Vehicle Detection and Counting using Machine Learning

Harshitha LC[1], Sushanth MU[2], Pavan P[3],Poornashree M[4]
[1,2,3,4]*Maharaja institute of technology*

(E-mail:[3]pavanpgowda123@gmail.com)

Abstract— A vehicle detection and counting system plays an important role in an intelligent transportation system, especially for traffic management. By analyzing the traffic video sequence recorded from a video camera, this paper presents a machine learning based technique for vehicle detection and counting. The proposed method uses openCV along with a python wrapper called cv2 technique to find foreground objects in a video sequence. In order to detect moving vehicles more accurately, we have used an SSD with Mobilenet model which is developed on TensorFlow. Finally, we have used ROI technique to count number of vehicles in a video. Experimental results show that the vehicle counting and detection has better accuracy.

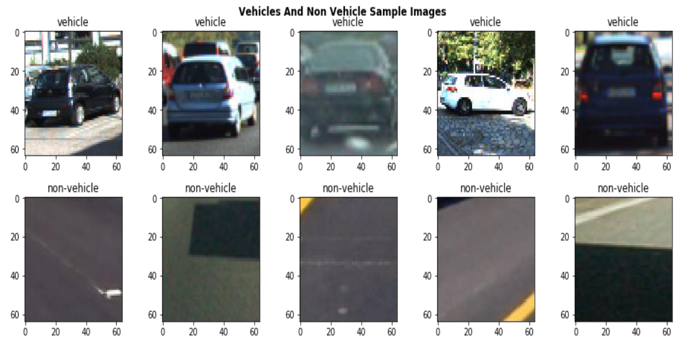
Keywords—*openCV, SSD, TensorFlow, ROI*

I. INTRODUCTION

In recent years there has been an increase in the number of motor vehicle resulting in problems like traffic congestion, accidents resulting in death and injuries. Developing countries bear a large amount of burden due to lack of proper infrastructure and traffic surveillance system. It has been noted that increase in preliminary transportation infrastructure system like more number of roads or flyovers can't solve these problems completely. An automated traffic surveillance system can reduce these problems by a large margin. The first step in traffic surveillance is recognizing vehicles in surveillance video cameras. Here, we are going to propose a system that is capable of detecting vehicles at a steady rate. Automatic detection of vehicles from videos is a challenging task. But then vehicle detection and counting is crucial to decrease traffic congestion and increase security. Preferably video should be recorded from CCTV cameras placed at the top of traffic signals. Data recorded from traffic flow can be recorded directly and can be communicated as needed. This paper specifically focuses on automatic car detection process. The process described here uses Haar like features previously introduced by Viola and Jones.

II. PROPOSED SYSTEM

We are using a machine learning based technique where a certain number of dataset is required for training and testing purpose. Algorithm from object detection API extracts features from these images. To understand how feature extraction is used, let's say each feature is a single value obtained by subtracting sum of pixels under white rectangles from sum of black rectangles. That way, there can be 20000 features resulting in huge calculation. But among these large numbers of features most are irrelevant. We need to select the features that classify the object to be detected i.e. vehicle or non-vehicle. The dataset comes from the GTI Vehicle Image Database, KITTI Vision Benchmark Suite, and examples extracted from the project video itself. You can see a sample of images from the dataset below:



We can clearly see both vehicle and non-vehicle images. Non-vehicle images tend to be other elements of the road such as the asphalt, road signs or pavement. The distinction is very clear. Most images also display the vehicle in the center, but in different orientations, which is good. Moreover, there is a good variety of car types and colors, as well as lighting conditions. Using which we can train the system to identify the vehicle and even count the number of vehicle in a video using the ROI technique.

For each feature, it finds the best threshold which will classify the objects based on the trained model. But obviously, there will be errors or misclassifications. This can be improved as we take large number of training datasets.

Here we propose a framework to combine several features into a cascade, i.e. a sequence of tests on the image or on particular regions of interest, organized into several stages. When the object passes through all stages of classifiers, it gets recognized.
To implement the framework, we have used OpenCV along with a python wrapper called cv2. Here we would describe the steps one by one.

1. Asset Preparation: Asset preparation means the training of image classifier to detect objects. Images can be occupied from car image datasets available online. Here, we have

collected images by manually cropping from video frames. Save them in jpeg or jpg format, preferably having less resolution.

2. Creating metadata: Metadata is a file containing number of objects, and dimensions of the objects within each image. This metadata invoked by openCV library would be used to train the classifier and can be accessed using Microsoft excel.

3. Training the classifier: We will start to train the system based on the data sets we have collected/created. Over a period of time the system will finish training the model that is required for detection and counting of the vehicles. The trained models will be imported as a frozen detection graph. Here each frames will be processed by SSD with Mobilenet model is developed on TensorFlow. This is a loop that continues working till reaching end of the video. By default we have used an SSD with Mobilenet model in this project. The paper about SSD: Single Shot MultiBox Detector (by C. Szegedy et al.) was released at the end of November 2016 and reached new records in terms of performance and precision for object detection tasks, scoring over 74% mean Average Precision at 59 frames per second on standard datasets such as PascalVOC and COCO.

4. Testing the classifier: Here OpenCV along with a python wrapper CV2 to invoke the classifier and detect objects from the video. Vehicles within the frame are highlighted with a rectangle. First we need to create a Video Capture object to access the video. The name of the video file must be passed as an argument. We would also need a frozen detection graph passed as an argument. **Now we can use a loop to read the video and detect cars using the detect MultiScale function.** Once the objects are detected we use cv2 rectangle function to draw a rectangle surrounding the detected vehicles. This concept is applied for the entire image and the final count of objects is present in variable count.

5. Counting the vehicle: We have used ROI technique to count number of vehicles in a video. Many common image operations are performed using Region of Interest in OpenCV. In computer vision, the **ROI** defines the borders **of** an object under consideration to perform some operation on it and also a ROI allows us to operate on a rectangular subset of the image. The typical series of steps to use ROI is: create a ROI on the image, perform the operation you want on this subregion of the image, reset back the ROI.

Here are some video snapshots showing how the cars get marked with a rectangle:

Video snapshot 1.

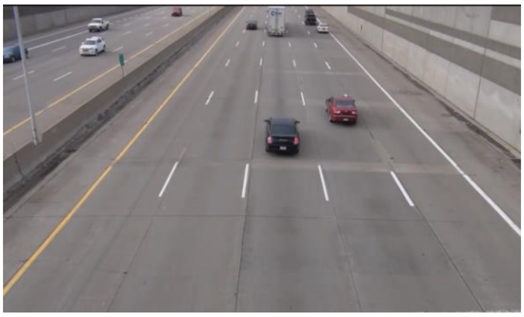


Figure 1: Image1

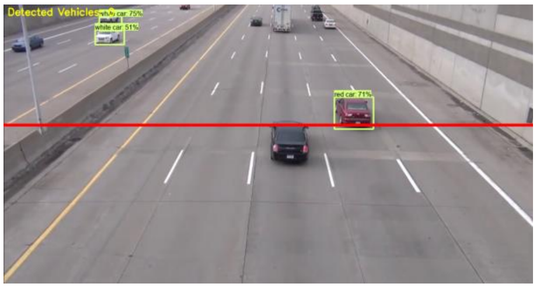


Figure 2: Image1 with cars in a rectangle

Video snapshot 2.

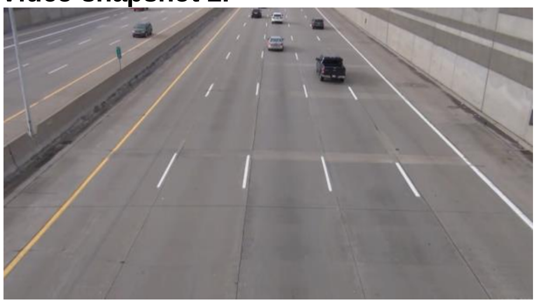


Figure 3: Image2

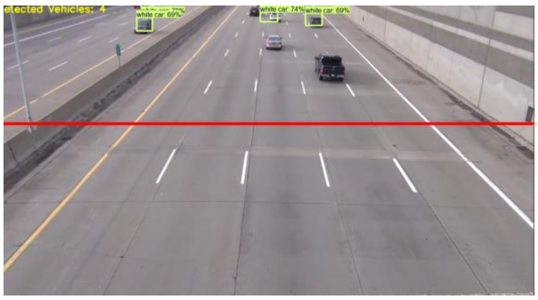


Figure 4: Image2 with cars in a rectangle and counting the vehicle

It is clearly observed in Fig. 2 and Fig. 4, some cars that are obscured because of distance and aren't detected either. This can be very easily improved by taking more data sets and this is a very useful technique for detecting vehicles from images or video frames.

REFERENCES

https://towardsdatascience.com/vehicle-detection-and-tracking-44b851d70508

These are the following citation:

1]Real-time Moving Vehicle Detection, Tracking, and Counting System Implemented with OpenCV Da Li1, 2 Bodong Liang1, 3 Weigang Zhang2 1. School of Automotive and Transportation Engineering, Shenzhen Polytechnic, Shenzhen 518055, China 2. School of Information Engineering, Chang'an University, Xi'an 710064, China 3. Corresponding author, Email: liangbodong@szpt.edu.cn.

2] Vehicle Detection Video Through Image Processing: The Autoscope System Panos G. Michalopoulos.

3] Moving Vehicle Detection for Measuring Traffic Count Using OpenCV Nilesh J. Uke Sinhgad College of Engineering, Department of Information Technology Pune, India Email: njuke.scoe@sinhgad.edu Ravindra C. Thool Department of Information Technology, S.G.G.S. College of Engineering and Technology, Nanded, India Email: rcthool@sggs.ac.in.

4] Automatic Vehicle Counting from Video for Traffic Flow Analysis Erhan Baş, A. Murat Tekalp, Fellow, IEEE, and F. Sibel Salman College of Engineering, Koç University, 34450 Sariyer, Istanbul, Turkey.

5] Video-based freeway-monitoring system using recursive vehicle tracking Brian Gloyer, Hamid K. Aghajan, Kai-Yeung Siu, Thomas Kailath.