Vision Based Vehicle Detection Using Deep Learning

HOD:
Dr. Sandeep Shinde
Computer Department, VIT,
PUNE

Guide:
Prof. Ashwini Shingare
Computer Department, VIT,
PUNE

Anuj Pachauri
Computer Department, VIT, PUNE
Gr. No:1710856
anuj.pachauri17@vit.edu

Atharva Patrudkar
Computer Department,VIT,PUNE
GrNo:1710076
atharva.patrudkar17@vit.edu

Siddhant Saxena
Computer Department, VIT, PUNE
Gr. No:1710864
siddhanth.saxena17@vit.edu

Anant Thombare
Computer Department, VIT, PUNE
Gr. No:1710903
anant.thombare17@vit.edu

***Abstract*— Recently, there was a change observed in deep learning architectures for better application in vehicular traffic control systems. In TensorFlow, the pre-trained model is extremely efficient and may be transferred easily to unravel other similar problems. However, thanks to inconsistency between the first dataset utilized in the pre-trained model and therefore the target dataset for testing, this will cause low-accuracy detection and hinder vehicle counting performance. One major obstacle in retraining deep learning architectures is that the network requires an outsized corpus training dataset to secure good results. Therefore, we propose to perform data annotation and transfer learning from an existing model to construct a replacement model for vehicle detection and counting within the world urban traffic scenes. Then, the new model is compared with the experimental data to verify the validity of the new model. Besides, this paper reports some experimental results, comprising a group of innovative tests to spot the simplest detection algorithm and system performance. Furthermore, an easy vehicle tracking method is proposed to assist the vehicle counting process in challenging illumination and traffic conditions. The results showed a big improvement of the proposed system with the typical vehicle counting of 80.90%.**

I. INTRODUCTION

In the earlier days before the rise of machine learning, the process of vehicle counting was done manually. It was performed by a person standing by the roadside; using an electronic device to record the data using a tally sheet. In some cases, the person may do the counting by observing video footage captured by city cams or closed-circuit television (CCTV) placed above the road or highway. According to a study in [1], manual vehicle calculation performance is 99% accurate. This investigation is based on manual calculation of various vehicles from a 5 minutes video recording. Although the manual method provides high accuracy, it requires an extensive amount of human resources. Besides, it tends to be error-prone, especially on severe traffic flow and multiple road lines. Therefore, manual calculations are usually performed with only a small sample of data, and the results are extrapolated for the whole year or season for long-term forecasts. Vision-based vehicle detection through highly cluttered scenes is difficult. At present, this approach can be categorized into traditional and complex deep learning methods. Recently, deep learning networks (DLN) based on convolutional neural networks (CNN) have obtained state-of-the-art performance on many machine vision task. Therefore, researchers began to use it for vehicle detection and counting.

II. Literature Survey

With recent advancements in deep learning, computer vision applications such as object classification and detection can be developed and deployed more effectively. These applications have been proposed and shown significant performance improvements and enabling real-time processing of streaming data for analytic and making decision. A. Deep Neural Networks Deep architectures are useful in learning and have shown impressive performance for example in the classification of digits in the MNIST dataset [14]; CIFAR [15] and ImageNet [16] for object classifications. In this scheme, the lowest layer, i.e. feature detectors are used to detect simple patterns. After that, these patterns are fed into deeper, following, layers that form more complex representations of the input data. There are several approaches to learn deep architectures. One of the most frequently used in computer vision is convolution neural networks (CNNs), where the networks preserve the spatial structure of the problem by learning internal feature representations using small squares of input data. Features are learned and used across the whole image, allowing for the objects in the images to be shifted or translated in the scene and still detectable by the network. This is one of the reasons why deep architecture is so useful for object recognition such as in picking out digits, faces, objects and so on with different challenging conditions. Thus, to get a good classification result, the network is trained with a vast number of images such as using ImageNet [16] as the dataset to classify pictures. Besides the classification task, the deep architecture is widely used for object detection that draws a bounding box around each object of interest in the image and assigns them a class label. The bounding box indicates the position and scale of every instance of each object category. There are several approaches to object detection in computer vision such as Faster R-CNN, YOLO and SSD. Typically, deep convolutional neural network models may take days or even weeks to train on huge datasets for good performance. A way to reduce the training time is to reuse the model weights from pre-trained models that were trained using millions of natural images such as from ImageNet dataset. Such a methodology is called transfer learning. In this technique, the constructed models can be downloaded and used directly, whereby a neural network model is first trained on a problem similar to the one we have chosen. One or more layers from the pre-trained model are then used in a new model trained on the problem of interest. The pre-trained model has the advantage that it is already learned a rich set of image features. Besides, the model is transferable to the new task by fine-tuning the network. In this case, the model can be re-trained on a small number of images such that the network weights are small adjusted to support the new task. Thus, it has the benefit not only to decrease the training time for a neural network model but also can result in lower generalization error. For example, in [17] use ImageNet initialized models for object detection on the Pascal VOC dataset challenge, [18] use ImageNet initialized models for semantic segmentation. Other works that utilized the ImageNet dataset for training deep learning models for image classification such as in [19], [16].

In general, object detection is a task in computer vision that involves identifying the presence, location, and type of one or more intended objects in a given test image. It is a challenging problem that consists of three main processes, namely object recognition, localization and classification. In recent years, deep learning techniques have been applied to many vehicle detection problems and show promising results such as on standard benchmark datasets and in computer vision challenges. Several approaches are using deep learning techniques for object detection. Shaoqing Ren et al. [17] proposed a method, namely Faster R-CNN to improve both speeds of training and detection of the existing Fast R-CNN [20]. The method consists of two modules, namely, (a) region proposal network, where the convolutional neural network is used for proposing regions and the type of object to consider in the region and (b) Fast R-CNN for extracting features from the proposed regions and outputting the bounding box and class labels. Faster RCNN has proven to be efficient for object detection and secured the first-place on both the ILSVRC-2015 and MSCOCO-2015 object recognition and detection competition tasks. Joseph Redmon et al. [21] proposed an algorithm namely, you only look once (YOLO) for object detection. The algorithm is claimed to be much faster than the standard R-CNN [22] and achieving object detection in real-time. The authors then further improved the model performance and referred to as YOLO v2 [23] and YOLO v3 [24]. Another widely used model for object detection in the industry is the single-shot multi-box detector (SSD) [25]. It improves R-CNN [22] detection speed by eliminating the need of the region proposal network.

III. WORKING

OpenCV - OpenCV is a cross-platform library using which we can develop real-time computer vision applications. It mainly focuses on image processing, video capture and analysis including features like face detection and object detection.

TensorFlow Object Detection API- The TensorFlow object detection API is the framework for creating a deep learning network that solves object detection problems. This includes a collection of pretrained models trained on the COCO dataset, the KITTI dataset, and the Open Images Dataset. These models can be used for inference if we are interested in categories only in this dataset. However, to enable this framework we have used the COCO dataset thereby allowing us to predict the vehicle and other parameters.

Single Shot Multi-box Detector - Venerable VGG-16 architecture which is know for it's strong performance in high quality image classification tasks. so instead of the fully connected layers we've used the conv5 onwards layers thereby enabling to extract features at multiple scales. The layer which close to the image is know to have the higher resolution and as we keep on going ahead the size of the conv decreases along with the size of the resolution so as the conv keeps decreasing the bounding boxes are formed over the convolutions these are known as the Multi-boxes along with this the size of the conv decreases to 1x1 as we reach the last conv and is of the lowest resolution and hence gets further classified along with the features of our project.

COCO Dataset – COCO is a large-scale object detection, segmentation, and captioning dataset. We have used the v1_coco_2018 version data to process the features of the model and enable the extraction of the required parameters of the project.

Docker Setup with Nvidia GPU – Run the docker in the GPU without installing anything, just the nvidia-docker.

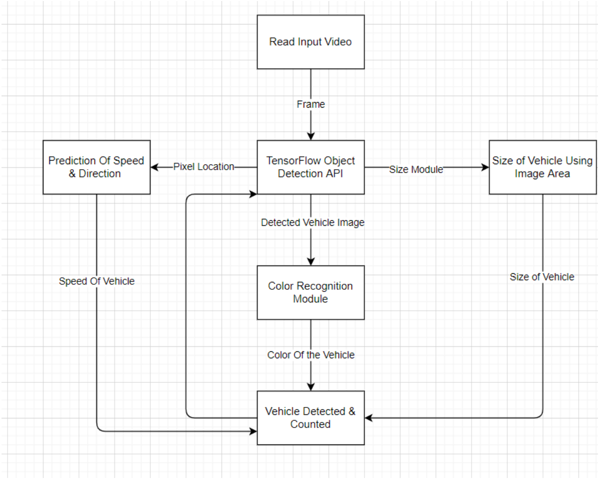

Fig.1- System Architecture

IV. APPLICATIONS

A method for detecting small objects in highway scenes is used to improve vehicle detection accuracy. The highway road surface area is extracted and divided into the remote area and the proximal area, which are placed into the convolution network for vehicle detection.

- A multi-object tracking and trajectory analysis method is proposed for highway scenes. The detection object feature points are extracted and matched by the ORB algorithm, and the road detection line.

V. ADVANTAGES

- To store the detected vehicles collectively in a proper path from video frames and save as a new image.
- The ability to store the output and the detected data using csv, after the end of the process for the source.

VI. CONCLUSION

Comparison of three models (Faster R-CNN ResNet101, SSD Inception, which were pre-trained on the COCO dataset showed that is achieving the best result. The results presented can be used as a reference for future development of a similar counting system. However, SSD performs worse in the morning and night condition of the custom dataset. Thus, the solution is to retrain the model with a custom dataset from the poor illumination condition environment using a data annotation tool and employs transfer learning with the weight training initialization method. The resulting model improves the counting accuracy very significantly. A tracking mechanism based on consecutive frames comparison was also proposed to aid the counting system. This mechanism may work only on vehicles moving in one direction without occlusion. In future studies, perhaps some uniformity can be done on the meta -architectures and detectors. Besides, the model used for retraining was a light-weighted SSD, which is called tiny-SSD. This is due to the limitation on the available hardware specification. To retrain SSD the recommended minimum GPU memory is 4GB, any specification below that is only suitable for training tinySSD. Thus, it is recommended that future studies need to consider the retraining of SSDinstead of tiny-SSD to compare the performances

VII. RESULT

The system is capable of Detection and classification of the vehicles, Recognition of approximate vehicle color, Detection of vehicle direction of travel, Prediction the speed of the vehicle, Prediction of approximate vehicle size, The images of detected vehicles are taken from video frame by frame and then they are saved as new images under "detected-vehicles" directory.

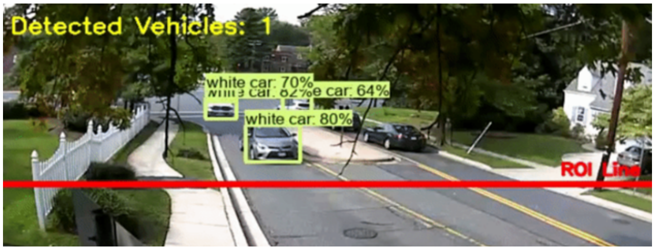

Fig.2- Vehicle Detection & Counting

VIII. Future Scope

There is good scope for improving the current solution for cloud storage. The project serves basic functionalities but can be extended to provide some advanced features. More Powerful Detection Models Currently this allows us to detect and process frames and obtain a speed of 59fps and mean average of over seventy. However, this rate could be significantly improvised by using more typical models containing higher convolutions. Data Analysis Using AWS Machine Learning algorithms can be applied to the present data and cloud services for storage to obtain insights. Users can get an overview of the most used vehicles, colors, speed. Ability to process different types of input videos Users might want to process videos for different types of road traffics such as two way lane road.

IX. ACKNOWLEDGEMENT

We would like to express our special thanks of gratitude to our teachers, as well as our college VIT Pune who provided us the opportunity to work on this project which also helps us doing a lot of research. We are really thankful to them. Secondly we would also like to thank our colleagues and friends who help us a lot finishing this project within the limited time frame.

X. References

[1] P. Zheng and M. Mike, “An investigation on the manual traffic count accuracy,” in 8th International Conference on Traffic and Transportation Studies (ICTTS 2012), 2012.

[2] Z. Zhang, K. Liu, F. Gao, X. Li, and G. Wang, “Vision-based vehicle detecting and counting for traffic flow analysis,” 2016 International Joint Conference on Neural Networks (IJCNN), pp. 2267–2273, 2016.

[3] M. S. Chauhan, A. Singh, M. Khemka, A. Prateek, and R. Sen, “Embedded cnn based vehicle classification and counting in non-laned road traffic,” in ICTD ’19, 2019.

[4] B. Dey and M. K. Kundu, “Turning video into traffic data - an application to urban intersection analysis using transfer learning,” IET Image Processing, vol. 13, pp. 673–679, 2019.

[5] H. Song, H. Liang, H. Li, Z. Dai, and X. Yun, “Vision-based vehicle detection and counting system using deep learning in highway scenes,” European Transport Research Review, vol. 11, pp. 1–16, 2019.

[6] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy, “Speed/accuracy trade-offs for modern convolutional object detectors,” 2016.

[7] A. A. Garc ´ ´ıa, J. A. Alvarez, and L. M. Soria-Morillo, “Evaluation of ´ deep neural networks for traffic sign detection systems,” Neurocomputing, vol. 316, pp. 332–344, 2018.

[8] N. Yadav and U. Binay, “Comparative study of object detection algorithms,” IRJET, vol. 11, 2017.

[9] A. Arinaldi, J. A. Pradana, and A. A. Gurusinga, “Detection and classification of vehicles for traffic video analytics,” in INNS Conference on Big Data, 2018.

[10] B. Dey and M. K. Kundu, “Turning video into traffic data – an application to urban intersection analysis using transfer learning,” IET Image Processing, vol. 13, pp. 673–679, 2019.