An Efficient Vehicle Counting Method Using Mask R-CNN

Zaynab Al-Ariny[1], Mohamed A. Abdelwahab[2,3], Mahmoud Fakhry[4], and El-Sayed Hasaneen[4]
[1]Electrical Engineering Department, Faculty of Engineering, Sohag University, Sohag, Egypt
[2]Electrical Engineering Department, Faculty of Energy Engineering, Aswan University, Aswan, Egypt
[3]Graduate School of Information Science and Electrical Engineering, Kyushu University, Fukuoka 819-0395, Japan
[4]Electrical Engineering Department, Faculty of Engineering, Aswan University, Aswan, Egypt

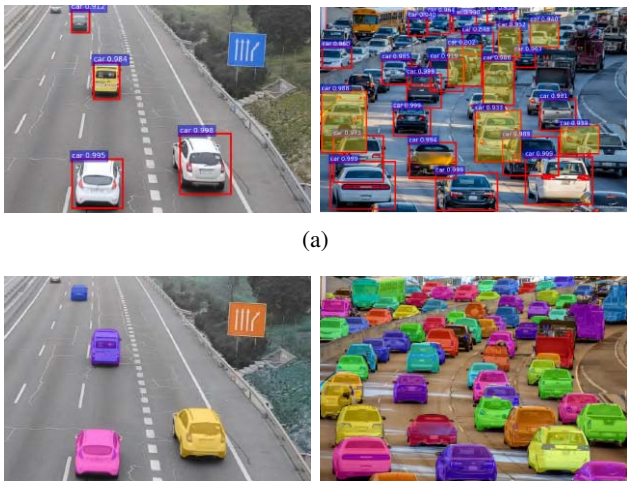
***Abstract*—In this paper, an accurate approach for vehicle counting in videos using Mask R-CNN and KLT tracker is proposed. Vehicle detection is performed for each N frames using Mask R-CNN instance segmentation model. This model outperforms other deep learning models that using bounding box detection as it provides a segmentation mask for each detected object, the outperformance comes up clearly in cases of occlusions. Once the objects are detected, their corner points are extracted and tracked. An efficient method is introduced to assign point trajectories to their corresponding detected vehicles. The proposed counting algorithm distinguishes precisely between the new vehicles and the counted ones. The experiments performed on diverse challenging videos show excellent results compared to state-of-the-art counting methods.**

Keywords*—Vehicle Counting, DNN, Mask R-CNN, KLT Tracker.

I. Introduction

Traffic control is one of the most trended research nowadays. The importance of traffic control is represented in providing information about road occlusions, accidents and any type of road disruptions. So, the relevant authorities can take a suitable response and drivers can be directed to better routs. The main task to achieve this goal is "vehicle counting", which can be performed through two main steps, vehicle detection, and vehicle tracking. Many approaches have long been existed using hand-crafted methods [1–3]. But the most recently developed approaches move towards the deep learning methods [4, 5], which get ahead of the other hand-crafted methods.

The efficiency of vehicle detection is considered as the key point for vehicle counting or traffic monitoring in general. Many researchers provide different vehicle detection approaches since the beginning of the present century as traffic monitoring becomes a trend. Several detection approaches are done based on the differentiation between the moving foreground and the static scene of the background, these approaches are known as motion-based approaches. For example, the vehicle objects can be detected by detecting the difference through the consecutive frames as explained in [6]. Another method described in [7] uses many samples of the background scene to build a model for the static background. This model is used to subtract the background and so that the foreground objects can be detected. In [8], authors introduce an adaptive bounding box size. The Gaussian Mixture Modeling (GMM) is used to create background model, then the vehicle objects are detected from the foreground. In other approaches, common distinctive characteristics of vehicles (appearance features) are exploited. These characteristics are usually represented in color, shape or textures. By detecting appearance features, distinguishing vehicle objects from the other objects can be performed. Many different effective feature extraction methods are proposed for this appearance-based approach such as using of Histogram of Oriented Gradient (HOG) and scale Invariant Feature Transformation (SIFT) as introduced in [9] and [10] respectively.



(a)

(b)

Fig. 1: (a) Samples of vehicle bounding-box detection using R-CNN in different traffic density scenes. (b) The corresponding samples of Instance Segmentation Detection using Mask R-CNN.

For vehicle counting, several approaches are provided. For example, the authors in [2] suggest a method to estimate vehicle numbers from a top-view video sequence through two steps. First, detection training is performed using an incremental PCP-based algorithm. Then, a process including two steps is used for estimation of vehicle numbers in the captured frame. Initially, number of vehicles is estimated using the spatial information of the capture frame. Then, information from previous frames used to refine the previous guess. In [11], a vehicle counting approach in airborne or stationary camera videos is given, where the vehicle detection and tracking are achieved simultaneously. First, trajectories are extracted by tracking corner points, then the stationary background trajectories are removed based on the histogram changes of pixels around corner points. The remaining trajectories are divided into separate vehicle trajectories based on their motion properties. Finally, vehicles are counted exploiting the obtained trajectories. The suggested approach in [3] is using the background subtraction method

978-1-7281-4801-4/20/$31.00

to extract vehicle contours from a video frame. If a vehicle centroid position is detected within a predefined virtual detection area, then the vehicle is tracked and counted. In [12], a fast vehicle counting method was introduced by creating a background model for a narrow region. Vehicle counting is achieved by considering only the detected vehicles in this narrow region, so no need for the tracking step and a fast vehicle counting is achieved.

All the previous approaches give good results, but still there are some difficulties affect the detection accuracy. For example, motion-based detection methods are highly affected by the dynamic background environment as the idea of these methods mainly depends on the vehicle motion. The motion resulting from rains, snow or even illumination changes maybe affect the detection accuracy. For appearance-based approaches, good knowledge about the object characteristics is necessary needed. Also, much work is required to get accurate and reliable feature extraction to obtain good detection results [1]. Deep learning methods overcome all these difficulties and are considered as the state-of-the-art technology in the computer vision tasks. One of the suggested approaches using deep learning methods is explained in [4]. In this approache, a vehicles counting method using Deep Neural Network (DNN) is introduced where the detection is done in each frame. However, the vehicle counting is carried out for static images not successive frames of a video. In [5], the author introduces an efficient vehicle detection and counting approach using a deep neural network and Kanade-Lucas-Tomasi (KLT) tracker [13]. Vehicle detection is carried out using Regions with Convolutional Neural Networks (R-CNN) [14]. A fine-tuning is performed using 295 vehicles image to get better detection results for vehicle objects. Although this approach gives good results, a problem comes up in the case of high occlusion regions as shown in Fig 1. As clear in this figure, R-CNN gives good performance for normal traffic density. However, in high traffic density the overlapping of detection bounding boxes occurs and leading to a detection problem.

In this paper, we propose an efficient approach that handles the occlusion dilemma. An instance segmentation detection is performed using Mask R-CNN [15] which is more accurate than the bounding boxes detection as no overlapping occurs. Then, an effective algorithm is proposed to find corner points of the detected vehicles and assign each corner point to the corresponding vehicle segment so counting is carried out more accurately. The rest of this paper is structured as follows: a brief description of the Mask R-CNN is given in section II. The proposed approach is explained in detail in section III. Section IV introduces the achieved results. Finally, section V represents the conclusion.

II. Mask R-CNN

Mask R-CNN is an object instance segmentation framework first published in 2017. It was developed by enhancing the Faster R-CNN by adding a parallel channel to give a segmentation mask for each detected object in addition to the bounding-box and the class label. Mask R-CNN is performed by two main steps as shown in Fig. 2. First, Region Proposal Network (RPN) is used to predict the regions that may contain objects. Pixel-to-pixel alignment occurs using the RoIAlign layer, which is the magic key of Mask R-CNN. Then, two extended R-CNN heads are used to generate the object bounding box, class label and object mask in parallel [15].

We can think of instance segmentation detection as pixel-wise detection, which is more accurate than bounding-box detection. Identification is done for each pixel of all known objects in an input image. Fig. 1-a shows vehicles detection examples of two different frames using R-CNN bounding-box detection. It gives a good detection result as shown in the first frame, but it is coming up with some detection errors in the regions where vehicles are very crowded or occluded as shown in the second frame. Samples of bad detection regions are yellow highlighted. We can notice that in some cases, multiple vehicles are detected as one vehicle. In some other cases, the detected bounding box of a vehicle includes a part of other vehicles. Fig. 1-b shows output for the same frames with instant segmentation detection using Mask R-CNN. The results given by Mask R-CNN are better than that of the bounding-boxes detection. We can notice that even occluded vehicles are highly identified by separated masks without overlapping.

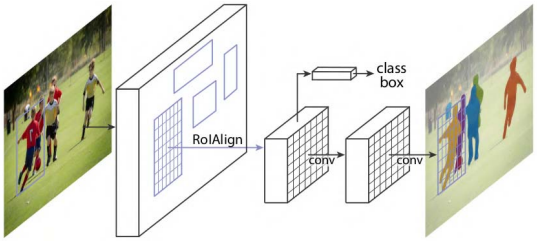


Fig. 2: The main steps of Mask R-CNN framwork. [15]

(a) (b) (c)

Fig. 3: (a) Sample of a captured frame (b) The chosen ROI mask (c) The frame after applying the ROI mask.

III. The Proposed Approach

The proposed approach can be summarized in four steps: frame preparation by applying region of interest (ROI) mask, vehicle detecting using Mask R-CNN [15], Vehicle tracking using KLT tracker [13], and vehicle counting. Fig 5 summarizes the proposed approach steps. In the following section, we are going to explain these steps in detail.

A. Frame Preparation

For each captured frame, a ROI mask is applied to the frame before the detection step to reduce the required detection time. The ROI mask is chosen to extract a suitable part of the frame where vehicles maybe exist. The detection is performed only within this area so that no extra time is consumed to detect non-vehicle objects far away from the road. Fig. 3 shows one sample of the used ROI masks.

B. Vehicle Detection

Vehicle detection is a critical step in our approach, therefore Mask R-CNN is used in the detection step to get accurate instance segmentation detection results. To reduce the time complexity, the detection step is executed every N frames. Depending on the scope of our research, the Mask R-CNN model has been modified to detect vehicle objects only.

The Mask R-CNN model provides segments with a unique label for each detected object. Fig. 4 shows an example of vehicle detection using Mask R-CNN where pixels of each detected vehicle has its unexampled label so we can distinguish between different vehicles by its segment label.

(a) (b)

Fig. 4: (a) A captured frame sample (b) The corresponding segmentation mask result from the detection step using Mask R-CNN model, the unique label of each detected vehicle is represented by a different segment color.

C. Vehicle Tracking

It is necessary to track each detected vehicle object through the video frames to make sure that each vehicle is counted only once even if it is detected multiple times. A counting method based on KLT tracker [13] is used for this step. The corner points are detected and the tracking is made by tracing the small change in coordinates for each detected corner point from one frame to another. In this step, two basic tasks are implemented:

1) Searching for new upcoming corner points included in the detected vehicle segments and tracking them. In this task, two processes are executed in parallel to reduce the required processing time:
   a) Detecting vehicle objects within the ROI using the Mask R-CNN.
   b) Extracting the new corner points.

   After that, the segmentation mask from the detection step is used to discard all corner points outside the segments which do not belong to any vehicle segment. The remaining corner points are assigned to their vehicle segments i.e, take their labels. Then, the tracking is performed for the new and the old corner points (which are detected in previous N-frames).
2) Tracking the already detected points from the previous frame and update the xy coordinates with the new positions.

The detection of new corner points is done only for each N frames, where Mask R-CNN object detection is performed. On the other hand, the tracking of the previous corner points should be executed for each captured frame. The vehicle detection is done only within the ROI, so there is no need to keep tracking corner points that are located outside the ROI. Therefore, every N frames, all the corner points outside the ROI are deleted to reduce the time complexity of the tracking step.

D. Vehicle Counting

The proposed counting approach is based on detecting and tracking corner points within the detected vehicle segments. The counting is updated every N frames. For each detected vehicle segment in a frame, we check all the corner points located within it as the following:

1) If all corner points within a detected vehicle are new detected points, then this vehicle is considered to be detected for the first time and it takes a new label.
2) If all corner points within a detected segment are old tracked points, then this vehicle object is already detected in a previous frame.

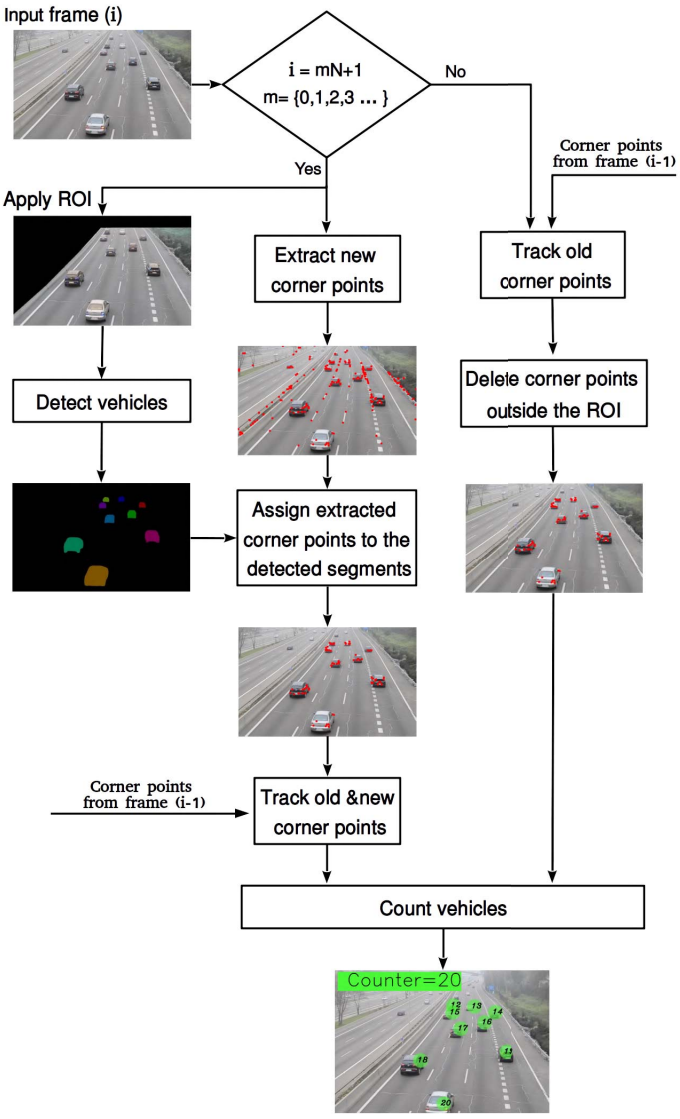


Fig. 5: Block diagram of the proposed counting system.

3) If the segment contains some of old tracked corner points and some new detected points, then the vehicle object is also already detected in the previous frame.

A counter is incremented each time a vehicle object is detected for the first time, as described in the first case. Therefore, the counter represents the number of detected vehicles at any time through the video frames. Fig. 5 shows a simple block diagram of the proposed system.

IV. Experimental Results

The proposed approach is tested in two experiments using various video datasets that recorded under different conditions. The first experiment is executed using M-30 and M-30-HD videos from the GRAM-RTM dataset provided in [16]. The achieved results compared with the results of [17], [2], [18], and [5] which tested their counting approaches using the same dataset. In the second experiment, we test the proposed approach using HighwayI and HighwayII videos and compare the results with those achieved by [18] and [5] which also used the same videos for their experiments. The results are compared by calculating the precision in each case. The precision is computed using equation 1. In this section, we are discussing the result achieved by the two experiments

Frame 15

Frame 120

Frame 180

Frame 460

Frame 795

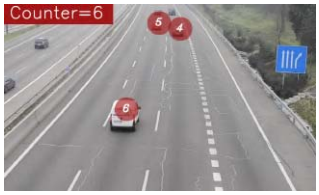
Frame 1110

Frame 1245

Frame 2040

Fig. 6: Samples of the proposed counting approach's results using the M-30 video.

Frame 1

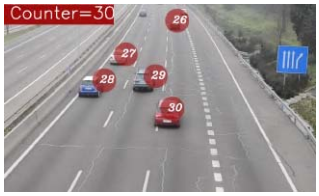


Frame 220

Frame 465

Frame 550

Frame 735



Frame 900

Frame 1080

Frame 1410

Fig. 7: Samples of the proposed counting approach's results using the M-30-HD video.

TABLE I: Vehicle counting precision for GRAM dataset.

Video name\True No.	M-30\77		M-30-HD\42	
	Count	Precision (%)	Count	Precision (%)
C. Bouvie [17]	69	89.62	33	78.57
J. Quesada [2]	75	97.41	39	92.86
H. Yang [18]	71	92.2	37	88.1
M. Abdelwahab [5]	72	93.51	42	100
Proposed Approach	76	98.7	42	100

TABLE II: Vehicle counting precision for the HighwayI and HighwayII videos.

Video name	HighwayI (Precision %)	HighwayII (Precision %)
H. Yang [18]	93.3	92.31
M. Abdelwahab [5]	96.43	95.83
Proposed Approach	100	97.9

in more detail.

$$Precision(\%) = 100 - Error(\%)$$
$$where \quad Error(\%) = \frac{|TrueNo. - Estimated|}{TrueNo.} \times 100 \quad (1)$$

The proposed approach is developed using python 3.6, tensorflow 1.3.0, and keras 2.2.4 running on Ubuntu 18.04.2 LTS. The detection step is developed using a pre-trained python implementation of Mask R-CNN model which is available for free downloading from [19]. The model is modified to detect the vehicle objects only (car, motorcycle, bus or truck). As explained in section III-B, the detection is executed every N frames to minimize the processing time. In all experiments, N is selected to be 15. The model provides a score for each detected object, which represents the confidence level of the detection. For the results achieved in this paper, we defined a score threshold = 0.8 to discard low score detected vehicles and get better detection results.

A. Experiments I

M-30 and M-30-HD videos are used in this experiment. M-30 is recorded on a sunny day with 800×480 resolution. We test the proposed approach on the first 2040 frames of M-30, where the true number of vehicles passed through this part of the video is 77 vehicles. Table I shows the precision of the proposed approach results and the other previous methods results. Our approach gives 98.7% precision which is the highest precision among the previously introduced methods. It counts 76 from 77 vehicles as appear from the samples of the results shown in Fig. 6.

M-30-HD video is recorded in the same road location of M-30 but on a cloudy day. It has a 1200×720 resolution, which is much better than the resolution of M-30. The proposed approach is tested on the first 1440 frames, where the true number of the passing vehicles is 42. The proposed approach count 42 vehicles which represent 100% precision. As shown in table I, this is the same precision as achieved by [5] and it is much better than the precision obtained by the other methods. Fig. 7 shows some samples of the counting results using M-30-HD video.

B. Experiments II

In this experiment, we test our proposed approach using HighwayI and HighwayII videos which are recorded in worse conditions than those videos used in the first experiment. HighwayI video is recorded in a daytime with a 320×240 resolution. Even though the challenge in this video is repre-

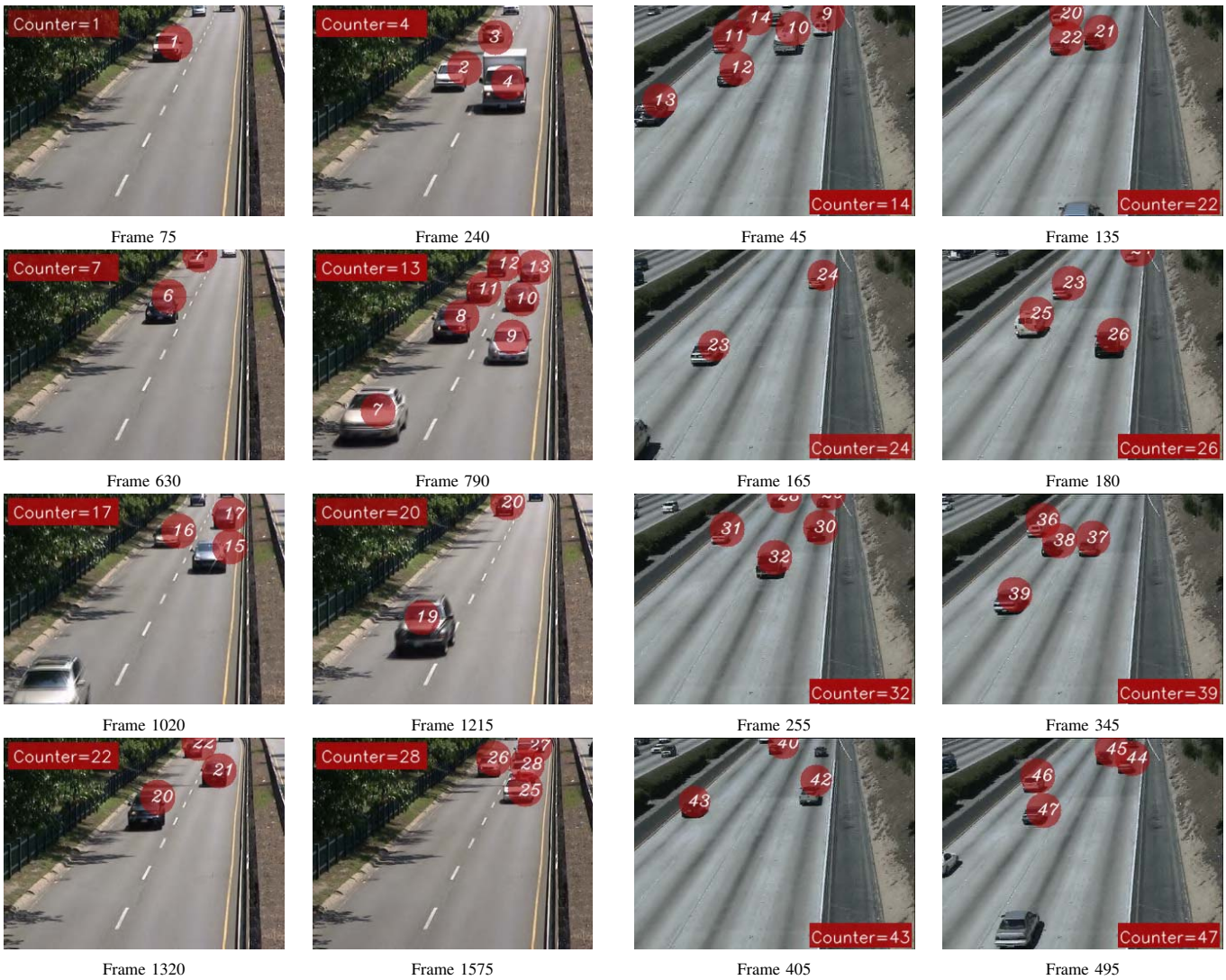


Fig. 8: Samples of the proposed counting approach's results using the HighwayI video.

Fig. 9: Samples of the proposed counting approach's results using the HighwayII video.

sented in the trees' shadow that occluded with the vehicles, the proposed approach counts 28 vehicles which is the true number of vehicles in this video. Table II shows that the achieved precision of 100% surpasses the precision obtained by the other two methods introduced in [18] and [5] which are 93.3% and 96.43% respectively. Fig 8 shows some samples of the counting results.

HighwayII video is recorded also in a daytime with 320×240 resolution, but for a wider highway. The scene in this video contains many crowded vehicles. The proposed approach counts 47 out of 48 which represents 97.9% precision compared to 91.31% and 95.83% obtained by the other methods. The missed vehicle is not detected due to the low resolution of the video. Fig. 9 shows some samples of the proposed approach results using this video. For both videos in this experiment, the proposed approach achieves the best results.

V. Conclusion

In this paper, an efficient approach for vehicles counting over video frames was proposed. Instance segmentation detection by Mask R-CNN model was used to detect vehicles accurately. Then, each detected vehicle was counted and tracked through the video frames employing the KLT tracker. The achieved results show that the proposed approach gives good performance and a higher precision than other counting methods. The superiority of the proposed approach is represented in the precise detection in crowded regions and the accurate proposed counting algorithm. Finding a robust detection model with high performance for the low-resolution video to give better result is considered as future work. Also using a faster detection model will enhance the performance of the proposed method.

References

[1] S. Kamkar and R. Safabakhsh, "Vehicle detection, counting and classification in various conditions," in *ET Intelligent Transport Systems*, 2015.

[2] J. Quesada and P. Rodriguez, "Automatic vehicle counting method based on principal component pursuit background modeling," in *IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 3822–3826.

[3] N. Seenouvong, U. Watchareeruetai, C. Nuthong, K. Khongsomboon, and N. Ohnishi, "A computer vision based vehicle detection and counting system," in *8th International Conference on Knowledge and Smart Technology (KST)*, 2016.

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

[5] M. A. Abdelwahab, "Accurate vehicle counting approach based on deep neural networks," in *2019 International Conference on Innovative Trends in Computer Engineering (ITCE)*. IEEE, 2019, pp. 1–5.

[6] H. Jia-feng, "Vehicles detection based on three-frame-difference method and cross-entropy threshold method," 2011.

[7] S. Gupte, O. Masoud, R. Martin, and N. Papanikolopoulos, "Detection and classification of vehicles," in *IEEE Transactions on Intelligent Transportation Systems*, 2002.

[8] S. Sivaraman and M. M. Trivedi, "Looking at vehicles on the road: A survey of vision-based vehicle detection, tracking, and behavior analysis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 14, 2013.

[9] N. Buch, J. Orwell, and S. Velastin, "3d extended histogram of oriented gradients (3dhog) for classification of road users in urban scenes," in *1st Iberian Robotics Conference*, 2014.

[10] J.-Y. . Choi, K.-S. Sung, and Y.-K. Yang, "Multiple vehicles detection and tracking based on scale-invariant feature transform," in *2007 IEEE Intelligent Transportation Systems Conference*, 2007.

[11] M. A. Abdelwahab and M. M. Abdelwahab, "A novel algorithm for vehicle detection and tracking in airborne videos," in *IEEE International Symposium on Multimedia (ISM)*. IEEE, 2015, pp. 65–68.

[12] M. Abdelwahab, "Fast approach for efficient vehicle counting," *Electronics Letters*, vol. 55, no. 1, pp. 20–22, 2019.

[13] J. Shi and C. Tomasi, "Good features to track," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, June 1994, pp. 593–600.

[14] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[15] H. Kaiming, G. Georgia, D. Piotr, and G. Ross, "Mask r-cnn," in *2017 IEEE International Conference on Computer Vision*, 2017.

[16] R. Guerrero-Gomez-Olmedo, R. J. Lopez-Sastre, S. Maldonado-Bascon, and A. Fernandez-Caballero, "Vehicle tracking by simultaneous detection and viewpoint estimation," in *IWINAC 2013, Part II, LNCS 7931*, 2013, pp. 306–316.

[17] C. Bouvie, J. Scharcanski, P. Barcellos, and F. L. Escouto, "Tracking and counting vehicles in traffic video sequences using particle filtering," in *IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, 2013, pp. 812–815.

[18] H. Yang and S. 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, no. 1, pp. 75–85, 2017.

[19] W. Abdulla, "Mask r-cnn for object detection and instance segmentation on keras and tensorflow," https://github.com/matterport/Mask_RCNN, 2017.