# Reducing Unnecessary Wait Time at Traffic Lights Using OpenCV

Author: Sumit Khatri
Designation: CSE Student,
Institute: Maharaja Agrasen Institute Of Technology.

**ABSTRACT**

Traffic Light Optimization uses Machine Learning to find an efficient solution in order to reduce the unnecessary waiting time on the traffic signals. It works by changing time which is a variable parameter relative to the traffic congestion on a traffic signal by using a computer software program. This project aims to minimize the time spent by people on traffic signals by adjusting the timer of green and red lights depending upon the density of
traffic at a particular time on that traffic signal.

**KEYWORDS:** Video Surveillance, OpenCV, Computer Vision, Vehicle Detection

## 1. INTRODUCTION

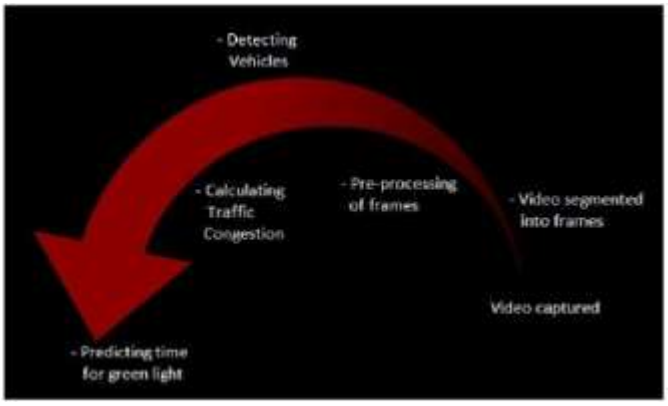
One of the most ignored problems in the world is of Traffic Congestion. It not only wastes time of people that could be spent doing some other productive work but also leads to an incredible amount of fuel wastage and increased pollution levels, both of which are detrimental to the environment. Computer Vision is a really powerful tool that can be utilized to analyze complicated scenarios. It can be used to analyze and identify the patterns in which the traffic is flowing and the timing for which a particular traffic light is kept on. The proposed model can be used effectively to optimize the traffic signals by taking a look at the traffic density at a particular signal at that time.

Video based vehicle detection have come along way and have been playing a crucial role in real-time traffic management over the past decade. Video based traffic monitoring systems work more effectively and efficiently over the traditional methods like the one known as loop detectors.

This usage of video feed for Traffic Light Optimization can also have other benefits as the
feed can then be used to obtain vehicle classifications, lane change detection etc. Vehicle data can be used for a variety of transportation projects which can vary from intelligent transportation systems to autonomous vehicle driving. Magnetic loop detectors are used currently to detect the total number of vehicles passing over them.

Vision based traffic monitoring systems are the need of the hour as some large metropolitan cities require other vehicle related data too for instance, keeping a check on the speed of the vehicles, lane changes, vehicle classification, the vehicle can be a stolen one, so as to notify the police of the last seen location of the vehicle.

Our system comprises a camera that is installed on a pole or some other tall structure, constantly keeping an eye on the traffic scene below.




## 2. Literature Survey

Different Types of approaches were already made earlier to develop systems which can detect the vehicles, count them and classify them and the video feed generated can be used for traffic surveillance. This section takes a look at such systems and the knowledge of the methods that were used to develop these systems.

The key technique of video-based vehicle detection belongs to a classic problem of motion segmentation. One of the widely used techniques to identify moving objects (vehicles) is background subtraction or background learning

Nilakorn et al. [1], proposed a vehicle detection which uses a number of steps for background subtraction and object detection and then goes on to use OpenCV techniques such as thresholding, hole-filling, and adaptive morphological dilation to remove the noise and enhance the foreground objects in a particular frame from a video.

Some Studies, like the one done by D. A. Forsyth, J. Ponce [2] and D. G. Lowe [3] proved that by using feature vectors from the image region can be really efficient for the goal of vehicle detection. D. A. Forsyth, J. Ponce[2] also tried to represent that the estimation for accurate vehicle dimension can be done by using a set of coordinate mapping functions.

Bhushan et. al. [4] went a step further and proposed a vehicle detection method which uses morphological operations including binarization, top-hat processing, edge detection and masking operations. The proposed system however fails to produce good results in cloudy environments.

Habibu Rabiu [SJ proposed a vehicle detection and their classification for urban intersections with high density traffic. This system uses background subtraction and Kalman filter algorithms to detect and track the vehicles and their positions. Linear Discriminant Analysis Classifier is used for proper classification of vehicles.

Abhijeet Suryatali and V.B. Dharmadhikari in [6] proposed a system that uses Computer Vision Based vehicle detection which counts the vehicles and also classifies them into heavy and lightweight vehicles. Object detection is done by using Kalman Filter for background subtraction and then using the OpenCV library to detect the vehicle in that particular frame .

## 3. Working Of Model

In this model, we are working on the idea to hold the green light according to the amount of traffic present on the road i.e.traffic congestion on the road. So to calculate the amount of traffic on the road we first need to detect vehicles in the frame and for this detection we are using OpenCV library of python which majorly works on images and videos.

We choose OpenCV library to detect the vehicles because it works as an unsupervised learning technique which does not require any kind of pre-trained model for detection. In our model we first capture frames from the video and on these frames we use OpenCV for detection. After detection we also want our model to show a bounding box on the vehicle detected as shown in the image below.

The steps used to perform the required functionality are:

- Background Subtraction
- Image Thresholding
- Morphological Transformation

- Contour Finding
- Congestion sensed signal

### 3.1 Background Subtraction

Background subtraction (BS)[7] is a common and widely used technique for generating a foreground mask (namely, a binary image containing the pixels belonging to moving objects in the scene) by using static cam eras. As the name suggests, BS calculates the foreground mask performing a subtraction between the current frame and a background model, containing the static part of the scene or, more in general, everything that can be considered as background given the characteristics of the observed scene.

Foreground objects= current frame - background layer

So in our scenario the background i.e. road is subtracted from the frame and only fore ground i.e. vehicles are left in the frame (as shown in the image below) but this frame contains noise which needs further filtration.

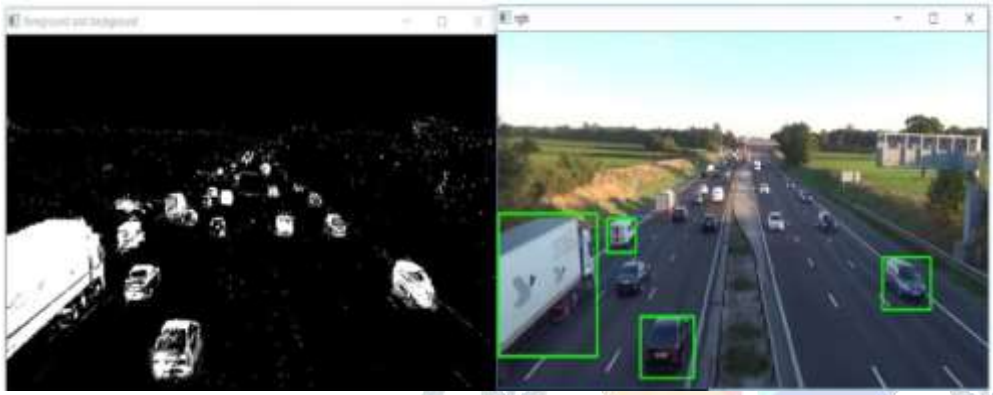


Figure: Background Subtraction

### 3.2 Image Thresholding

] is a technique in OpenCV, which is the assignment of pixel values in relation to the threshold value provided. In thresholding, each pixel value is compared with the

threshold value. If the pixel value is smaller than the threshold, it is set to 0, otherwise, it is set to a maximum value (generally 255). Thresholding is a very popular segmentation technique, used for separating an object considered as a foreground from its background. In Computer Vision, this technique of thresholding is done on grayscale images. So initially, the image has to be converted in grayscale colour space. In our model we use binary thresholding which uses the same threshold all over the frame.

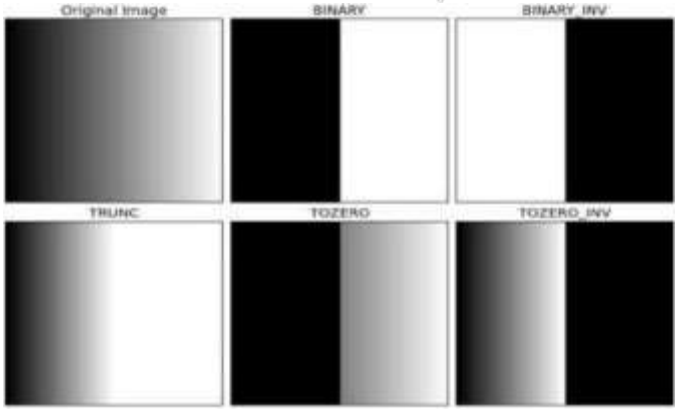



Figure: Image Thresholding

### 3.3 Morphological Transformation

Morphological trans formations [ 9] are some simple operations based on the image shape. It is normally performed on binary images. Two basic morphological operators are Erosion andDilation. In erosion, we are just omitting the boundaries of the front image or the object image that is in the process we are thinning the object .Here we use cv2.erode() function . In the dilation process we are just going to thick the boundaries of a binary image. The bright area of the binary image dilates around the black regions of the background. It' s actually the reverse process of Erosion. Here we use cv2.dilate() function.

### 3.4 Finding Contours

Contours[10] are defined as the line joining all the points along the boundary of an image
that are having the same intensity.Contours come handy in shape analysis, finding the size of the object of interest, and object detection. OpenCV has findContour() function that helps in extracting the contours from the image. It works best on binary images, so we should first apply thresholding techniques, morphological operations, etc. In our case each contour that is present on the frame is a vehicle on the road. Now as we detected the vehicle we have to just build a bounding box on it so that it can be clearly visible in the actual frame.

Figure: Contour with bounding box

### 3.5 Congestion Sensed Signal

Now the approach on which the whole model works is, we have only green light which we can optimize to reduce the waiting time as re light time depends on the green light time ofrest three sides. And to optimize green light we calculate the area of the contours we have just found in the frame and if that area is less than 20% of the frame area then we which
' OFF' the green light. While checking for the traffic congestion we also keep in considerationthe case of starvation of any of the four sides, and to remove it we maintain a minimum andmaximum threshold time for which the green light can we 'ON'.In this way they keep ' ON' green light when the traffic area is more than 20% of the frame area and the time is between minimum and maximum threshold, if any of these conditions fail in any of the captured frame then the green light is turned 'OFF' and the cycle goes on.

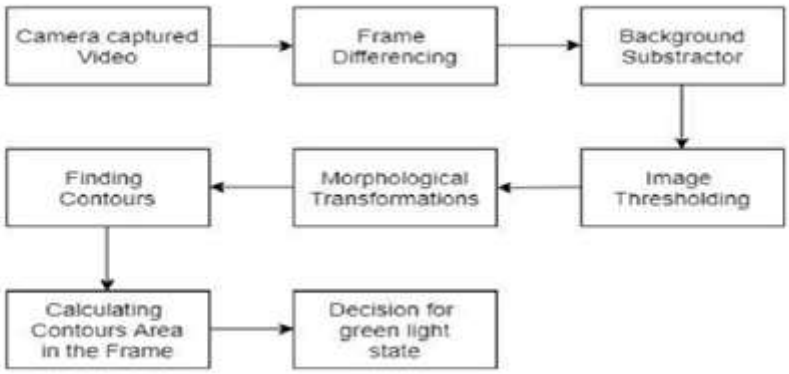



Figure: Block Diagram of the proposed model

## 4. Conclusion and Future Work

The implementation of the proposed solution is done in Python using OpenCV. Video feeds of traffic are ta ken from a variety of sources and this can further be expanded to work on live video streams. The accuracy can either decrease or increase depending upon the camera angle. Most accurate predictions are seen with videos that are taken where the vehicles are in direct Line of Sight of the camera.

Fu rt her improvements can be done by using Neural Networks as an approach to detect vehicles with better accuracy. A next step would be to make it work better in places withdim lighting at night.

## 5. References

[1] Seenouvong, Nilakorn & Watchareeruetai, Ukrit & Nuthong, Chaiwat & Khongsomboon,Khamphong & Ohnishi, Noboru. (2016). A computer vision based vehicle detection and counting system. 224-227. 10.1109/KST.2016 .7440510.

[2] D. A. Forsyth, J. Ponce, "Computer Vision: A Modern Approach," Prentice Hall, 2003.

[3] D. G. Lowe, " Distinctive Image Features from Scaled-Invariant Keypoints," International Journal of Computer Vision, pp. 91-110, 2004.

[4] Nemade, Bhushan. (2016). Automatic Traffic Surveillance Using Video Tracking.Procedia Computer Science. 79. 402-409. 10.1016/ j.pro cs.2016.03.052.

[5] Rabiu, Habibu & Bashir, Hassan. (2015). Intelligent Traffic Light System for Green Traffic Management.

[6] Abhijeet Suryatali, V. B. Dharmadhikari {2015), Engineering International Conference on Circuits, Power and Computing Technologies [ICCPCT-2015]

[7] https:/ / open cv-python-tutroals.readthedocs.io/en/latest/py_tutorials/py_video/py_bg_subtraction/py_bg_subtraction.html

[8] https://opencv-python-tutrolas.readthedocs.io/en/latest/py_tutorials/py_imgproc/py_thresholding/py_thresholding.html

[9] https:/ / docs.opencv.org/ master/d9/d61/tutorial_py_morphological_ops.html

[10]https:/ / docs.opencv.org/ mast e r/ d4/ d73/ tutorial_py_contours_begin.htm;