software of the surveillance cameras and is also useful for AI self driving cars and robotic machinery.

4. PROPOSED SYSTEM

The Traffic Detection performs the main functionality of determining the amount of traffic. OpenCV firstly converts the input image from BGR to grayscale, after which the picture undergoes masking, i.e. useless and redundant data is removed using suitable coordinates. Then canny edge detection helps in outlining and defining each distinct object seen in the image (refer Fig 3). The next step is to calculate the HOG (Histogram of Gradient) transform by determining x-sobel (x derivative) and y-sobel (y derivative) of the image (refer Fig 6).

The dataset then undergoes a grid search where most efficient gamma values are retrieved as the output. This is the basis on which the dataset is trained using an SVM model. This model is used to detect vehicles as well as classify them using an RBF kernel. The RBF kernel uses the most efficient gamma values and this is why it has a very high rate of accuracy. Linear kernel can also be used which uses C values but RBF kernel is preferred for its better accuracy.

Image processing involves issues related to image representation, compression techniques and various complex operations, which can be carried out on the image data. The operations that come under image processing are image enhancement operations such as sharpening, blurring, and brightening and edge enhancement. Traffic density of lanes is calculated using image processing which is done of images of lanes that are captured using digital camera.

Initially, the input image file is given for preprocess using file explorer

This file is processed by OpenCV which is used to preprocess the image and detect details (process mentioned above).

The statistical data is fed to the trained model which computes and delivers the result

The output is displayed to the user using GUI

Block Diagram is given as follows:

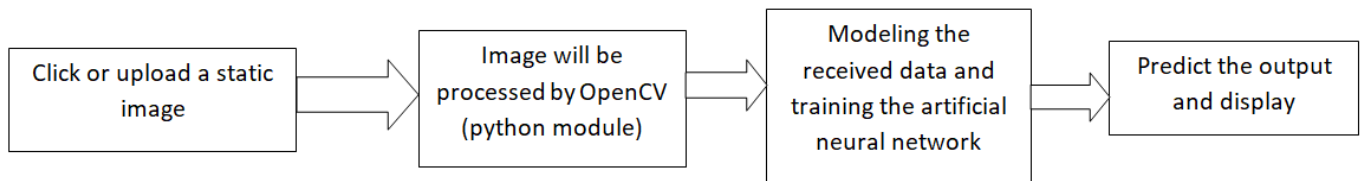


Fig. 1. Block Diagram

TESTCASES

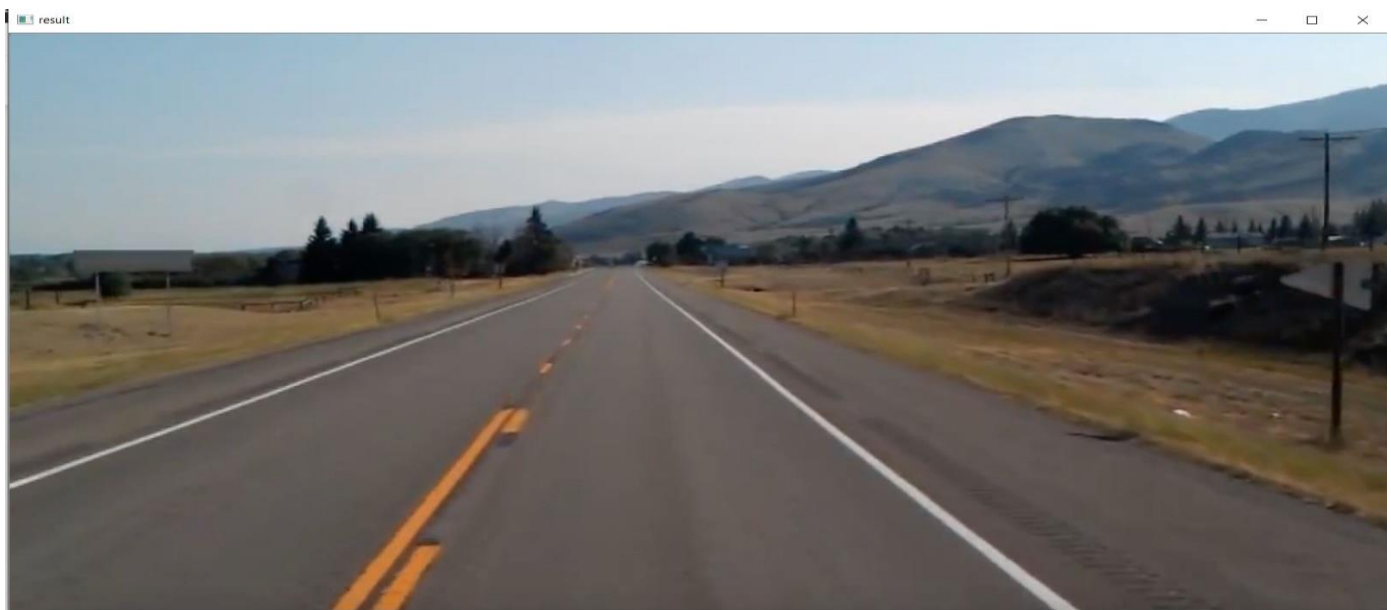


Fig. 2. Lane detection input



Fig. 3. Lane canny edge detection

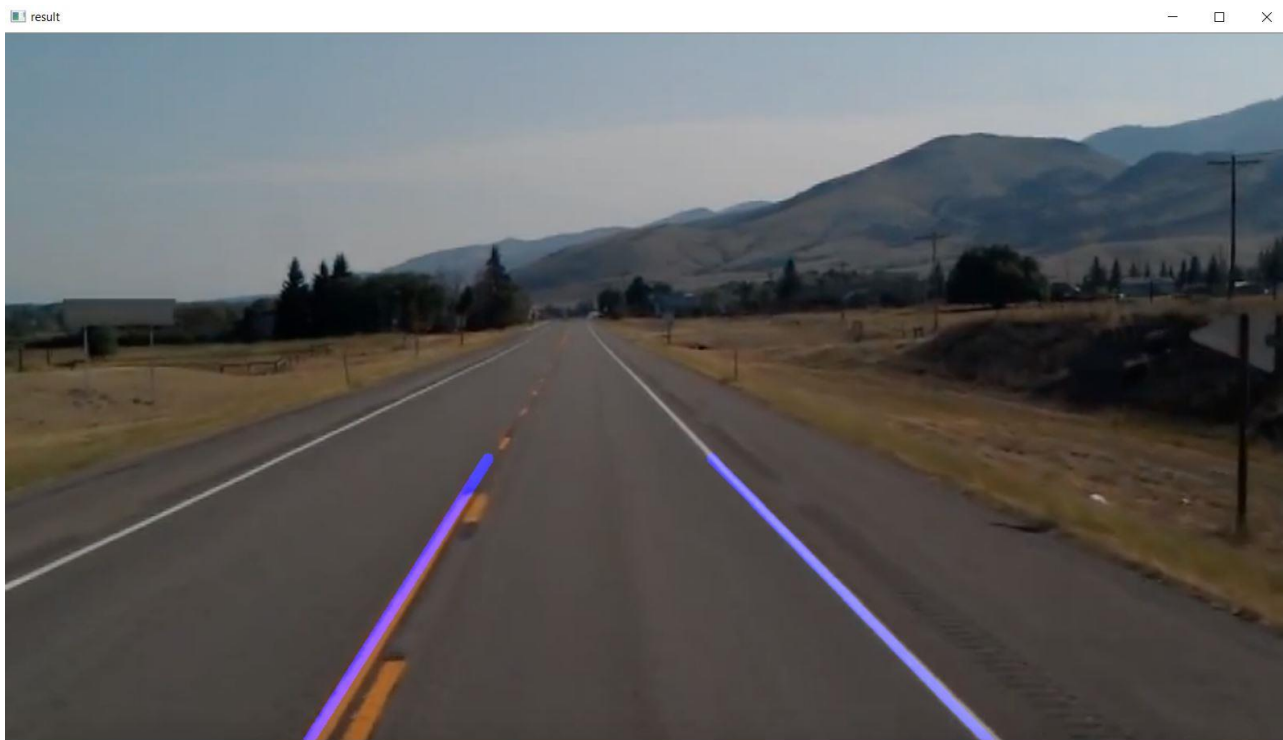


Fig. 4. Lane detection output



Fig. 5. vehicle detection input

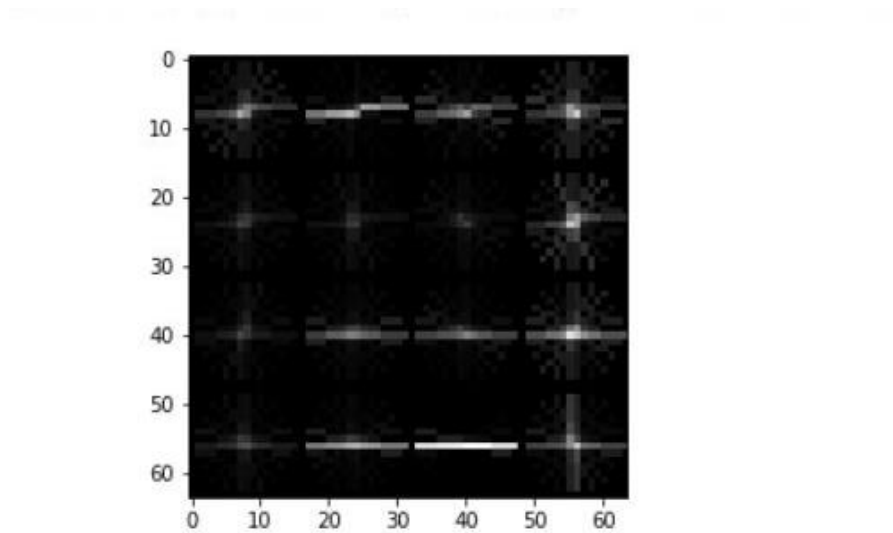


Fig. 6. HOG Transform

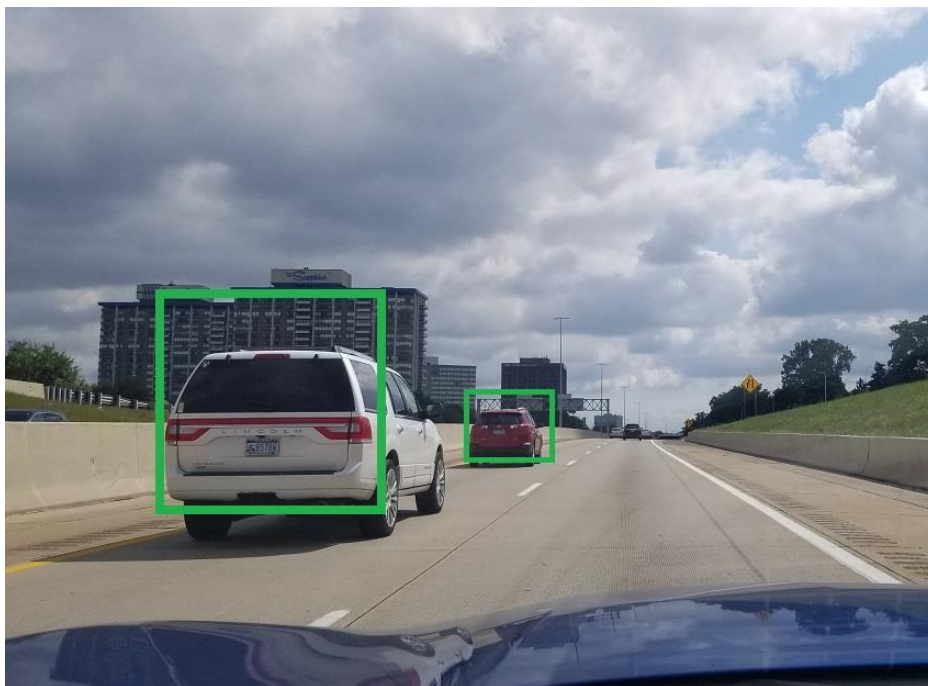


Fig.7. Vehicle detection

SIGN RECOGNITION

Traffic sign recognition collects information about road signs and helps the driver to make timely decisions, making driving safer and easier. We have picked (name) dataset from Kaggle and then detection and recognition of traffic signs from image sequences using the signs from the dataset. To obtain better shape classification performance, we used a linear support vector machine with the distance to border features of the segmented blobs. Support vector machine

based classification is employed for the detection and recognition of traffic signs. Our trained linear SVM model detects and recognizes every and all types of road signs. The algorithms are trainable to detect and recognize important prohibitory and warning signs from images captured. Recognition of traffic signs is implemented using a multi-classifier non-linear support vector machine with edge related pixels of interest as the feature.

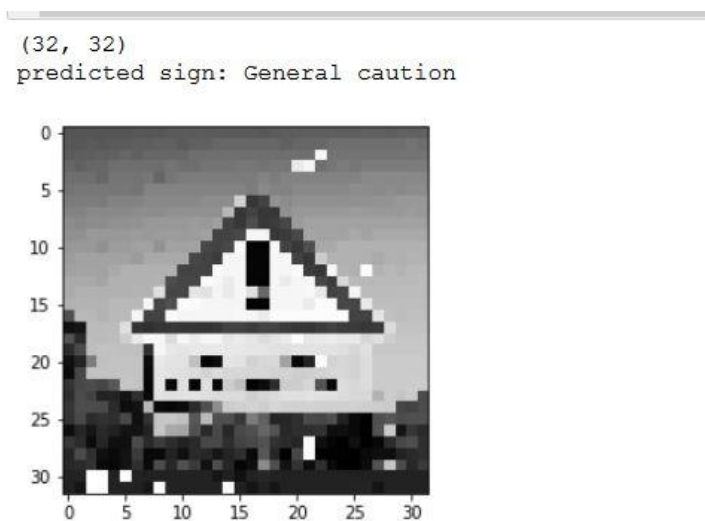
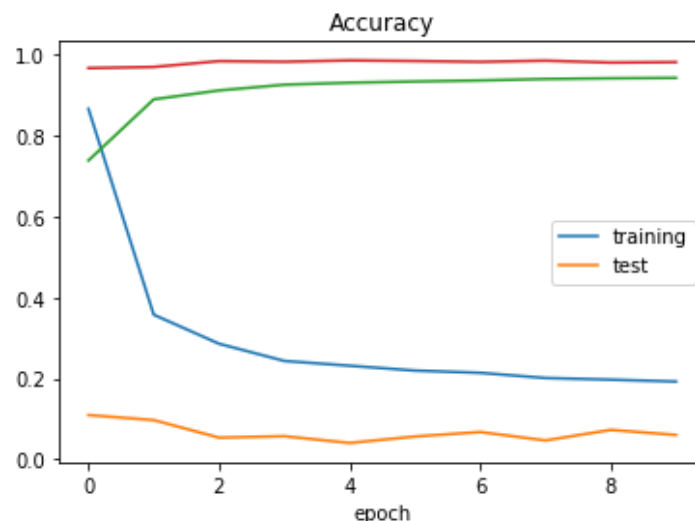


Fig. 9. Sign prediction

We have used the concept of Convolutional Neural Network as well as the assistance of Tensorflow. A Convolutional Neural Network (ConvNet/CNN) is a Deep Learning algorithm which can take in an input image, assign importance (learnable weights and biases) to various aspects/objects in the image and be able to differentiate one

from the other. Tensorflow on the other hand is an open source artificial intelligence library, using data flow graphs to build models. It allows developers to create large-scale neural networks with many layers. The image below is displaying our Tensorflow performance statistics.



Result of Test Cases

According to tests we performed, we were able to develop the model using OpenCV and with the help of our model. We are able to detect and recognize traffic in the input image as well as other navigation attributes. Thus, Traffic Detection is implemented successfully and above is the test results obtained.

ADVANTAGES AND APPLICATIONS

Advantages:

Convenient to use – This system is easy to use and is very simple to handle.
Effective – this system successfully observes and detects the amount of traffic as well as other attributes.
Joblib python package – PKL file is saved hence reduces training time of model.

FUTURESCOPE:

Continuous detection

We may also include features that enable the user to use a video as input and continuously detect and monitor traffic at regular time intervals and take further action based on total average traffic observed.

Expand the working scope

We can implement extra essential features specifically catering to the needs of self driving cars, security surveillance and robotics. Based on amount of traffic, the model could further suggest routes and steps to be taken as well as avoided.

Android Application

The current computer application can be converted into an android application and can be deployed in Google Play Store.

CONCLUSION

Thus, we have developed a traffic detector which is capable of detecting traffic from any road surveillance picture.

Graphical User Interface

GUI takes input as jpg file from user.

OpenCV

It observes the image and processes it for detection of vehicles and other attributes such as road signs, lanes, footpath, pedestrians, etc.

Model

The SVM model is trained to recognize vehicles as well as categorize them and detect whether there is traffic or not.

Applications:

Traffic Detection provides the following features/functions.

Input traffic image:

The input image file is pre-processed and analyzed using OpenCV library.

Detection of vehicles and all other attributes:

The model further keeps a count for number of vehicles and declares whether there is traffic or not.

Further training, testing and validation

Done to reduce over fitting (avoids focusing on more than necessary data by CNN and reduces time).

Using all of the above-mentioned tools we were successful in implementing our software.

REFERENCES

- [1] Chirag Thakkar, Rajesh Patil “Smart Traffic Control System Based on Image Processing (IJRASET)”
- [2] M.Sujatha, Renuga Devi, Bragadeesh.S.A “Traffic Congestion Monitoring using Image Processing and Intimation of Waiting Time(IJPAM)”
- [3] Al Hussain Akoum “Automatic Traffic using Image Processing (JSEA)
- [4] <https://bitbucket.org/jadslim/german-traffic-signs>
- [5] <https://stackoverflow.com/questions/51306862/how-to-use-tensorflow-gpu/51307381>
- [6] <http://www.cvlibs.net/datasets/kitti/>
- [7] <https://pypi.org/project/Flask/> (front end)