Vehicle Detection, Counting and Classification Using OpenCV By Contour Extraction

Ujjwal Agarwal(190921)
Ashok Kumar Jyani(190194)
Sparsh Sihotiya(190860)

Abstract

With the drastic increase in the use of motor vehicles traffic control and analysis has become an important problem. Analysis of traffic can include the number of vehicles passing through a given area in some timespan or the type of vehicles passing through or the average speed of vehicles at different points etc. Vehicle detection, tracking, classification and counting is very important for various applications such as highway monitoring, traffic planning, toll collection etc. . There have been multiple studies on this topic and different methods to approach this problem. One way to do is through a vision-based approach which we have done. Another possible method could be to use sensors which even though are effective to a large extent, have other issues such as high costs or regular need for maintenance and are difficult to modify comparatively. In our project we capture frames of a pre-recorded videos to perform **background subtraction** to detect and count the number of vehicles passing. The extracted background is further used classification where we have used the **contour area** as a comparison measure to classify vehicles as **Cars/Bikes/Heavy Motor Vehicles(HMV).**

Introduction

Traffic counts, speeds, and vehicle classification are critical pieces of information for a wide range of transportation applications such as transportation planning and modern intelligent transportation systems. Sensors are used to estimate traffic parameters in 'Traffic Monitoring' and 'Information Systems' connected to vehicle classification. Magnetic loop detectors or wireless sensor networks are currently used. To count vehicles travelling over them, magnetic loop detectors or wireless sensor networks are commonly utilized, which are big, expensive, and difficult to install without disrupting traffic. Compared to older technologies, vision-based video surveillance systems provide a number of advantages. A far broader set of traffic characteristics, such as vehicle classes and lane change, are available in addition to vehicle counts. A classification and counting system, can provide critical information to a decision-making agency. In our project we can identify and classify vehicles in numerous lanes and in any traffic flow direction.

OpenCV

OpenCV is a cross-platform library using which we can develop real-time computer vision applications.
By using it, one can process images and videos to identify objects, faces, or even handwriting of a human.

OpenCV is extensively used in companies, research groups, and governmental bodies. There are lots of applications which are solved using OpenCV, some of them are listed below

- face recognition
- Automated inspection and surveillance
- Vehicle counting on highways along with their speeds
- Robot and driver-less car navigation and control
- object recognition

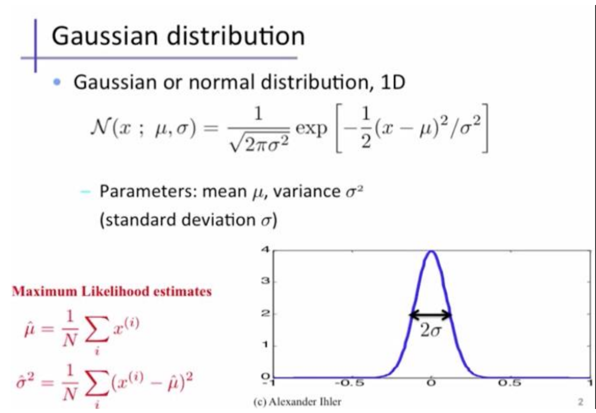
Related Work

For many years tracking moving vehicles in video streams has been an active area of research in computer vision. For example, a real time system was created that uses a feature-based method along with occlusion reasoning for tracking vehicles in congested traffic scenes. To handle occlusions, instead of tracking entire vehicles, vehicle sub-features are tracked. Another moving object recognition method uses an adaptive background subtraction technique to separate vehicles from the background. The background is modeled as a slow time varying image sequence, which allows it to adapt to changes in lighting and weather conditions. Other popular video-based traffic counting systems use high-angle cameras to count traffic by detecting vehicles passing digital sensors. As a pattern passes over the digital detector, the change is recognized, and a vehicle is counted. The length of time that this change takes place can be translated into speed estimates. When driving in the dark environment, drivers normally turn on the headlights to obtain a clear vision on the road. These headlamps produce illumination on the ground and this region will be classified as moving object. This headlight detection method includes high intensity region detection. Some researchers proposed a video based real-time vehicle counting system using optimized virtual loop method. They used real time traffic surveillance cameras deployed over roads and compute how many vehicles pass the road. In this system counting is completed in three steps by tracking vehicle movements within a tracking zone called virtual loop. Despite the large amount of literature on vehicle detection and tracking, there has been relatively little work done in the field of vehicle classification. This is because vehicle classification is an inherently hard problem. Moreover, detection and tracking are simply preliminary steps in the task of vehicle classification.

Background Subtractor MOG

It is a Gaussian Mixture-based Background/Foreground Segmentation Algorithm. It was introduced in the paper "An improved adaptive background mixture model for real-time tracking with shadow detection" by P. KadewTraKuPong and R. Bowden in 2001. It uses a method to model each background pixel by a mixture of K Gaussian distributions (K = 3 to 5). The weights of the mixture represent the time proportions that those colors stay in the scene. The probable background colors are the ones which stay longer and more static.

Gaussian distribution (also known as normal distribution) is a bell-shaped curve, and it is assumed that during any measurement values will follow a normal distribution with an equal number of measurements above and below the mean value.



Mixture-of-Gaussians (MoG) background model is widely used to segment moving foreground for its effectiveness in dealing with gradual lighting changes and repetitive motion of leaves. However, the MoG technique appears to have two flaws: the first is that it takes a long time to build the background model, and the second is that it can't handle situations when objects enter the scene, stay for a long time, and then leave, as is common in subways, bus stations, and railway stations. When the objects stay longer, they would gradually merge into the background, which would affect the follow-up application, such as crowd counting or event analysis. The values of a particular pixel is modelled as a mixture of adaptive gaussians as multiple surfaces appear in a pixel and lighting can change.

Methodology

The first step is to extract images from the video frame by frame. Then we try to understand the background in terms of how it differs from the foreground. Furthermore, as previously said, our research is based on a video feed from which we extract frames and learn about the background. Moving objects are regarded as the foreground while static objects can be called the background in a traffic scenario filmed with a static camera set on the roadside. Image processing algorithms are used to learn about the background using the above-mentioned technique.

This module consists of three steps, background subtraction, image enhancement and foreground extraction. Background is subtracted so that foreground objects are visible. This is done usually by static pixels of static objects to binary 0. After background subtraction image enhancement techniques such as noise filtering, dilation and erosion are used to get proper contours of the foreground objects. The final result obtained from this module is the foreground.

Background subtraction, image enhancement, and foreground extraction are the next steps. The background is removed so that the foreground objects may be seen. This is commonly accomplished by converting static pixels of static objects to binary 0. Image enhancement techniques such as noise filtering, dilation, and erosion are employed after background subtraction to acquire proper contours of the foreground object. We also convert the video to gray scale as it reduces the computational complexity .

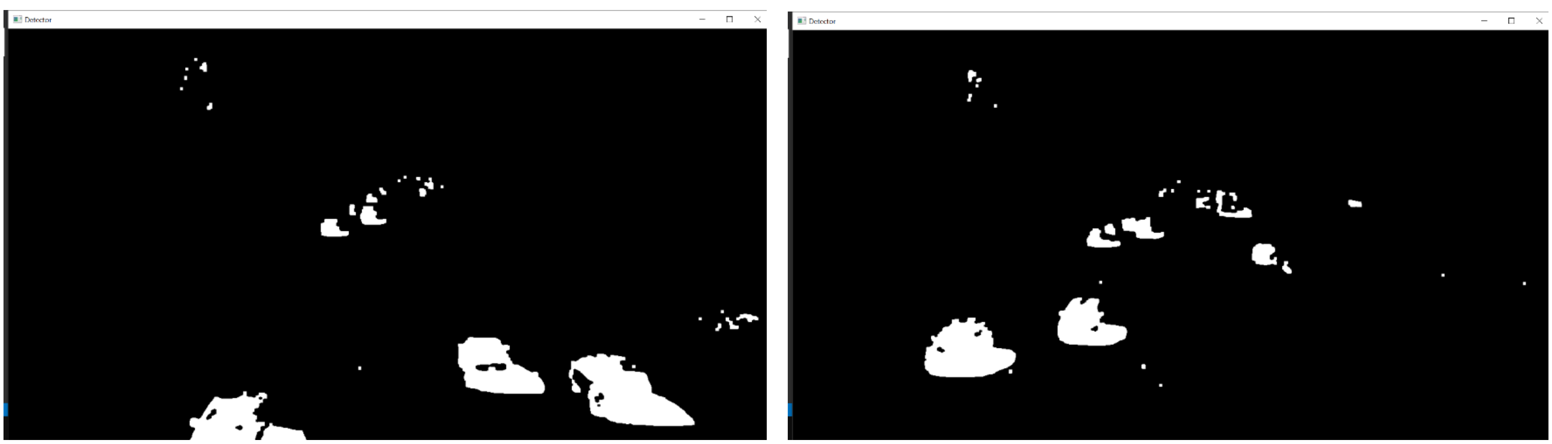
To reduce noise, we convolve the image with a **gaussian filter**. Gaussian smoothing is used as a pre-processing stage in computer vision algorithms in order to enhance image structures at different scales. The Gaussian blur is a type of image-blurring filter that uses a Gaussian for calculating the transformation to apply to each pixel in the image. The formula of a Gaussian function in one dimension is

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{x^2}{2\sigma^2}}$$

Next, we do further image pre-processing by using **dilation and erosion**. Dilation adds pixels to the boundaries of objects in an image, while erosion removes pixels on object boundaries. The number of pixels added or removed from the objects in an image depends on the size and shape of the *structuring element* used to process the image. In the morphological dilation and erosion operations, the state of any given pixel in the output image is determined by applying a rule to the corresponding pixel and its neighbors in the input image. The rule used to process the pixels defines the operation as a dilation or an erosion. In our project we have **used 'MORPH_ELLIPSE'** as the structuring element. We do dilation to join any broken parts in our image or to increase the pixel size as erosion shrinks the image.

Below are 2 images of the dilation done in our project:

Dilation

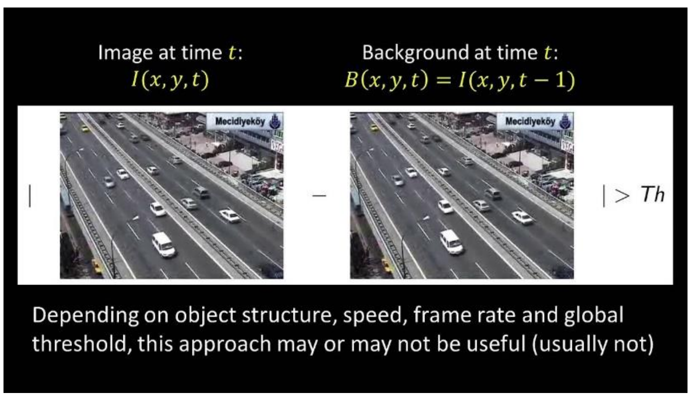


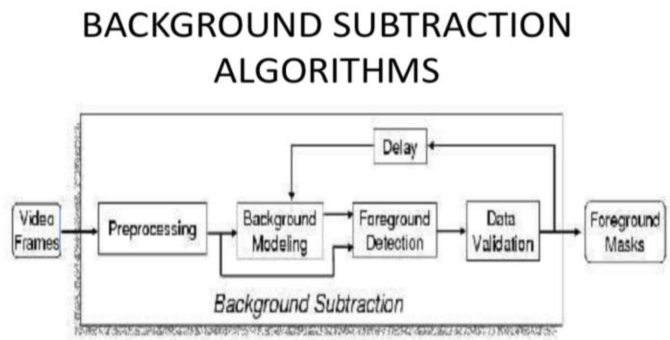
After applying foreground extraction proper contours are acquired. Features of these contours such as centroid, aspect ratio, area, size and solidity are extracted are highly used for classification of vehicles. We extract the contours of objects by using the '.findcontours' function. Contours are a useful tool for shape analysis and object detection and recognition. For the retrieval mode we have used **CV_RETR_TREE** which retrieves all of the contours and reconstructs a full hierarchy of nested contours. For the contour approximation method, we have used **CV_CHAIN_APPROX_SIMPLE** which compresses horizontal, vertical, and diagonal segments and leaves only their end points.

In brief we performed a series of tasks that are converting frame to gray scale, applying background mask, subtracting mask, performing binary threshold, morphology using erosion and dilation, gaussian blur and applying masked data to the frame. Contours were detected after these operations.

Background subtraction

Background subtraction is a critical pre-processing stage in the design of any visual surveillance system since it determines the correctness of the entire object classification process. If we already have a backdrop image, such as a picture of a building or a road, background subtraction may be a simple task. Background images can be deleted, and foreground objects can be obtained in the instances described above, however the situation varies most of the time. The backgrounds can be dynamic (changing all the time) or the scene's initial information may not be provided. Furthermore, because objects in the video move for ex. people or cars, background subtraction becomes more challenging. If the objects in the video have shadows, typical background subtraction will detect the shadows as foreground objects as well.





Flow diagram of a generic background subtraction algorithm.

Several algorithms have been introduced for the situations mentioned above, some of them are implemented in OpenCV such as BackgroundSubtractorMOG which use Gaussian distributions to create the model of the background in the image. It uses about 3 to 5 Gaussian distributions for this purpose. Another background subtraction implemented in OpenCV is called BackgroundSubtractorGMG which combines the background image estimation technique with Bayesian segmentation. The algorithm used in the implementation of the project is called BackgroundSubtractorMOG.

Contour Extraction

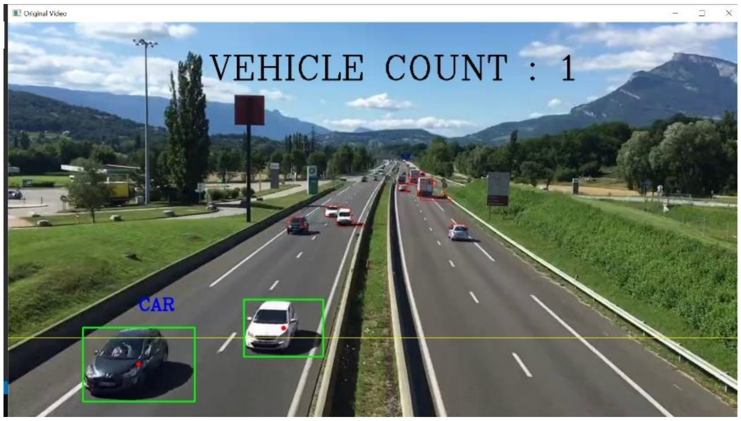
Contours are the boundaries of the shape which are used for the shape detection and recognition. The accuracy of the process of finding the contours can be defined as the canny edge detection performed on a binary image. OpenCV provide cv2.findContours() method to find the contours.

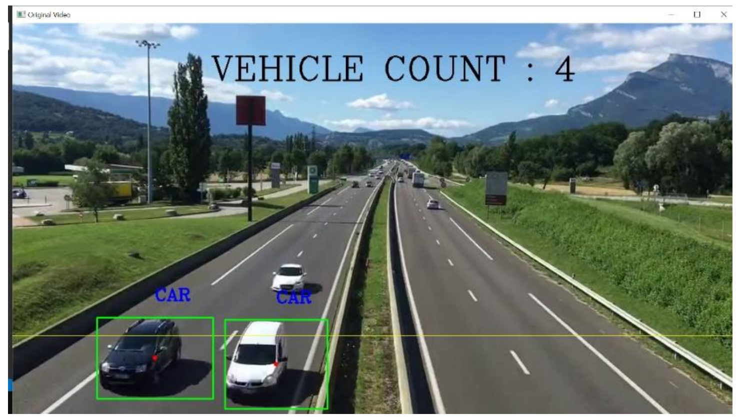
Counting Vehicles

In our project we have counted the number of vehicles by checking if the center of the rectangle bounding and tracking the object lies within a range of an imaginary line (shown as a yellow color line at the bottom). When the center is within the range it is noticeable as the line changes color and the main counter at the top of the screen increases. This method helps to reduce the number of false positives as we avoid cases where sometimes rectangles show up on non-vehicle objects. For ex. In the video we have used there is a tree at the left which is detected and a rectangle shows up around its top at times.

Below are 2 images of our project:

Vehicle counting





Classification using Contour Comparisons

This approach extracts a section of the contour and compares it to previously anticipated values to determine whether the vehicle is a car, a bike, or an HMV. The contours of an image are basically all connected pixels. After background reduction and detection of foreground items, contours of these foreground objects are discovered. We have used findContours() method to detect the contours of the foreground object. The method gives a list of all the contours in the image. It is necessary to choose the biggest contour in the image. Hence, a minimum limit on the width and height of the contours is defined so that bigger contours can be chosen. Following the selection of these contours, a number of features of these contours are retrieved and used to classify the cars. Area, solidity, and aspect ratio are some of these qualities. Particular attention is given to areas of contours that are compared to the vehicles' expected values. For classification the following values are assumed:

1. **Area of contour < 3000 - Bike**
2. **10000 < Area of contour < 20000 - Car**
3. **Area of contour > 40000 - Heavy Motor Vehicle**

Conclusion

There is a great amount of potential applications of vehicle identification and tracking on expressways and highways due to the development in expressways, highways, and traffic congestion. We have developed a vision-based system for effective detection and counting of vehicles on the road in this study. A drawback of the project is that setting the region of interest requires human intervention. For

vehicle counting, the user must draw an imaginary line where the center of the contours intersects thus, the accuracy is dependent on the human supervisor's opinion.

Furthermore, because the camera angle has an impact so camera calibration techniques could be employed to detect vehicles with an increased efficiency. Also, the project would be unable to detect automobiles at night since it requires foreground objects to be visible in order to extract contour properties and features for classification.

References

- Neeraj K. Kanhere, Stanley T. Birchfield, Wayne A. Sarasua, Tom C. Whitney, Real-time detection and tracking of vehicle base fronts for measuring traffic counts and speeds on highways", Transportation Research Record, No. 1993. (2007)
- Sheeraz Memon, Sania Bhatti, Liaquat A. Thebo, Mir Muhammad B. Talpur, Mohsin A. Memon, " A Video based Vehicle Detection, Counting and Classification System", International Journal of Image, Graphics and Signal Processing(IJIGSP), Vol.10, No.9, pp. 34-41, 2018.DOI: 10.5815/ijigsp.2018.09.05
- Uke, Nilesh & Thool, Ravindra. (2012). Moving Vehicle Detection for Measuring Traffic Count Using OpenCV. Journal of Automation and Control Engineering. 10.12720/joace.1.4.349-352.