



---

## Traffic Detection Using OpenCV

Teena Varma, Cline Colaco, Mandar Bagwe and Siddharth Bose

*Department of Computer Engineering Xavier Institute of Engineering, Mumbai University, Mumbai, India*

---

**Key words:** Machine learning, image processing, OpenCV, neural networks, SVM, tensorflow

**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.

### Corresponding Author:

Teena Varma

*Department of Computer Engineering Xavier Institute of Engineering, Mumbai University, Mumbai, India*

Page No.: 18-22

Volume: 13, Issue 2, 2020

ISSN: 1997-5422

International Journal of Systems Signal Control and Engineering Application

Copy Right: Medwell Publications

---

## INTRODUCTION

Object detection and identification technology should be implemented for the modern world problems in our lives such as traffic. Tackling traffic efficiently and easily by bringing an advanced system into play is the perfect solution. 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

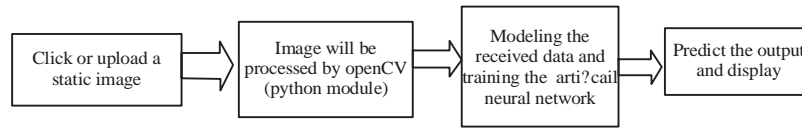


Fig. 1: Block diagram

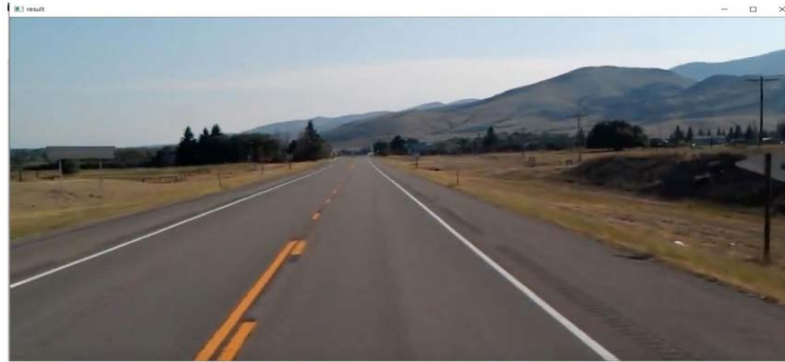


Fig. 2: Lane detection input

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 (Fig. 1).

**Literature review:** 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 Network Intrusion Detection System (NIDS). Network intrusion detection systems are placed at a strategic point or points within the network to monitor traffic to and from all devices on the network. It performs an analysis of passing traffic on the entire subnet and matches the traffic that is passed on the subnets to the library of known attacks<sup>[1]</sup>.

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.

**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<sup>[2]</sup>. 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 (Fig. 2).

## MATERIALS AND METHODS

**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<sup>[3]</sup>. Then canny edge detection helps in outlining and defining each distinct object seen in the image (Fig. 3-5). 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 (Fig. 6).

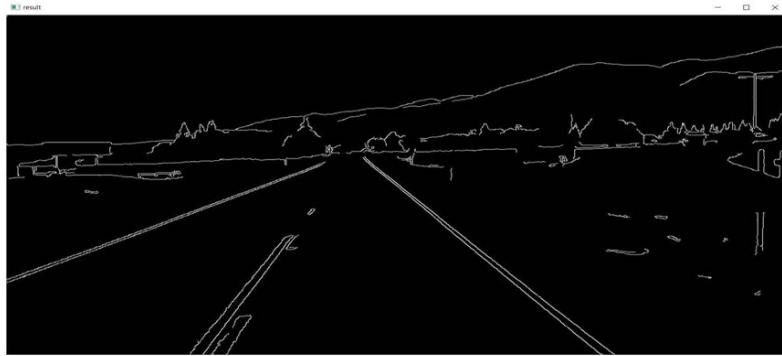


Fig. 3: Lane canny edge detection



Fig. 4: Lane detection output



Fig. 5: Vehicle detection input

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

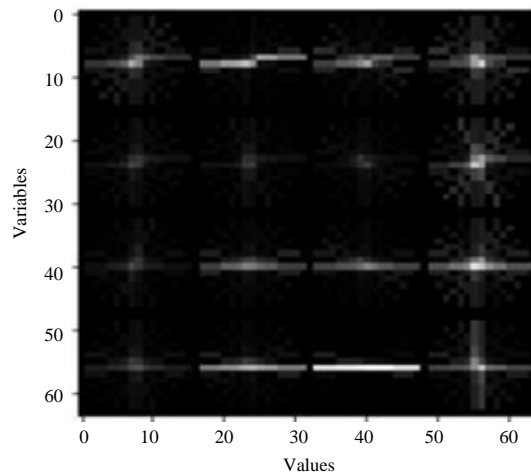


Fig. 6: HOG transform

very high rate of accuracy. Linear kernel can also be used which uses C values but RBF kernel is preferred for its better accuracy<sup>[4,5]</sup>.



Fig. 7: Vehicle detection

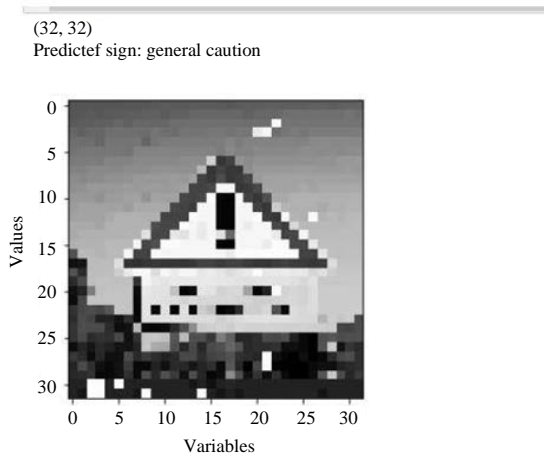


Fig. 8: Sign prediction

Image processing helps us in understanding and analyzing the image properties and the objects that are seen in the image. Sharpening, blurring and brightening and edge enhancement are some of the functions of image processing, this making it easier for our trained model to understand and work on the image. Through this process the density of traffic is calculated by retrieving the total amount of vehicles present in the image given.

Initially, the input image file is given for preprocessing using file explorer (Fig. 7 and 8). This file is processed by OpenCV which is used to pre-process 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.

**Test case**

**Sign recognition:** Traffic sign recognition helps in detecting and identifying road signs for the convenience of the driver. We have picked (name) dataset from Kaggle

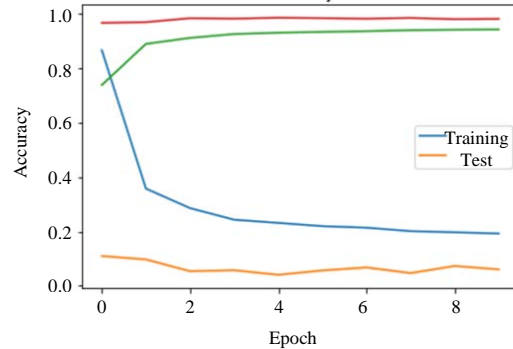


Fig. 9: Teusorflow performance statistics

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.

We have used the concept of Convolutional Neural Network as well as the assistance of Tensorflow. We have made use of tan H activation function and ADAM optimizer. 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. Figure 9 is displaying our Tensorflow performance statistics.

**RESULTS AND DISCUSSION**

**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. Jolib python package-PKL file is saved hence reduces training time of model.

### **Future scope**

**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.

### **IMPLICATIONS**

**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**

01. Thakkar, C. and R. Patil, 2015. Smart traffic control system based on image processing. *Int. J. Res. Applied Sci. Eng. Technol.*, 5: 572-580.
02. Sujatha, M., D. Renuga and S. Bragadeesh, 2017. Traffic congestion monitoring using image processing and intimation of waiting time. *Int. J. Pure Applied Math.*, 115: 239-245.
03. Al Hussain, A., 2017. Automatic traffic using image processing. *J. Software Eng. Appl.*, 10: 765-776.
04. Anonymous, 2020. How to use TensorFlow GPU?. Stack Exchange, Inc., New York, USA.
05. Geiger, A., P. Lenz, C. Stiller and R. Urtasun, 2015. The KITTI vision benchmark suite. Karlsruhe Institute of Technology, Karlsruhe, Germany.