Second ASCE India Conference on "Challenges of Resilient and Sustainable Infrastructure Development in Emerging Economies" (CRSIDE2020)

March 2-4, 2020

Use of Computer Vision to Automatize Traffic Data Collection under Mixed Traffic Condition

Abhishek Kumar Prajapati[a*], Aman Gora[b], Amit Agarwal[c] *and* Indrajit Ghosh[d]

[a] Indian Institute of Technology (IIT) Roorkee, Roorkee-247667, India, aprajapati@ce.iitr.ac.in
[b] Indian Institute of Technology (IIT) Roorkee, Roorkee-247667, India, agora@ar.iitr.ac.in
[c] Indian Institute of Technology (IIT) Roorkee, Roorkee-247667, India, amitfce@iitr.ac.in
[d] Indian Institute of Technology (IIT) Roorkee, Roorkee-247667, India, indrafce@iitr.ac.in

***Abstract*— To tackle the planning related challenges for an urban agglomeration (e.g., congestion, air-pollution exposure, etc.), minimally, traffic flow data is required to build/calibrate/validate a model and/or for policy testing. Traditionally, manual traffic data collection is a tedious job, highly uncertain process, prone to errors in and expensive. Further, data like trajectories of moving vehicles cannot be extracted using the manual process. A one-stop solution to these problems is to use computer vision techniques in extracting traffic data from the video data. The objectives of the present study are to develop a deep-learning-based system for autonomous extraction of classified vehicular positions and trajectories in space and time from the input video data and to examine the performance of the model in mixed traffic stream.**

The system uses the YOLOv3 object detection model with the 'TensorFlow' API for detection and classification of the vehicles. A unique ID is assigned to every vehicle using Intersection over Union approach to detect the same in subsequent frames, to count the classified vehicles in various time-bins and to draw a unique trajectory for each vehicle. The model used for detecting vehicles is a unique model which is specially trained to suit India's traffic conditions. Vehicle detection classes include bus, car, truck, autorickshaw, and motorcycle. The system exhibits the Mean Absolute Percentage Error (MAPE) as 20.6% for counting of the classified vehicles.

***Keywords*— Heterogeneous Traffic, Classified Vehicle Count, Trajectory estimation, Vehicle Detection, Deep Learning**

I. Introduction

Urbanization has led to areas with dense urban activities which results to densely populated areas. High population density has led to a tremendous increase in the number of private vehicles, an increased number of goods vehicles, and massive pedestrian traffic. This has increased the likelihood of road-accidents, congestion and higher emissions. To solve such issues, city planners and researchers are trying to use various models to explore the traffic management strategies so that the negative effects can be diminished. The models can also help in making efficient decisions, forecast and assess the consequences.

Empirical traffic data are the basic input in any traffic management strategies and in building and analyzing traffic models. For autonomous collection of the data under homogeneous traffic conditions, several equipments are available. Among these, induction loops are widely used to serve both purposes of traffic management and traffic modeling purposes. In general, induction loops are deployed for each lane and may not be useful to collect data under mixed traffic conditions. Due to the limited success of such approaches and abundance of the video surveillance systems (also known as closed-circuit-television, CCTV) in most of the cities, a few image-processing-based data collection techniques are developed [1-3], which resonates induction loops in data collection. The video surveillance systems exhibit meteoric growth nowadays and usually include heterogeneous cameras with various resolutions, mounting points, and frame rates. A massive amount of data is generated through CCTV. This data can serve as a base for the automated traffic surveillance system.

In developing countries, like India, traffic includes different types of vehicles such as cars, buses, trucks, two-wheelers (motorized, non-motorized), three-wheelers (motorized, non-motorized), etc. Two-wheelers and three-wheelers are comparatively small in size and due to their higher maneuverability, lane discipline is absent [4]. Induction loops may not be useful to collect data under mixed traffic conditions. Alternatively, researchers have been using either manual data collection techniques or video-filming-based methods. These methods are useful in collecting aggregated data such as classified traffic counts in various time-bins but are not useful in collecting microscopic data (e.g., vehicle-trajectories). Vehicle detection and analysis, through image processing and computer vision, owing to its nonintrusive nature and resourcefulness in computing nontrivial data has been a special area of interest. There are wide varieties of algorithms used in this area and a detailed review is presented in the following section. Using this, the Spatio-temporal attributes of the vehicles

* Corresponding author

can be obtained over a certain length of the road, which is useful in obtaining vehicular trajectory data. A lot of useful information can be extracted from the trajectory data, such as driver behavior.

II. Literature Review

Mainly, collecting traffic data over a stretch of road length is useful under mixed traffic conditions when compared to the data collected at a point on the road section [5-7]. Mallikarjuna et. al. [1] develop an offline image-processing-based system to obtain data from a video which uses the background-subtraction approach. The classification mechanism enables vehicular classification into four different categories, namely light motor vehicles (LMVs), motorized (TWs), heavy motor vehicles (HMVs), and motorized three-wheeler autos. The system is also capable of calculating individual vehicle characteristics such as trajectory. However, in this background-subtraction approach, vehicles are classified using the size which has a limited success rate in dense traffic conditions. Also, this cannot be applied to a video from a moving camera.

There is a very limited number of researches in the area of analyzing data from video surveillance systems using deep learning. An overview of the recent advances in the topics related to the present work i.e. objects detection followed by multi-object tracking (MOT) is provided in the next two sections.

A. Object Detection

Currently, most of the object detectors are based on convolutional neural networks (CNN) and can be broadly divided into two categories. The first one is single-stage detectors and the second is two-stage detectors. Single-stage detectors, in general, are fast and predict objects bounding boxes together with classes in a single network pass. Examples of the single-stage detectors are You Only Look Once (YOLO) [8] and Single-Shot-Detector (SSD) [9]. These architectures work well in cases where target objects occupy a significant amount of the image. An example of such data is the UA-DETRAC vehicle detection dataset [10]. Based on this data, Dmitriy Anisimov and Tatiana Khanova [11] showed that a thoroughly constructed SSD-like detector can run faster than 40 frames per second on a modern CPU while maintaining good precision. Further, an improved example of the good speed-precision trade-off is YOLO v2 architecture [12], which was specialized for vehicle detection using additional loss normalization, multi-layer feature fusion strategy, and anchor clustering. The most significant example of two-stage detectors is the R-CNN family of detectors [13–15, 16] that currently occupy leading places in the COCO [17] and Cityscapes [18] datasets. In comparison to the single-stage detectors, two-stage detectors first predict regions and then refine and classify each of them during the second stage. Early R-CNN [14] work adopted a straightforward approach: regions are generated by a selective search algorithm and then fed to the classification CNN. The overall speed of the R-CNN is low because of the selective search compute time and requirement to run heavy classifier per each region. To overcome this limitation, Fast R-CNN was proposed [13]. Instead of running CNN for each region, Ross Girshick fed the whole image to the CNN and pooled Regions of Interest (RoI) from the last feature map. Replacing selective search in the Faster R-CNN [16] with tiny CNN, called region proposal network (RPN), further boosted the precision and speed of the detector. A thorough comparison between single and two-stage detectors is presented in the work, speed-accuracy tradeoff being central to it [19]. Many vehicle-detection works adopted variants of the Faster R-CNN architecture. Wang et al. [20] studied the application of the focal loss [21] for vehicle surveillance. Li [22] proposed to better handle blurring and short-term occlusions by processing multiple adjacent frames. They showed that being a relatively simple technique, focal loss provides significant performance improvements. Hu et al. [23] focused on improving the scale robustness of Faster R-CNN and suggested context-aware RoI (CARoI) pooling that uses deconvolution with bi-linear kernels to accurately represent the features for small objects. CARoI pooling works on top of the multiple layers and also fuses high- and low-level semantic information for enhanced performance. However, the improvements in terms of the speed-accuracy trade-off in the faster R-CNN are lesser than the YOLOv3 model. Therefore, as a starting point, the present study uses YOLOv3 for object detection.

B. Multi-object tracking (MOT)

The progress in the precision of the detectors mentioned above made tracking-by detection approach a leader in the multi-object tracking (MOT) task. In this approach, tracking is viewed as a data association problem where the goal is to combine detections of unique vehicles across multiple frames into unique tracklets. Classical methods that use tracking-by-detection rely only on motion clues coming from the detector and deal with the Data Association problem using optimization techniques. Some well-known examples include Multiple Hypothesis Tracking (MHT) [24] and Joint Probabilistic Data Association Filter (JPDAF) [25]. These methods undertake the association problem on a frame-by-frame basis. However, their combinatorial complexity, which is exponential in the number of tracked objects, makes them unsuitable for real-time tracking. On the contrary, a recent SORT tracker [26] showed that combination of a simple Hungarian algorithm and Kalman filtering technique for movement forecasting could achieve real-time processing speed while maintaining favorable performance. Most of the recent improvements in the MOT task involve fusing motion features with appearance one to distinguish highly occluded objects better and reidentify lost instances. The appearance clues usually come from convolutional neural networks [27, 28]. However, Tang et al. [29] showed that hand-crafted features, like the histogram of oriented gradients and color histograms, might also be used if no training data is provided. From the practical point of view, computing visual features for each tracked object leads to a highly increased computational burden, especially if the number

of objects is high. Thus, such approaches have limited application for mixed traffic conditions in dense urban agglomeration which is the main focus of the present study. Together with detector processing time, cumulative performance usually is far from being near real-time. It is non-trivial to state that concurrent vehicle detection and tracking is a domain of active research [30, 31]. The coupling of these tasks might solve the performance problem mentioned above. Detectors already incorporate appearance features; detection precision will also benefit from the temporal image context. In the present study, to achieve higher accuracy, YOLOv3 is used and for robust and easy tracking, the Intersection Over Union algorithm is used.

III. Methodology

This paper aims to develop an autonomous approach which is capable of estimating traffic flow and trajectory, i.e. for counting and classifying vehicles by their movement directions and its trajectory calculation. To achieve that goal, the problem was divided into four sub-tasks: vehicle detection, vehicle tracking, vehicle counting, and trajectory estimation. The proposed approach leads to a modular and easy to test model composed of the detection and tracking modules. The process is demonstrated in Fig. 1. The following sections exhibit each module and their data requirements. The presented approach is made open-source and hosted at https://github.com/teg-iitr/Computer-Vision-for-traffic-data-collection.

A. Data Set

For the proposed approach, two different datasets are used; they are The Indian Driving Dataset (IDD, see section A.1) [32] and the Pre-trained COCO weights (see Sec. A.2). The former is used for the purpose of training the YOLOv3 [33] object detection model and, the later is taken as the initial training weights and as an input to the model. New vehicle-classes like autorickshaw, two-wheeler, and truck are included. Also, these vehicle classes contribute to a wider portion of the Indian traffic.

A.1 Indian Driving Dataset (IDD)

The IDD consists of 6,993 labeled images for training, obtained from a front-facing camera attached to a car. It has fifteen vehicle classes in total which include vehicular classes like bicycles, motorbike, car, bus, truck, and autorickshaw. These are the classes used in the current proposed system.

A.2 Common Object in Context (COCO) Dataset

COCO [11] is a large-scale object detection, segmentation, and captioning dataset. For object detection, it has 80 object classes, including vehicular classes like bicycle, car, motorbike, bus, truck, etc. The authors used this dataset to obtain the YOLOv3 weights. These pre-trained weights are used in the proposed approach.

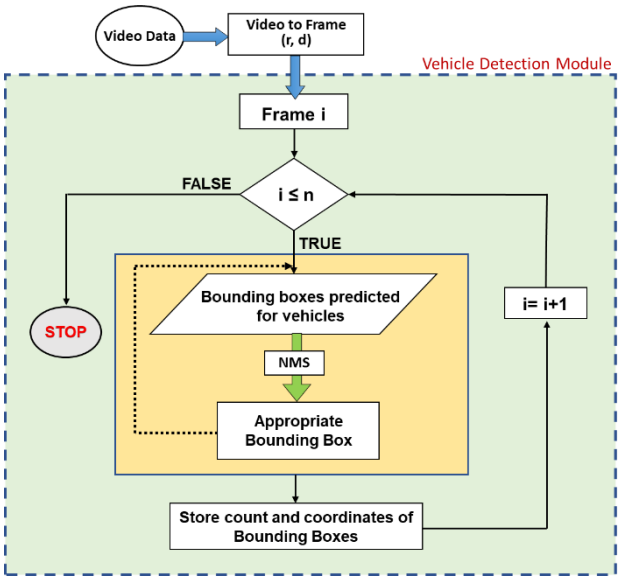

Figure 2: Flow-chart for the detection module.

B. Vehicle Detection Module

For the purpose of vehicle detection in each frame of video data, the YOLOv3 object detection model is used which is the state-of-the-art deep learning-based model for object recognition related tasks.

B.1 YOLOv3 Object Detection Model

YOLOv3 is an improved version over YOLOv2. It includes a much deeper network "Darknet-53", i.e. 53 convolutional layers whereas YOLOv2 uses Darknet-19 [12, 33]. It has a better feature extractor and a better object detector with feature map up-sampling and concatenation. The objectness score is predicted using logistic regression. Unlike in YOLOv2, instead of "Softmax" for class prediction, independent logistic classifiers and binary cross-entropy loss is used. Just like in YOLOv2 it uses Batch Normalization. Its feature map is taken from two previous layers and is up-sampled by 2 times. Another feature map is also taken from an earlier network layer and is merged with the up-sampled features using concatenation. This is a typical encoder-decoder architecture that resonates with the evolution of the Single Shot Detector (SSD) to De-convolutional Single Shot Detector (DSSD). In this, the sum of squared error loss is used during the training. Finally, "k-means clustering" is used here to find a better bounding box prior.

B.2 YOLOv3 for vehicle detection

The YOLOv3 object detection model is trained exclusively for the detection of various vehicle classes. At first, the video is converted to frames at rate 'r' and duration 'd'. The outer loop in Vehicle Detection Module (VDM) runs across the frames while the inner loop runs within a frame to detect individual vehicles. VDM consists of the YOLOv3 i.e. deep learning-based object detection model. 'TensorFlow' object detection API is used for detecting vehicles in a frame. The VDM takes raw video frames as input, processes it through the deep convolutional neural network and, yields the bounding box

coordinates and the class of the detected vehicle. In order to avoid multiple bounding boxes for a detected vehicle/pedestrian, the Non-Maximum Suppression (NMS) method is used. NMS is a post-processing step that transforms many imprecise bounding boxes, ideally, in a single bounding-box for each detected vehicle. The flow-chart shown in Fig 2 represents the detection module. Fig. 3 compares the raw frame

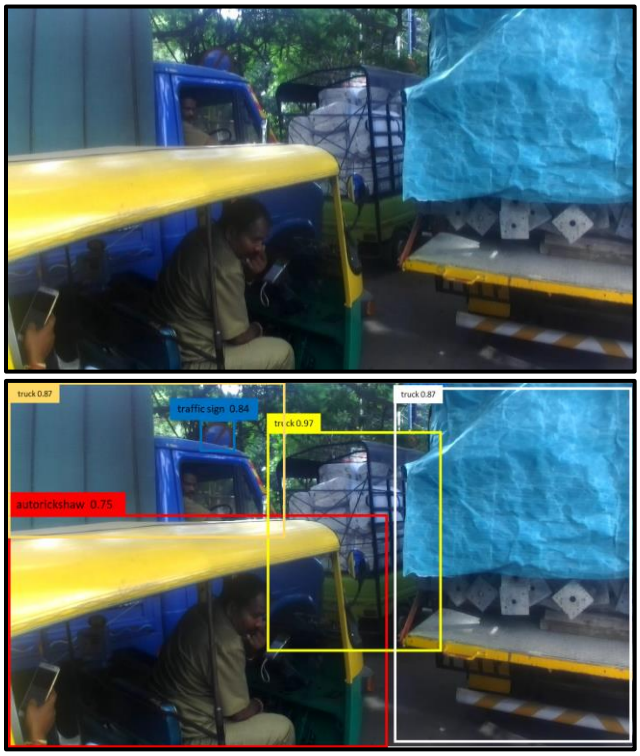

Figure 3: Raw video frame and processed frame.

and the preocessed frame showing the detected vehicles along with there respective class.

C. Vehicle Tracking Module

Object tracking is the process of taking initial set of object detections (bounding box coordinates), assigning a unique ID to each of the initial detected vehicle and then tracking each of the vehicle as they move around frames in the video, while maintaining the assignment of unique IDs. The assignment of unique ID is important to extract the vehicle-trajectory. For this purpose, the "mean Intersection Over Union (IOU) and Linear Sum Assignment" [34] tracking methods are used, which does the task robustly with minimal error associated. The concept of IOU is that if a bounding box in the current frame overlaps the one in the previous frame, it's probably the same object or vehicle (in present case). For this purpose a "Shape score" is calculated for every individual bounding box in the current frame. Considering bounding boxes in the previous frame, the bounding box which has highest shape score, is assigned the same vehicle ID. If the shape and size dose not vary much, the score is higher. This is demonstrated in Fig. 4 with sample scores**.** The red bounding box belongs to the current frame and the green bounding box belongs to the previous frame.

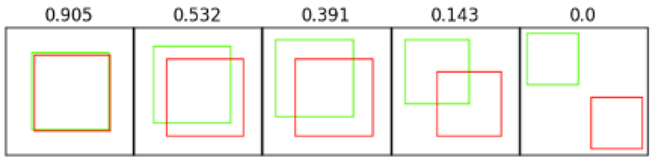

Figure 4: Sample IOU scores (Shape scores)

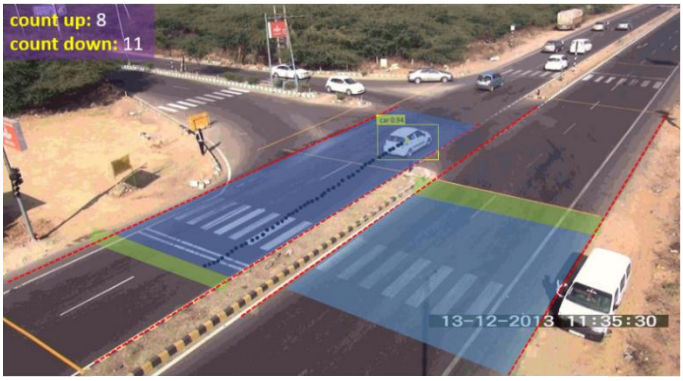

Figure 5: Counting Zone and trajectory plotting.

D. Vehicle Counting Module

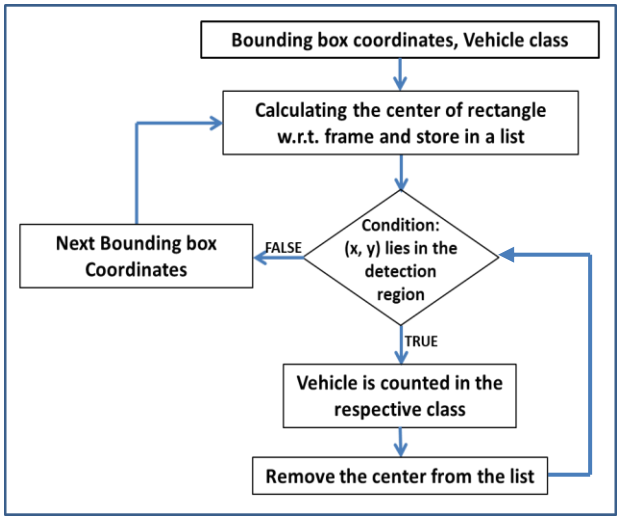

Figure 6: Counting Algorithm.

The counting algorithm works across the frames. The inputs to the algorithm are the bounding box coordinates and classes of the detected vehicle. A region (green region) is defined using two lines in the frame (Fig. 5) called a counting zone.

The algorithm for counting is presented in Fig. 6. First, the centroid for each bounding box is calculated and are stored in a list. In the counting step, the counter will count just the vehicles which are passing in a specific direction. The loop in Fig. 6 is iterated for every centroid in the list. Additionally, if a vehicle stops, turns or moves in the wrong direction in the detection zone, it will not be counted. In this technique, counting is according to the number of moving vehicles detected in the detection zone (between red dotted lines for each direction, i.e. boundary conditions) and classified in one of the mentioned classes. After the vehicle is counted, it is removed from the list of detected vehicles in order to avoid re-counting. The

aggregated count is shown at the top left corner of the frame and the class and time of passing are saved in a text file.

E. Vehicle Trajectory Module

After the vehicle is detected and tracked across the frame, its trajectory is plotted, using the saved centroid coordinates. To avoid chaos trajectory is plotted using simple dots/circles, only in the translucent overlay frame (Blue region) as shown in Fig. 5. With the help of allotted unique ID of individual vehicles, unique trajectories are plotted. A text file containing detected vehicle class, its unique ID and coordinates of trajectory, with respect to the frame is also stored. This data is then plotted on the road network.

IV. Corroboration of Results

A. Case study

In order to test the functionality of the proposed system, a case study is conducted. For the purpose, a CCTV footage of T-intersection in the Delhi NCR (National Capital Region), having coordinates (28º26'56.5''N 77º07'20.0''E) is used (see Fig. 5). The study area has two-lane roads. CCTV footage (video data) of the location is fed to the system to extract traffic parameters. In counting, the classified vehicle count is obtained and then compared to the manually obtained count. The comparison is shown in *Table 1*.

B. Results

For detection, average precision (AP), a popular metric in measuring the accuracy of object detectors like Faster R-CNN, SSD, YOLOv3, etc. is used. The average precision (AP) values for each detected vehicular class is shown in Fig. 7. The mean average precision (mAP) of the system is 0.70.

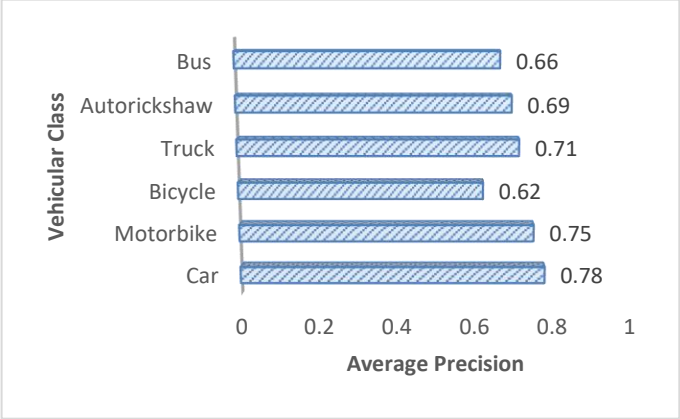

Figure 7: Average precision values for vehicular classes

Further, the Mean Absolute Percentage Error (MAPE) is used for verifying the accuracies in the vehicular counts. This is calculated using the following formula.

$$MAPE = \frac{1}{n} \times \sum_{i=1}^{n} \frac{|Predicted\ value - Observed\ value|}{Observed\ value} \times 100$$

Error, less than 10% is considered highly accurate, between 11-20% is good, 21-50% is acceptable and more than 50% is inaccurate [35]. MAPE for both side of the road is calculated manually using the observed count and the predicted count by the system (Table. 1).

Table 1: MAPE Calculation

Vehicle Class	Left		Right	
	Observed	Predicted	Observed	Predicted
Bicycle	3	2	0	0
Motorbike	127	121	47	40
Auto-rickshaw	3	2	5	3
Car	348	342	238	236
Bus	2	2	1	1
Truck	7	9	12	9
MAPE	20.6 %			

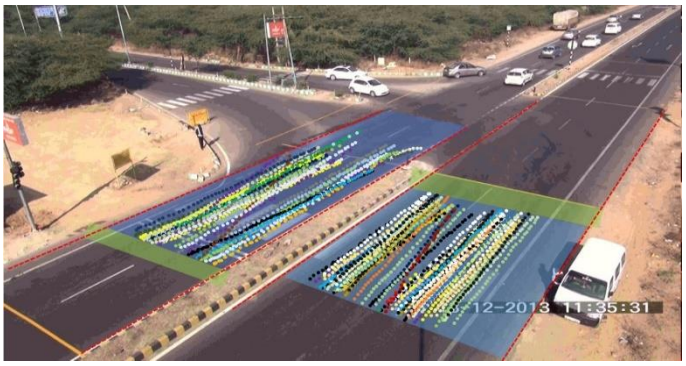

Figure 8: Agrregated Trajectories for Cars

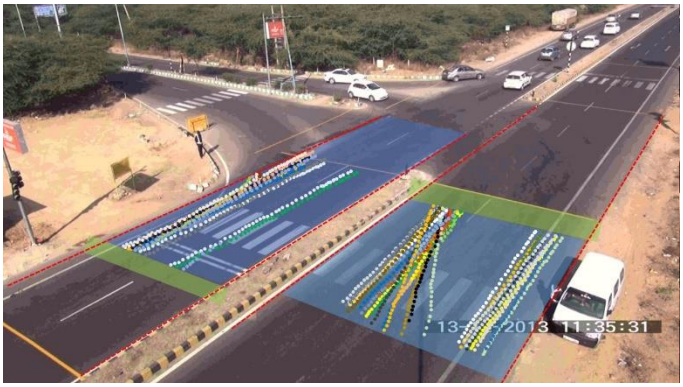

Figure 9: Agrregated Trajectories for Motorbike

Fig. 8 and Fig. 9 are showing the trajectories of car and motorbike respectively in the detection zone. Different colors in the figures are represeting trajectories of the vehicles. Interestingly, a few vehicles turning left or taking 'U' turn are also identified in it. Further, from the trajectories of the vehicles, it can be observed that ratio of lateral to longitudnal displacement of the motorbike is higher than car which is a typical behavior of the smaller vehicles with higher

maneuverability. Thus, the trajectories also validate the detection of the classified vehicles.

V. Conclusion

This study presents a traffic data extraction system, based on computer vision and image processing algorithms. In the proposed system, the vehicles are identified according to predefined vehicle classes, counted in the counting zone and finally, the trajectories are plotted for the detection zone. The system is able to classify the vehicles using a combination of image processing methods and processing through a Deep Convolutional Neural Network (DCNN). The results indicate that the presented method works effectively. The classified detection test error is about 30 percent. The counting error is about 20.6 percentage which makes it suitable for vehicle detection and traffic analysis purposes. Further, the system also provides the trajectories of the classified vehicles. The system holds the potential to (a) to reduce the costs (time and monetary) of carrying out a traffic survey with lesser chances of errors and good accuracy (b) obtain both the microscopic and macroscopic traffic parameters for a huge dataset quickly.

References

[1] Mallikarjuna, C & Phanindra, A. & Rao, K. R. Traffic Data Collection under Mixed Traffic Conditions Using Video Image Processing. Journal of Transportation Engineering-asce - J TRANSP ENG-ASCE. 135. 10.1061/(ASCE)0733-947X (2009)135:4(174), 2009.

[2] Zhang, G., Avery, R. P., & Wang, Y. (2007). Video-Based Vehicle Detection and Classification System for Real-Time Traffic Data Collection Using Uncalibrated Video Cameras. Transportation Research Record, 1993(1), 138–147. https://doi.org/10.3141/1993-19

[3] Fathy, M., and Siyal, M., Y., "A window-based image processing technique for quantitative and qualitative analysis of road traffic parameters," in IEEE Transactions on Vehicular Technology, vol. 47, no. 4, pp. 1342-1349, Nov. 1998. doi: 10.1109/25.728525

[4] Agarwal, A. and Lämmel, G. Modeling seepage behavior of smaller vehicles in mixed traffic conditions using an agent-based simulation. In: Transp. in Dev. Econ. 2016. 2:8, pp. 1–12. DOI: 10.1007/s40890-016-0014-9.

[5] Singh, B. “Simulation and animation of heterogeneous traffic on urban roads.” Ph.D. thesis, Indian Institute of Technology, Kanpur, India, 1999.

[6] Nagaraj, B. N., George, K. J., and John, P. K. “A study of linear and lateral placement of vehicles in mixed traffic environment through video-recording.” J. Indian Road Congress, 42, 105–136, 1990.

[7] Arasan, V. T., and Koshy, R. Z. “Methodology for modeling highly heterogeneous traffic flow.” J. Transp. Eng., 1317, 544–551, 2005.

[8] Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: unified, real-time object detection. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR). p. 779–88, 2016.

[9] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC. SSD: single shot multibox detector. In: Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics) 9905 LNCS. 2016, p. 21–37. https://doi.org/10.1007/978-3-319-46448-0_2

[10] Wen L, Du D, Cai Z, Lei Z, Chang MC, Qi H, Lim J, Yang MH, Lyu S. UA-DETRAC: A new benchmark and protocol for multi-object detection and tracking. 2015. arXiv:1511.04136

[11] Anisimov D, Khanova T. Towards lightweight convolutional neural networks for object detection. In: 2017 14th IEEE international conference on advanced video and signal-based surveillance (AVSS). 2017, p. 1–8. https://doi.org/10.1109/AVSS.2017.8078500 , ,

[12] Sang J, Wu Z, Guo P, Hu H, Xiang H, Zhang Q, Cai B. An improved YOLOv2 for vehicle detection. Sensors.2018;18(12):4272. https://doi.org/10.3390/s18124272

[13] Girshick R. Fast R-CNN. In: Proceedings of the IEEE international conference on computer vision 2015 Inter, 2015, p.1440–8. https://doi.org/10.1109/ICCV.2015.169

[14] Girshick R, Donahue J, Darrell T, Malik J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. 2012, p. 580–7, http://arxiv.org/pdf/1311.2524v3.pdf

[15] He K, Gkioxari G, Dollar P, Girshick R. Mask R-CNN. In: 2017 IEEE international conference on computer vision (ICCV). vol. 2017 Oct, 2017, p. 2980–8. IEEE. https://doi.org/10.1109/ICCV.2017.322

[16] Ren S, He K, Girshick R, Sun J. Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell. 2017;39(6):1137–49. https://doi.org/10.1109/TPAMI.2016.2577031

[17] Lin TY, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick CL. Microsoft COCO: common objects in context. In: Lecture notes in computer science (including subseries lecture notes in artificial intelligence and lecture notes in bioinformatics). vol. 8693 LNCS, 2014, p. 740–55. https://doi.org/10.1007/978-3-319-10602-1_48

[18] Cordts M, Omran M, Ramos S, Rehfeld T, Enzweiler M, Benenson R, Franke U, Roth S, Schiele B. The cityscapes dataset for semantic urban scene understanding. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR). 2016, p. 3213–23. IEEE. https://doi.org/10.1109/CVPR.2016.350

[19] Huang J, Rathod V, Sun C, Zhu M, Korattikara A, Fathi A, Fischer I, Wojna Z, Song Y, Guadarrama S, Murphy K. Speed/accuracy trade-offs for modern convolutional object detectors. In: 2017 IEEE conference on computer vision and pattern recognition (CVPR). 2017, p. 3296–7. IEEE. https://doi.org/10.1109/CVPR.2017.351, arXiv:1611.10012, http://ieeexplore.ieee.org/document/8099834/

[20] Wang X, Cheng P, Liu X, Uzochukwu B. Focal loss dense detector for vehicle surveillance. In: 2018 international conference on intelligent systems and computer vision (ISCV). vol. 2018 May, 2018, p. 1–5. IEEE. https://doi.org/10.1109/ISACV.2018.8354064.

[21] Lin TY, Goyal P, Girshick R, He K, Dollar P. Focal loss for dense object detection. In: Proceedings of the IEEE international conference on computer vision. 2017 Oct, p. 2999–3007. https://doi.org/10.1109/ICCV.2017.324

[22] Li S. 3D-DETNet: a single stage video-based vehicle detector. 2018, arXiv:1801.01769

[23] Hu X, Xu X, Xiao Y, Chen H, He S, Qin J, Heng PA. SINet: A scale-insensitive convolutional neural network for fast vehicle detection. In: IEEE transactions on intelligent transportation systems. vol. 20, no. 3. 2019, p. 1010–9. https://doi.org/10.1109/TITS.2018.2838132.

[24] Kim C, Li F, Ciptadi A, Rehg JM. Multiple hypothesis tracking revisited. In: 2015 IEEE international conference on computer vision (ICCV). vol. 22. 2015, p. 4696–704. IEEE. https://doi.org/10.1109/ICCV.2015.533.

[25] Rezatofghi SH, Milan A, Zhang Z, Shi Q, Dick A, Reid I. Joint probabilistic data association revisited. In: 2015 IEEE international conference on computer vision (ICCV). 2015, p. 3047–55. No. December, IEEE. https://doi.org/10.1109/ICCV.2015.349.

[26] Bewley A, Ge Z, Ott L, Ramos F, Upcroft B. Simple online and real time tracking. In: 2016 IEEE international conference on image processing (ICIP). vol. 2016-Aug, p. 3464–8. IEEE. 2016. https://doi.org/10.1109/ICIP.2016.7533003,

[27] Chu P, Ling H. FAMNet: Joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking. 2019. arXiv:1811.07258arXiv:1904.04989

[28] Wojke N, Bewley A, Paulus D. Simple online and real time tracking with a deep association metric. In: 2017 IEEE international conference on image processing (ICIP). vol. 2017 Sept, 2017, p. 3645–9. IEEE. https://doi.org/10.1109/ICIP.2017.8296962

[29] Tang Z, Wang G, Xiao H, Zheng A, Hwang JN. Single-camera and inter-camera vehicle tracking and 3D speed estimation based on fusion of visual and semantic features. In: 2018 IEEE/CVF conference on computer vision and pattern recognition workshops (CVPRW). vol. 2018 June, 2018, p. 1080–7. IEEE. https://doi.org/10.1109/CVPRW.2018.00022 ,

[30] Li C, Dobler G, Feng X, Wang Y. TrackNet: simultaneous object detection and tracking and its application in traffic video analysis. 2019, p. 1–10, arXiv:1902.01466

[31] Luo W, Yang B, Urtasun R. Fast and furious: real time end-to-end 3D detection, tracking and motion forecasting with a single convolutional net. In: 2018 IEEE/CVF conference on computer vision and pattern recognition. 2018, p. 3569–77. IEEE. https://doi.org/10.1109/CVPR.2018.00376 ,

[32] Varma, G, Subramanian, A, Namboodiri, A, Chandraker, M, Jawahar, C, V. IDD:A Dataset for Exploring Problems of Autonomous Navigation in Unconstrained Environments - IEEE Winter Conf. on Applications of Computer Vision (WACV 2019)

[33] Redmon, J, Farhadi, A. YOLOv3: An Incremental Increase. 2018. https://arxiv.org/abs/1804.02767v1

[34] E. Bochinski, V. Eiselein and T. Sikora, "High-Speed tracking-by-detection without using image information," 2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Lecce, 2017, pp. 1-6. doi:10.1109/AVSS.2017.8078516

[35] Kumar, S.V.: Traffic flow prediction using kalman filtering technique. In: Procedia Engineering, vol. 187, pp. 582-587. Vilnius, Lithuania (2017). doi: 10.1016/j.proeng.2017.04.417