Fourth International Conference on Electronics, Communication and Aerospace Technology (ICECA-2020)
IEEE Xplore Part Number: CFP20J88-ART; ISBN: 978-1-7281-6387-1

Vehicle Density Analysis and Classification using YOLOv3 for Smart Cities

Rashmi C R[1]
Asst. professor, Dept of CSE,
CIT, Gubbi, Tumkur, Karnataka, India
Email: rashmicr46@gmail.com

Dr. Shantala C P[2]
Professor & Head, Dept of CSE,
CIT, Gubbi, Tumkur, Karnataka, India
Email: shan1675@gmail.com

***Abstract*—Incorporation of the digital technologies in the surveillance of urban mobility such as monitoring the traffic density will help in improving the quantity of vehicles/arrangements to be provided for the public commutation, facility to be incorporated in reducing the traffic, infrastructure to be provided such as road widening, pedestrian path, over bridge, under pass etc., where traffic and transport is an issue. This can be implemented in the city, at which it is recognized to be developed as a smart city. The proposed research work analyzes the vehicle density using python-OpenCV and YOLOv3. Real time videos are recorded in four directions from Sony HD IP cameras in a designated area. Image frames from video sequence are used to detect moving vehicles. The background extraction method is applied for each frame which is used in subsequent analysis to detect and count all the vehicles. The blobs are detected for each vehicle which helps to track the vehicle in motion. The center of each vehicle with blob gives the count of vehicle based on the lanes considered. This work not only counts the vehicle in real time but also classifies the different vehicles using deep learning technique. YoloV3(You only look once) object detection system is used along with a pre-trained model called darknet to classify the vehicle into different categories (bus, car, motorcycle etc). This deep learning method showed better classification and detection rate compared to blobs and morphological method used for counting the vehicles. Classification is shown for vehicle and also person classification is considered to analyze the percentage of people and vehicles. The analysis of percentage of vehicles is shown using pie chart.**

Keywords: Vehicle density, darknet, vehicle classification, opencv

I. INTRODUCTION

In today's scenario of fast growing traffic condition, maintaining the proper transportation system is a very difficult job. It is due to the exponential growth of vehicles each day. Hence there is a necessity of automated road transportation mechanism without much human intervention. One of the basic mechanisms is to automate the vehicle counting system which offers intelligent transport surveillance system. It helps in finding the current state of traffic and also for managing it. This automation helps in monitoring and estimating the real time traffic flow in any location. Hence, it is one of the important techniques for optimizing the traffic signals. There are so many methods for estimating the traffic density, some of them are infrared or inductive loop detectors, radar and traffic cameras. Among these methods, computer vision based method could be appropriate since other methods may have less performance and high maintenance. But computer vision based methods also have drawbacks because of weather and lighting situations, HD video processing overhead etc., [1]. Intelligent transportation system is one of the key factors for the development of smart cities. This can be achieved by analyzing the real time traffic density through video processing. This paper proposes a method to find the traffic density in real-time using python-OpenCV library for video processing. Deep learning with pre-trained model is also used to classify the vehicles.

II. LITERATURE SURVEY

There are several methods for detection and counting of vehicles automatically. Some of the techniques employed are discussed below.

Boris A et al., have proposed automatic detection and counting of vehicles from real time traffic. They have discussed the real-time problems of traffic. The detection and count of vehicles are performed using the algorithms for marking road, vehicle detection and count, to evaluate parameters of traffic flows. Analysis is considered with different conditions and situations [2].

D. D. Pukale et al., have designed a system which has implementation of both software and hardware system for vehicle counting. Background subtraction and blob tracker are used in software implementation and Arduino for hardware. The system performance needs to be improved when results are verified [3].

Honghong Yang et al., have developed real-time vehicle detection and counting from complex traffic scenarios using low-rank decomposition with background subtraction. This proposed system has shown good performance for GRAM dataset and change detection benchmark 2014[4].

Xuezhi Xiang et al., have proposed vehicle detection and tracking based vehicle counting from the aerial videos. This systemhas considered two scenarios: moving background and static background. The results of 16 videos for fixed and moving background are 90% and 85% respectively [5].

Fatma Kerouh et al., have designed real-time android application for estimation of traffic density. The consistency of evaluation using phone parameters under different altitudes, camera position and weather conditions are shown. This app has advantages with portability, cost, accuracy and deployment [6].

Xun Li et al., designed vehicle multi-target detection for traffic video using deep learning. YOLO, YOLO-voc and YOLO V3 models are compared and design of new network is shown which is called as YOLO-vocRV model. This model has learned from 20000 iterations. The result is suitable for multiple target detection with different traffic scenarios. The error rate for free traffic flow is 1.4%, false detection rate is 3.7% and accuracy of blocking flow was up to 96.3% [7].

Nilesh J Uke et al., have proposed detection of moving vehicle for finding the vehicle count. The techniques used for detection of vehicle and counting are background subtraction, filtering and segmentation. The analysis of detected vehicle is carried out to classify the vehicles as heavy weight or light weight or motorcycle. They have used pre-recorded videos for analysis purpose [8].

Reha Justin et al., have developed python with digital image processing based vehicle detection and counting. The techniques like edge detection, frame differentiation, kalman filter and object detection have been employed by them to detect and count the number of objects. [9].

Junyan Lu et al., have presented detection of vehicle from aerial images using YOLO. The results obtained for rotating objects, small objects are good and also model showed good performance for compact and dense objects by meeting real-time requirements. The result is displayed for limited training images and it can be further extended to different categories of images. [10].

Huansheng Song et al., have presented a yolov3 based model for detection of vehicles in highways. The detection of vehicles with proposed segmentation method has provided a good accuracy but vehicle classification is not shown [11].

Urban traffic analysis and complex scenario analysis are shown in [16] and [19] respectively.

Comparisons of existing approaches are shown in table I.

Table I: Comparison of existing methods from the literature

Ref.	Methods	Remarks
Yi-Qi Huang et al., 2020 [13]	Single stage Yolov3-DL	• Classification of only cars with accuracy of 98%
Adel Ammar et al., 2020 [21]	Car detection using CNN and yolov3	• Detected cars from aerial images • Comparison of CNN and yolov3 is shown. • Yolov3 gave better performance
Jun Liu et al., 2020 [20]	YoloV3 and tracking algorithm	• Different focal length camera on windshield • Detects cars, license plate numbers
Muhammad Fachrie 2020 [17]	Yolov3	• Classification for four types of vehicles with deep learning method • Accuracy of 97%
Dinh Viet Sang et al., 2019 [18]	Sparse network using variational dropout	• 91% of weights are eliminated • 90% accurate
Donato Impedovo et al., 2019 [12]	Visual features and comparison of deep learning methods	• Visual features with accuracy of 98% for deep learning • Not for complex traffic scenarios.
Manoharan, Samuel 2019 [22]	Artificial intelligence	• Safety algorithm using artificial intelligence for self driving cars is shown
Koresh et al., 2019 [23]	CNN & RNN	• 15% increase in accuracy.

III. METHODOLOGY

This work compares two methods for vehicle detection and counting. The first method uses opencv image processing library to detect and count the number of vehicles as shown in Figure 1. The virtual machine instance is created at server for video processing.

A. Method 1: vehicle count using blobs and morphological operations

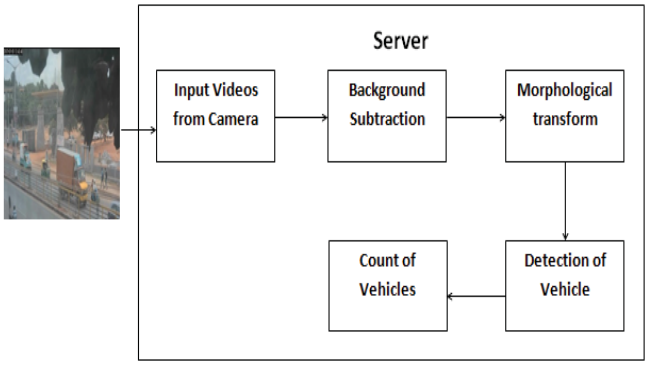


Fig 1: Vehicle counting system

i) Video Acquisition

The traffic video is collected from 4 Sony IP based HD cameras in different direction. The data is processed in a virtual instance created at server. The duration of data collection is for a week. Figure 2 displays an example video acquired in real-time traffic scenario.

Fig 2: Real-time traffic video acquired from IP camera in Tumkur city

ii) **Frame Extraction**

The processing of video starts with extracting the frames. Each frames of video is extracted from the cap.read() function in opencv shown as follows.

ret, frame = cap.read()

a) **Background Subtraction**

It is an effective technique for finding the moving objects in a frame. The fundamental concept of this technique is to find the difference in frame and apply thresholding to find the moving objects.

iii) **Morphological transform**

These are operations which consider image shapes. It is generally carried out for binary images. Morphological transform function takes two parameters, one is structuring element and another is original image. Structuring element decides the type of operation. Erosion and dilation are the two basic operations of morphological transform.

iv) Detection of vehicles

a) Blob detection

Blob is defined as collection of connected pixels that share similar property (E.g Grayscale value). OpenCV gives a suitable method to identify blobs and filter them based on diverse features. This also helps to find the trajectories of each vehicle. The algorithm is controlled by parameters shown below.

Thresholding:

minThreshold is used to convert the input images to many binary images with specific threshold value. The threshold value is incremented using thresholdStep till the maxThreshold. Hence the 1st value of threshold is minThreshold, 2nd is minThreshold + thresholdStep, 3rd is minThreshold + 2x thresholdStep and so on.

Grouping:

Connected white pixels are grouped for each binary image after thresholding. These are called as binary blobs.

Merging:

Center of binary blobs from each binary image is calculated and blobs which are closer than minDistBetweenBlobs are merged.

Center & Radius Calculation:

Center and radius of newly merged blobs are calculated and returned. This is used to mark the center of every vehicle detected. Blobs method can also be used to fine the direction and speed of vehicle movement.

B. Method 2: Vehicle count and classification using yolov3

a) Dataset

In this paper, real-time traffic data is collected for one week duration in Tumkur city, Karnataka, India. The data is collected from four Sony HD IP cameras oriented in different directions. Images have vehicle categories like car, bus, truck, motorcycle, pedestrian etc for typical Indian Roads. Fig 2 shows an example image from our dataset.

***b)* Training process**

All weights for YoloV3 model is initialized with the pre-trained weights from COCO dataset. COCO dataset is a large image object dataset with labels. It has 80 class labels for different objects. But proposed model requires only the classification for vehicles and people. Hence, it is custom trained to get only the required class labels. COCO pre-trained weights are used and customized for required classes.

A pre-trained model called darknet and yolov3 for real-time object detection is used. The architecture for vehicle detection is shown in figure 3. Vehicle detection target detection is shown in figure 4.

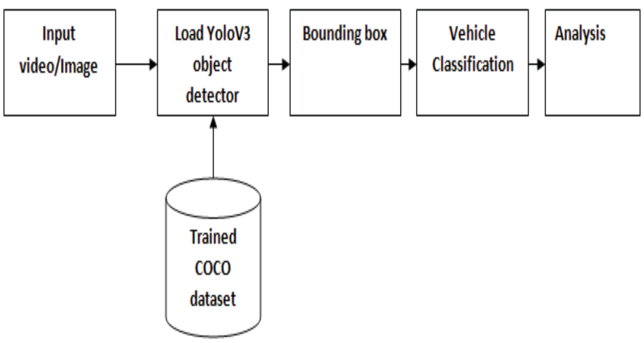


Fig 3: System architecture for vehicle classification using YOLOv3

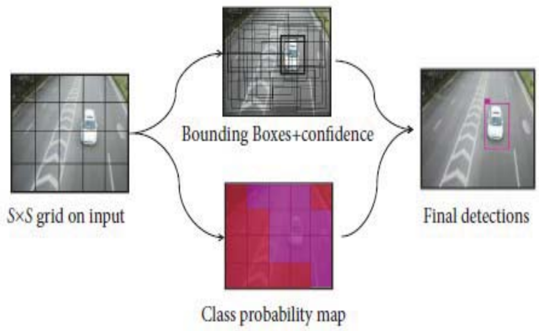


Fig 4: YoloV3 target detection method [15]

The methods and models used are described as follows

Darknet

This is a neural network model framework used for training. It is a 53 layered CNN and uses 3 × 3 and 1×1convolutional layer [14].

DNN

It is the library module present in opencv 3.1 that implements deep networks with forward pass and the pre-training is carried out using well-known framework of deep learning called as Caffe.

Confidence

The confidence shows the absence or presence of an object of any class. Low confidence value specifies that the object is misinterpreted by the network. Minimum probability to filter weak detections has default value 50% (0.5).

Threshold

This is the non-maxima suppression threshold with a default value of 0.3.

Non-Maximum Suppression

Applying non-maxima suppression suppresses significantly overlapping bounding boxes, keeping only the most confident ones. It also ensures that there are no redundant or extraneous bounding boxes.

The analysis for images with individual classification is implemented in python to display percentage of vehicles in pie-chart.

IV. RESULTS & DISCUSSIONS

i) Count of Vehicles: Method 1

The detected blob is used for counting the number of vehicles. The results are compared with manual counting and it is displayed in the table II. The result shows that the accuracy of the system also depends on the direction in which the camera is placed. Camera 1 and camera 2 shows good performance when compared to other two cameras.

Table II: Vehicle count with its accuracy for method 1

Sl. No	Cameras	Input videos	Manual Count	Automatic count	Accuracy
1	Camera 1	Video 1	13	15	86%
2	Camera 2	Video 2	36	47	76.59%
3	Camera 3	Video 3	24	45	53%
4	Camera 4	Video 4	33	56	58%

The results of method 1 using blobs and morphological operation are shown below.

Fig 5: Snapshot showing difference and background subtracted image

The pre-processing stage involves background subtraction and calculation of difference to detect vehicle as shown in figure 5.

Fig 6: Thresholded image

Thresholding provides the binary image to provide the proper vehicle detection and it is displayed in figure 6.

Fig 7: Blobs detected to track vehicle in motion (Trajectory)

Blobs are detected for tracking the vehicle in motion also called as trajectory detection. This trajectory helps to find the direction of vehicle movement and further it can be used to find the speed of vehicle. Figure 7 shows an example from real traffic video showing the blobs.

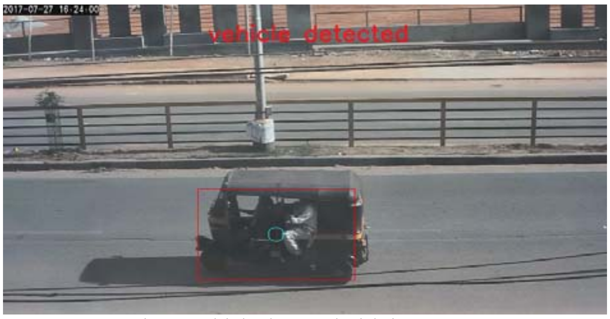


Fig 8: Vehicle detected with its center

Figure 8 shows the rectangle box drawn for the vehicle detected along with its centre. It immediately displays it as vehicle detected on screen.

Fig 9: Vehicle count displayed from camera 1

Camera 1 is oriented to one direction which focuses on one side of the road. It is a complex traffic environment where no lanes are present. Hence, manual lanes are drawn using image processing and vehicle is detected with blobs to count the number of vehicles crossing that lane. Figure 9 shows an example video frame taken from camera 1 and count of vehicles is also displayed.

Fig 10: Vehicle count displayed for camera 2.

Camera 2 is oriented in different direction to concentrate on another side of the road. Figure 10 shows an example video frame with count and trajectory detected for each vehicle.

Fig 11: Vehicle count taken from camera 3

Camera 3 is oriented in opposite direction as compared to camera 1 and camera 2. Figure 11 shows the example video frame shown the count of vehicles and trajectories.

Fig 12: Vehicle count taken from camera 4

Camera 4 is oriented to focus on two sides of road at a time. Figure 12 shows an example video frame from camera 4 with its count and trajectories.

This is the complex traffic scene with no lanes hence orientation of camera and weather conditions plays an important role on the performance.

ii) Vehicle count & classification: Method 2

Vehicle detection and its confidence score for individual vehicle in a video is shown in Table III. The confidence score describes the class probabilities. Table III shows correct result for classes like bus, car, motorcycle, truck but auto is misclassified because it is not considered in training process. Yolov3 can be used to customize the training process for auto in future.

Table III: Vehicle classification with confidence score

Sl. No	Vehicle	Confidence score	Classification result
1	Bus	92.56%	Bus
2	Car1	99.10%	Car
3	Car2	99.00%	Car
4	Motorcycle1	99%	Motorcycle
5	Motorcycle2	99%	Motorcycle
6	Person	99%	Person
7	Auto	98%	Fail (Misclassified as Car)

Accuracy is calculated using the following equation

$$Accuracy = \frac{number\ of\ predicted\ count}{number\ of\ manual\ count} * 100 \qquad (1)$$

Equation (1) describes the calculation of accuracy for video or image. Sensitivity and precision are calculated as follows.

$$Sensitivity\ or\ recall = \frac{True\ positives}{True\ positives + false\ negatives} \qquad (2)$$

$$Precision = \frac{True\ positives}{True\ positives + false\ positives} \qquad (3)$$

Equation (2) & (3) describes the calculation of recall and precision respectively. True positive (TP) specifies the number of correctly detected vehicle category, false positives (FP) indicates the number of incorrect detections and false negatives (FN) specifies the number of missed detections. In this paper, accuracy of yolov3 is better as shown in table IV. Table IV also shows recall and precision. Recall rate is better compared to precision because vehicle “auto” is not considered for classification in this paper.

Table IV: Yolov3 statistics

Sl. No	Input image	Manual Count	Predicted count	Accuracy (%)	Recall (%)	Precision (%)
1	Video 1	14	13	92%	90.9%	77%
2	Video 2	10	9	90%	88%	78%
3	Video 3	8	6	75%	71%	83.3%
4	Video4	7	7	100%	100%	85.7%

Fig 13: Vehicle classification

Figure 13 shows video frame showing the classification of vehicles with the confidence score.

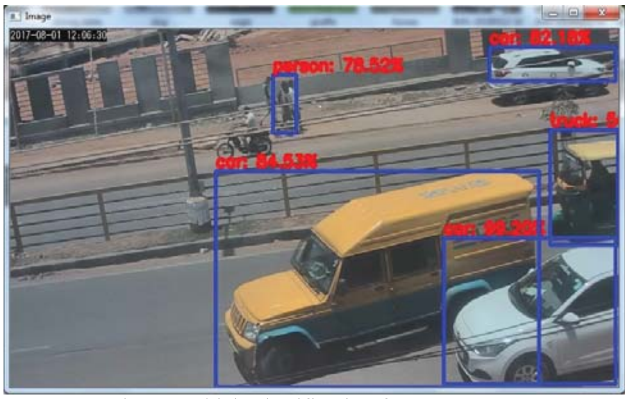


Fig 14: Vehicle classification from camera 1

Figure 14 shows vehicle classification from camera 1 with 99% of vehicle being detected.

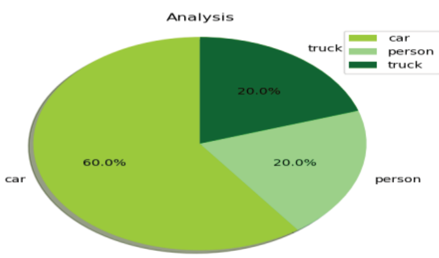


Fig 15: Analysis showing the percentage of each category of vehicles and percentage of people for figure 14

Figure 15 displays the percentage of vehicle classification as a pie chart for better visualization and it also specifies the percentage of people in that area. This helps to have a proper traffic management for both vehicles and crowd.

Fig 16: vehicle classification from camera 2

Figure 16 shows vehicle classification from camera 2 with few misclassifications for auto.

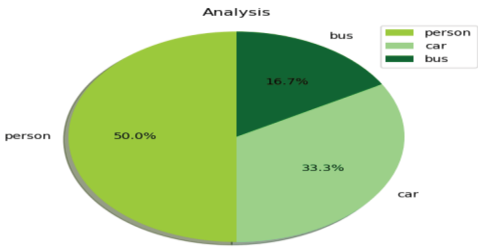


Fig 17: Analysis showing the percentage of each category of vehicles and percentage of people for figure 16

Figure 17 displays the pie-chart for figure 16 to provide the percentage of each category.

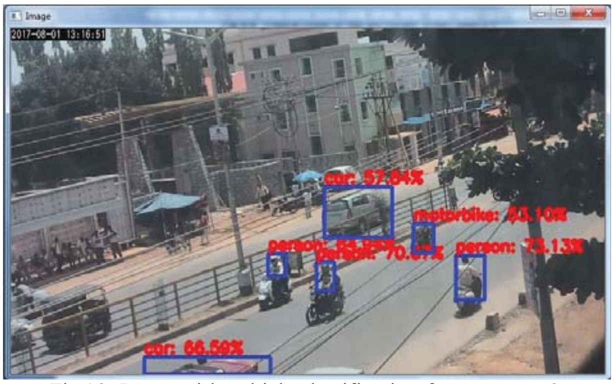


Fig 18: Image with vehicle classification from camera 3

Figure 18 shows vehicle classification from camera 3 with few missed detections shown.

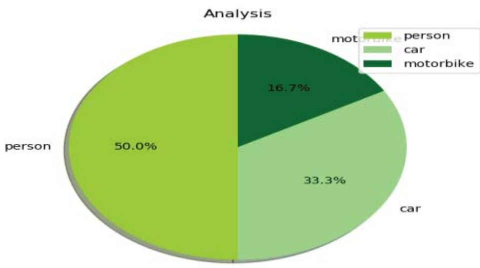


Fig 19: Analysis showing the percentage of each category of vehicles and percentage of people for figure 18

Figure 19 shows a pie-chart with percentage of each category of classification.

Fig 20: Image with vehicle classification from camera 4

Figure 20 shows the vehicle classification from camera 4 with 99% accurate class detection

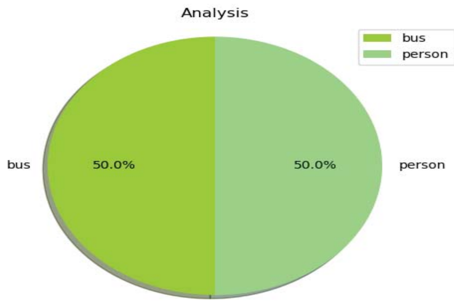
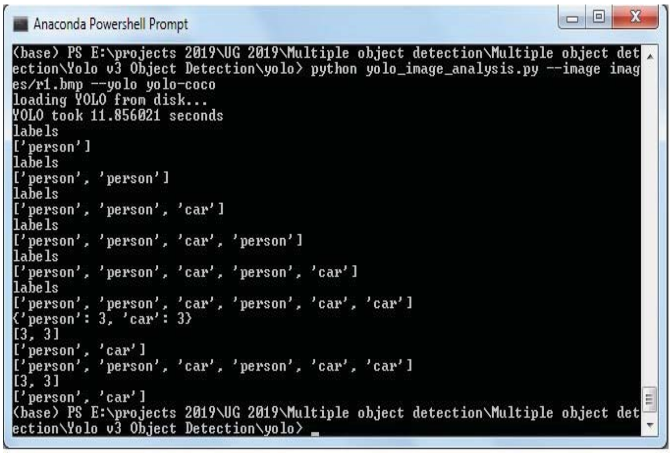


Fig 21: Analysis showing the percentage of each category of vehicles and percentage of people for figure 20

Figure 21 displays the pie-chart for figure 20 with percentage of each category.


Anaconda Powershell Prompt

```
(base) PS E:\projects 2019\UG 2019\Multiple object detection\Multiple object det
ection\Yolo v3 Object Detection\yolo> python yolo_image_analysis.py --image imag
es/r1.bmp --yolo yolo-coco
loading YOLO from disk...
YOLO took 11.856021 seconds
labels
['person']
labels
['person', 'person']
labels
['person', 'person', 'car']
labels
['person', 'person', 'car', 'person']
labels
['person', 'person', 'car', 'person', 'car']
labels
['person', 'person', 'car', 'person', 'car', 'car']
{'person': 3, 'car': 3}
[3, 3]
['person', 'car']
['person', 'person', 'car', 'person', 'car', 'car']
[3, 3]
['person', 'car']
(base) PS E:\projects 2019\UG 2019\Multiple object detection\Multiple object det
ection\Yolo v3 Object Detection\yolo> _
```

Fig 22: Individual classification and its estimated time shown in command prompt.

Figure 22 shows an example result showing the count of each category and the estimated time for classification using yolov3. Time taken can be improved by using GPU (Graphics Processing Unit).

The analysis of two methods clearly shows that yolov3 can be used for complex traffic scenarios as it shows the better performance.

V. CONCLUSION

There is possibility of large number of real-time applications of vehicle tracking and counting because of increased traffic. In this paper, two methods are compared and concluded that yolov3 is better for vehicle count and classification. An analysis is provided with better visualization using pie-chart showing the percentage of each category. This analysis helps for better traffic and crowd management in complex traffic scenarios (without the lanes). This paper shows a complex scenario with analysis of each classification for typical Indian roads with better accuracy from yolov3 model. It can be further extended for auto vehicle detection, speed calculation and direction of movement for each vehicle. The accuracy can be improved by considering the all possible vehicle category.

REFERENCES

[1] Debojit Biswas, Hongbo Su, Chengyi Wang, Jason Blankenship and Aleksandar Stevanovic, "An Automatic Car Counting System Using OverFeat Framework", Sensors 2017, vol 17, 1535; doi:10.3390/s17071535.

[2] Boris A. Alpatov, Pavel V. Babayan, Maksim D. Ershov, "Vehicle Detection and Counting System for Real-Time Traffic Surveillance", IEEE, JUNE 2018.

[3] D. D. Pukale, Palak Chauhan, Adhikari Siddhi Satish, Preeti Nawal & Neha Kumari, "Density Based Traffic Control System Using Video Processing", Imperial Journal of Interdisciplinary Research (IJIR) Vol-2, Issue-6, 2016.

[4] Honghong Yang, Shiru Qu, "Real-time vehicle detection and counting in complex traffic scenes using background subtraction model with low-rank decomposition", IET Intelligent Transport Systems, Vol. 12 Iss. 1, pp. 75-85, 2018.

[5] Xuezhi Xiang, Mingliang Zhai, Ning Lv and Abdulmotaleb El Saddik, "Vehicle Counting Based on Vehicle Detection and Tracking from Aerial Videos", Sensors 2018.

[6] Fatma Kerouh and Djemel Ziou, "Real-Time Android Application For Traffic Density Estimation", IEEE, Vol 6, 2018.

[7]. Xun Li ,Yao Liu, Zhengfan Zhao, Yue Zhang, and Li He, "A Deep Learning Approach of Vehicle Multitarget Detection from Traffic Video", Journal of Advanced Transportation, Volume 2018.

[8]. Nilesh J Uke, Ravindra C Thool, "Moving Vehicle Detection for Measuring Traffic Count Using OpenCV", 2013.

[9]. Reha Justin, Dr. Ravindra Kumar, "Vehicle Detection and Counting Method Based on Digital Image Processing in Python", ICSCAAIT, E-ISSN: 2348-2273, 2018.

[10]. Junyan Lu1, Chi Ma, Li Li, Xiaoyan Xing, Yong Zhang, Zhigang Wang, Jiuwei Xu, "A Vehicle Detection Method for Aerial Image Based on YOLO", Journal of Computer and Communications, 6, 98-107, 2018.

[11]. Huansheng Song, Haoxiang Liang, Huaiyu Li, Zhe Dai and Xu Yun, "Vision-based vehicle detection and counting system using deep learning in highway scenes", Springer openAccess 2019.

[12]. Donato Impedovo, Fabrizio Balducci , Vincenzo Dentamaro and Giuseppe Pirlo, "Vehicular Traffic Congestion Classification by Visual Features and Deep Learning Approaches: A Comparison", Sensors 2019, 19, 5213; doi:10.3390/s19235213.

[13]. Yi-Qi Huang, Jia-Chun Zheng, Shi-Dan Sun, Cheng-Fu Yang and Jing Liu, "Optimized YOLOv3 Algorithm and Its Application in Traffic Flow Detections", Appl. Sci. 2020, 10, 3079, doi:10.3390/app1009307.

[14]. Bilel Benjdira, Taha Khursheed, Anis Koubaa, Adel Ammar, Kais Ouni, "Car Detection using Unmanned Aerial Vehicles: Comparison between Faster R-CNN and YOLOv3", 2018.

[15]. Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, arXiv, May, *2016.*

[16]. Fukai Zhang, Ce Li and Feng Yang, "Vehicle Detection in Urban Traffic Surveillance Images Based on Convolutional Neural Networks with Feature Concatenation", Sensors 2019, 19, 594; doi:10.3390/s19030594.

[17]. Muhammad Fachrie," A Simple Vehicle Counting System Using Deep Learning with YOLOv3 Model", DOI: 10.13140/RG.2.2.15026.56001, 2020.

[18]. Dinh Viet Sang, Duong Viet Hung, "YOLOv3-VD: A sparse network for vehicle detection using variational dropout", SoICT 2019, December 4–6, 2019, Hanoi - Ha Long Bay, Viet Nam.

[19]. Lecheng Ouyang, Huali Wang, "Vehicle target detection in complex scenes based on YOLOv3 Algorithm", AMIMA 2019.

[20]. Jun Liu and Rui Zhang, "Vehicle Detection and Ranging Using Two Different Focal Length Cameras", Hindawi, Journal of Sensors, Volume 2020.

[21]. *Adel Ammar1, Anis Koubaa1, Mohanned Ahmed1, Abdulrahman Saad, "Aerial Images Processing for Car Detection using Convolutional Neural Networks: Comparison between Faster R-CNN and YoloV3", IEEE, 2020.*

[22].Manoharan, Samuel. "An improved safety algorithm for artificial intelligence enabled processors in self driving cars." Journal of Artificial Intelligence 1, no. 02 (2019): 95-104.

[23]. Koresh, M. H. J. D., and J. Deva. "Computer vision based traffic sign sensing for smart transport." Journal of Innovative Image Processing (JIIP) 1, no. 01 (2019): 11-19.