Vehicle Detection Using OpenCV

BURRA MASTAN VEDASAI SAHITYA NITHIN (2019BEE04)
VEMPATI SRIDHAR (2019BEC30)

***Abstract*— Traffic has increased enormously with economic development. Due to the increasing urban population and hence the number of cars, the need for controlling the traffic in streets, highways and roads are vital. Vehicle detection has been a part of the traffic surveillance system for many years. Still, the vehicle detection method is challenging. In this project, a system that detects the vehicle in any video or image file is implemented. The process of detection of vehicles includes object detection by considering the vehicle as the primary object. Nowadays vehicle detection techniques are developing and existing techniques are improving to get better and accurate results. This methods are widely used in traffic monitoring, military applications. In this project the opensource library of python OpenCv and Haar cascade classifiers are used for the vehicle detection. In the vehicle detection the frames are extracted from the given video and vehicles are detected in the frames by extracting the Haar features from the frames based on these features the vehicles are detected.**

Keywords: Vehicle detection, OpenCV, Haar Cascade Classifier

I. Introduction

The idea of vehicle detection comes from the knowledge of object detection. Object detection is a fascinating field in computer vision. It goes to a whole new level when we're dealing with video data. We can perform super useful high-value tasks such as surveillance, traffic management, fighting crime, etc. using object detection algorithms. There are a number of sub-tasks we can perform in object detection such as counting number of objects, finding the relative distance between the objects, finding the relative size of the objects. All these sub-tasks are play a vital role in solving some of the real world's toughest problems.

Nowadays, video object detection is being deployed across a wide range of industries. The use cases range from video surveillance to sports broadcasting to interesting applications.

Every object class has its own special features that helps in classifying the class – for example all circles are round, Object class detection uses these special features. For example when, looking for circles, objects that are at a particular distance from a point (i.e. the center) are sought. Similarly, when looking for squares, objects that are perpendicular at corners and have equal side lengths are needed. A similar approach is used for vehicle detection where the vehicle is detected using some of its features which are extracted using Haar cascade classifiers.

The objective of the project is to detect object of interest (Vehicle) in video frames and to keep tracking the same object. In the project the video which is taken for detection is converted in different frames and from each frame the haar features are extracted and compared with all the frames in the video file and the vehicle pattern is recognized which is represented by a rectangular box indicating the vehicle in the frame.

This project has wide range of applicable areas such as traffic management, and rescue tasks, prevents traffic jams and congestions which in turn reduces and noise pollution. Surveillance on vehicles, etc.

II. Experimental Setup/Methodology

To get started with the project it is divided into four categories. They are

1. Collecting Positive and Negative Sample Images from dataset (Video)
2. Training Haar classifier using the above files.
3. Using the Classifier, Creating xml file for the required object
4. Detecting the appropriate object in the video using the xml file created previously.

Step1: For positive images dataset, some of the images were manually cropped out of the video and rest of them were downloaded through ImageNet and were made gray-scale as speed and performance of processing these images are high and simple when compared to its coloured image counter part. All the negative images are then placed in a folder called raw data within the positive folder of the training tool folder. Positive images are those that contains the target object and negative images are the ones which don't have the target objects.

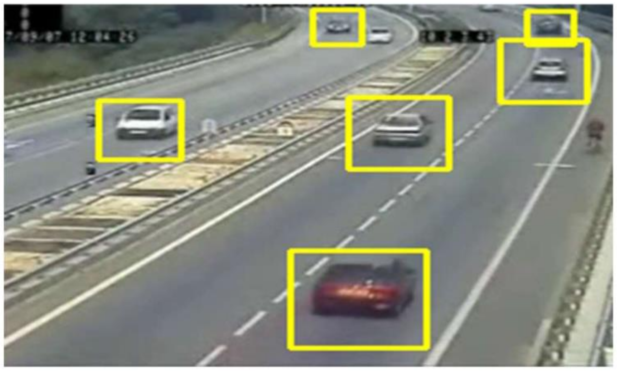
Step2: In this step, the positive images extracted from the video file are converted into gray scale because in gray scale it is easy to extract features from the images and most of the functions performs good for gray scale images rather than coloured images. These positive and negative images are fed into the classifier and features are extracted from the images.

Step 3: The next step involved detection of object which is accomplished by the detectMultiScale, the important parameter of this command is the scaling factor which plays an important role in avoiding false detection and miss detection. Various values of scaling factor were tried and the most suitable one was chosen for each car, bus, two-wheeler and pedestrain detection.

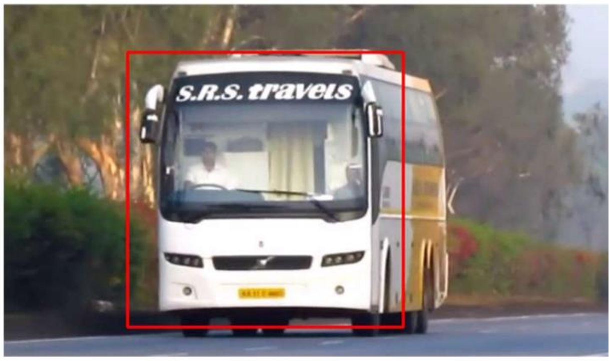
Step 4: Hence the moving vehicles are detected and highlighted using a box indicating it has been detected.

III. RESULTS AND DISCUSSION

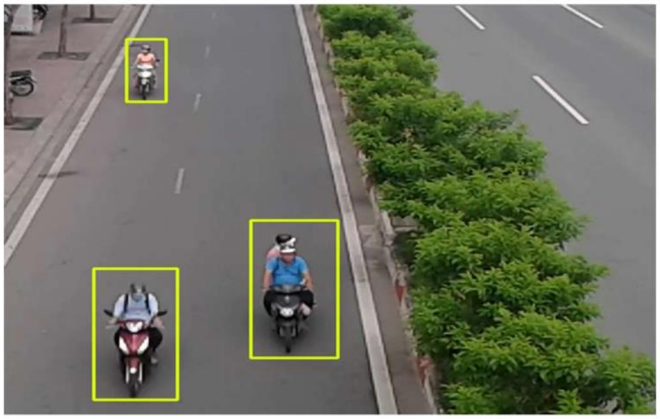
Sample images of the result obtained after using the various xml files used for detecting vehicles is shown below.



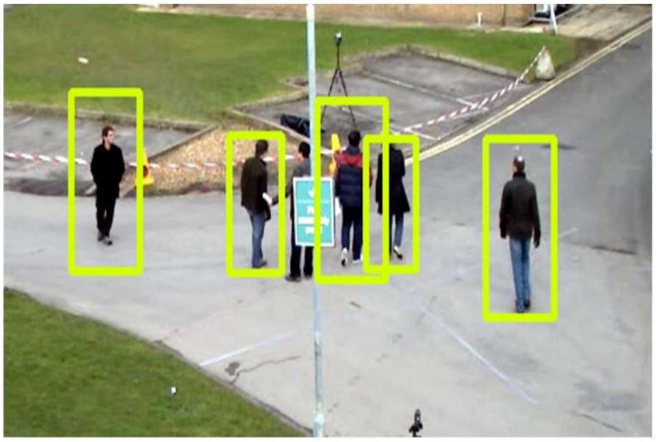
Detection of Car



Detection of Bus



Detection of Two –Wheeler



Detection of Pedestrian

In this way this method gave astonishing results which are quite good are useful. Coming to pros and cons of this project,

Pros:

- The vehicles can be detected in real time and can be applied to detect vehicles in different videos
- Easy to classify the vehicles with simple code and better understanding.
- Based on various scaling factors suitable factor can be taken to detect vehicles.

- Uses open source libraries and classifiers so it is accessible to everyone and can be easily implemented.

Cons:

- The vehicles is detected using more and more flickering which can be a problem while counting the number of vehicle
- The flickering should be smoothened with appropriate technique to get more and more better results.
- Presently the project cannot detect the number of vehicles but it can be further implemented to do this.

IV. Conclusion

Traffic is becoming an inevitable part of our daily lives especially in crowded cities which forces the demand for developing intelligent traffic surveillance systems. These system help in monitoring, analyzing, and controlling traffic, while operating in real time and providing accurate information about traffic conditions. This system is a simpler way of detecting vehicles which can be upgraded with neural networks and can reach to boundless applications. For more details go through my github. Thanks for your time.

V. Acknowledgment

We would like to give my special thanks to my teacher Mr. S.P. Paramesh. He encouraged us to do project based on machine learning by teaching the endless and fruitful applications of machine learning.
We would like to thank our peer Mr. Vikram. He has provided with a good encouragement and acted as a mentor for this project and provided consistent guidance for choosing this project.
We would like to thank our family members for their love and support.
Finally, we would like to thank you the readers for spending your valuable time for reading this report.

VI. References

[1] **Sandeer Soo,** Object detection using Haar-cascade Clasifier, Institute of Computer Science, University of Tartu

[1] **Science Direct,** Real – time detection of vehicles using the haar-like features and artificial neuron networks, The International Conference on Advanced Wireless, Information, and Communication Technologies (AWICT 2015)
https://www.sciencedirect.com/science/article/pii/S187705091503505X

[1] **Springer Link,** Real-Time Vehicle Detection and Tracking Using Haar-Like Features and Compressive Tracking

https://link.springer.com/chapter/10.1007/978-3-319-03413-3_27

[1] **cars.xml**
https://gist.githubusercontent.com/199995/37e1e0af2bf8965e8058a9dfa3285bc6/raw/e690cef3fb3ede5b869c3969cd6a9c5735d4ec7b/cars.xml

[1] **BURRA MASTAN VEDASAI SAHITYA NITHIN**
https://github.com/nithinsahitya/Vehicle_Detection_using_opencv.git