Vehicle Counting for Traffic Management System using YOLO and Correlation Filter

Asha C S
Electronics and Communication
National Institute of Technology
Karnataka Mangalore

A V Narasimhadhan,
Electronics and Communication
National Institute of Technology
Karnataka, Mangalore

***Abstract*—Vehicle counting is a process to estimate the road traffic density to assess the traffic conditions for intelligent transportation systems. With the extensive utilization of cameras in urban transport systems, the surveillance video has become a central data source. Also, real-time traffic management system has become popular recently due to the availability of handheld/mobile cameras and big-data analysis. In this work, we propose video-based vehicle counting method in a highway traffic video captured using handheld cameras. The processing of a video is achieved in three stages such as object detection by means of YOLO (You Only Look Once), tracking with correlation filter, and counting. YOLO attained remarkable outcome in the object detection area, and correlation filters achieved greater accuracy and competitive speed in tracking. Thus, we build multiple object tracking with correlation filters using the bounding boxes generated by the YOLO framework. Experimental analysis using real video sequences shows that the proposed method can detect, track and count the vehicles accurately.**

Keywords—YOLO, correlation filter, scale estimation, counting

I. INTRODUCTION

Due to increase in the number of roads and vehicles, traffic control has become an essential part of intelligent transport system. Many researches have been conducted for traffic management applications based on image and video processing approaches. The analysis of traffic video data include detection/recognition of vehicles, measurement of vehicle's speed, generation of tracking trajectory, counting of vehicles, congestion of traffic and collisions of vehicles. These applications have become popular recently due to the availability of low-cost cameras and embedded devices. Thus, the video data analytic is one of the prime research focus in the computer vision and big-data area.

Real time videos pose several challenges for an automated traffic analysis. The difficulties faced by the automated system include the presence of shadows, occlusion of vehicles, environmental variations such as rain, fog, dust, etc., which normally degrade the performance. Despite many dedicated efforts, an accurate method for vehicle counting under complex environment is still far from being achieved. There are several methods attempted to perform highway traffic counting based on videos. However, they are restricted to a particular road scenario and cannot be generalized.

Normally, vehicle counting is implemented by separating the dynamic part (moving objects) from the static part (background) of the scene. This is accomplished using the background subtraction (BS) and blob analysis method [1][6][7], background subtraction combined with Kalman filter [18], Gaussian Mixture Model (GMM) based background subtraction [6, 9] and particle filter based tracker [3][4]. BS is often achieved via frame averaging, Gaussian mixture model, and principal component images [8]. However, the background subtraction method has limitations while processing traffic data during heavy traffic conditions. It results in merging of vehicles during partial occlusion in the processed image data and predicting wrong bounding box. Besides, shadows of vehicles cause the inaccurate detection of vehicles. To improve the results, an expectation maximization is fused with a Gaussian mixture model to build the background model [2]. Background subtraction is then performed to extract the moving vehicles. The occlusions are handled using morphological features and color histograms [2]. In [5], vehicle counting for night-time videos has been proposed by extracting the bright regions (i.e. headlights), followed by the pairing of headlights, tracking, and counting. The techniques presented in [10] extract features and use classifier-based approach for vehicle counting.

The existing techniques mainly focus on the stable videos acquired by the fixed camera under simple background circumstances. In general, they extract the moving part of the scene based on background subtraction. Subsequently, blob analysis and tracking of bounding box is carried to count the vehicles. These techniques often fail to handle the occlusion, shadows, unstable (shaking) camera, incorrect position of the camera and complicated background. In order to mitigate these shortcomings, we proposed a novel method by combining object detection component with multiple object tracking. The proposed algorithm can process the video captured using stationary as well as unstable camera kept above or adjacent to the roads. In addition, the proposed algorithm can classify the objects into various types (car, bus, motorcycle, person, etc.). The contributions of the proposed method are as follows: (i) We acquired the dataset using handheld mobile camera placed at 4 different locations in a highway (ii) After that, a small region is cropped and fed into the YOLO framework to detect and classify the vehicles into 3 categories. i.e., small (motorbike, bicycle), medium (car, van, auto rickshaw) and large (lorry, bus, truck) (iii) Multi-object tracking is carried out for each detected sample by means of scale adaptive

978-1-5386-1112-8/18/$31.00 ©2018 IEEE

correlation filter (iv) Finally, vehicles are counted based on simple rules applied to tracking trajectory.

The paper is organized as follows: Section II explains the related work on object detection and tracking, the proposed work is explained in section III. Section IV details the experimental setup and results obtained using real highway traffic videos followed by the conclusion in section V.

II. RELATED WORK

A. *Object detection and recognition using YOLO*

The real-time object detection is an active area in the computer vision field and abundant researches have been proposed in the literature. At first, Haar features based cascaded Adaboost classifier has been proposed for face detection [11]. Later, Dalal et al. proposed Histogram of Gradient (HoG) based Support Vector Machine classifier for detecting the pedestrians [12]. Deformable Parts Model (DPM) has become attractive to identify the object using HoG and part based techniques [13]. Recently, deep learning based approaches have been widely used due to the availability of Graphical Processing Units (GPUs) and huge amount of datasets. These techniques [14][15] use CNN features with either sliding window or selective search method which is a time-consuming process. However, a robust method YOLO [15] treats the object detection as a regression problem to map pixels into bounding boxes with class probabilities. Moreover, it computes everything in a single evaluation, as a result, reports 45 frames/second speed.

Motivated by the generalization property, performance accuracy and speed of YOLO [15], we accommodate in the proposed work to serve the detection purpose. YOLO treats detection process as a regression problem to map the image into object bounding boxes. In this, the input image divides into M x M (ex. 13 x 13) grids, and each grid predicts B (ex.5) bounding boxes with the confidence scores. Each bounding box is associated with x, y, w, h, confidence score as predictions. Also, each grid cell predicts C (ex. 20) conditional class probabilities to indicate the likelihood of object in it. YOLO [15] has been trained using PASCAL VOC dataset and can predict 20 classes such as bicycle, boat, car, bus, person, motorbike etc. The confidence score of each bounding box and class predictions are combined to estimate the object. Thus, a single convolutional neural network is utilized to predict the bounding box in single evaluation.

B. *Correlation filter based tracking*

Object tracking is the process of maintaining the moving vehicle's trajectory in every frame of the video. The tracker locates the object in every frame starting from the initial bounding box. The bounding box can be user-specified or output of the object detector. The proposed work exploits the Correlation Filter (CF) based tracker [16] to track the vehicles. The ability of CF tracker to track multiple objects [17] in real time has motivated us to use in the proposed method. Moreover, scale estimation is an essential component in vehicle tracking due to the large variation of vehicle size. The block diagram of CF tracker is shown in the Figure 1. CF is trained using the k dimensional HoG features of an object collected from online data with Gaussian template as the desired output. Thus, for each example x, its feature vector is denoted as x^l ($l = \{1.....k\}$) is multiplied by a cosine window to smoothen the boundaries. For the desired Gaussian output y, the problem is formulated to generate the filter template with least error as:

$$\underset{h^l}{\operatorname{argmin}} \left\| \sum_{l=1}^{k} h^l * x^l - y \right\|^2 + \lambda \sum_{l=1}^{k} \left\| h^l \right\|^2 \quad [1]$$

Where h^l is the filter template in the spatial domain, λ is the regularization parameter, $*$ denotes the convolution operation. The solution of the Eq. (1) is obtained in the frequency domain as

$$H^l = \frac{Y \otimes X^l}{\sum_{l=1}^{k} X^l \otimes X^l + \lambda} \quad [2]$$

Where Y represents the Discrete Fourier Transform of y, X^l denotes the Discrete Fourier Transform of x^l, $\otimes$ denotes element-wise multiplication. In order to adjust to the new appearances, the filter template is updated in every frame. Accordingly, the numerator and denominator of Eq. (2) are updated as

$$N^l{}_t = (1-\eta) N^l{}_{t-1} + \eta Y_t X^l_t \quad [3]$$

$$D_t = (1-\eta)\, \mathrm{D}_{t-1} + \eta \sum_{l=1}^{k} X^l{}_t X^l{}_t \quad [4]$$

Where η is the learning rate fixed at 0.025. In every frame a rectangular patch z (Z in the frequency domain) is cropped and convolved with the filter template h^l in the frequency domain to produce the correlation output as

$$y = \Im^{-1} \left\{ \frac{\sum_{l=1}^{d} N^l{}_t \otimes Z}{D_t + \lambda} \right\} \quad [5]$$

The peak value of y determines the location of the target in the current frame. The size of the target is estimated in every frame using 1-D correlation filter, which is trained using 33 different scaled versions of the object. The size of the target in the present frame is found by searching the window with a maximum correlation score among the generated scaled patches. I-D scale filter is updated in every frame with learning parameter β.

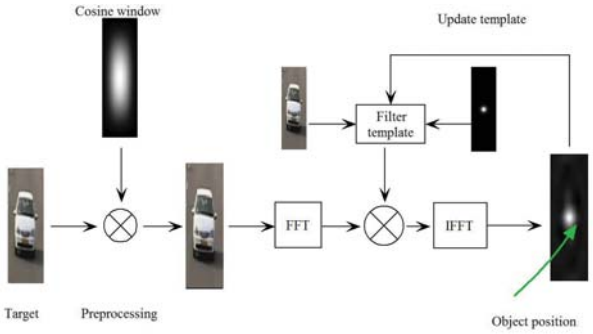

Figure 1: Block diagram of correlation filter based tracking.

III. PROPOSED METHODOLOGY

The proposed work combines the object detection unit with the correlation filter based tracker to count the traffic data. Existing methods based on background subtraction and blob tracking method achieve good results for fixed cameras only. However, it produces abnormal results for unbalanced handheld cameras. Hence, we combine state-of-the-art YOLO object detection framework to recognize the vehicles in video frames, followed by tracking using correlation filter with scale estimation. The working process of the proposed method is depicted in the Figure 2. The view of four different locations is shown in the Figure 3.

We initialize the counting process by cropping the region of interest in the first frame. The region is used as an entry window for each vehicle. The borders of the image frame act as an exit line for all the vehicles. Figure 4. depicts the sample frame and manually selected entry window. Thus the cropped area (entry window) is displayed in the Figure 5. YOLO framework is applied to entry window and each detection is used to begin the new track by assigning to a correlation filter based tracker after validating the object. All classes except "car", "bus", "motorcycle' are deleted from the detection process. The detection outputs are displayed in the Figure 6. and each object is denoted as OB_t^j. Here after, OB_t^j denotes the j^{th} object detected in the t^{th} frame using YOLO framework. Let t, $t-1$ and $t+1$ denote the present, previous and next frame respectively. Let CF_t^i denote i^{th} vehicle being tracked by the correlation filter in the t^{th} frame.

The overlap between two bounding boxes are defined as the ratio of intersection over union.

$$O = \frac{OB \cap CF}{OB \cup CF} \quad [6]$$

Where OB denotes the object bounding box and CF symbolises the tracked bounding box. The overlap factor O defines how well two bounding boxes overlap each other with 0 being no overlap and 1 indicates complete overlap. The following states are observed in the counting process.

1. Track state: The object detected in a frame may correspond to one or more tracked bounding boxes. If the overlap between OB_t^j and CF_t^i is higher than the pre-defined threshold τ (in this work, τ=0.3), then the corresponding object bounding box OB_t^j is already assigned to a CF tracker and condition of j^{th} vehicle is identified as tracked state. The size of the vehicle progressively increases in every frame due to the movement towards camera, hence scale adaptation is very essential. Thus, the correlation filter locates each vehicle precisely due to its high efficiency and scale estimation property. Its states are updated in every frame based on the vehicle motion and are referred as active trackers. The correlation filters stop updating when the vehicle disappears from the camera view. Subsequently, they are added to the passive trackers' list after eliminating from the active trackers' list.
2. Detect state: If the overlap between OB_t^j and CF_t^i is less than the predefined threshold τ (in this work, τ=0.3), then the OB_t^j is considered to be distinct object j which is detected in the frame t. In this condition, the tracked object is isolated from the detected object by an adequate distance. This state is identified as the detected state. There after the detected object is assigned to a new correlation filter based tracker to initiate the tracking process. The corresponding tracker is added to active trackers' list.
3. Terminate state: If the coordinates of tracker corresponding to each vehicle CF_t^i reaches the border of the frame, then the condition is defined as terminated. If the object is occluded, it reappears in the scene after a short period. However, object disappears completely as the result of vehicle moving out of camera view. Consequently, the vehicle count is incremented. The corresponding tracker is removed from the active trackers' list and added to the passive trackers' list. The vehicle count for each category is also incremented independently based on entity type.
4. Target lost state: The tracking accuracy is proved to be high using correlation filter [16]. However, it may lose target when smaller vehicle is occluded by the larger vehicle. This condition is referred as target lost. The failure of tracking is detected, when the bounding box do not move in any direction. To resolve this problem, the following assumptions are considered in the proposed work. They include (i) all the vehicles assigned to CF_t^i move in one direction. (ii) If the tracking of a vehicle is lost due to occlusion or fast motion then the corresponding vehicle will reach the boundary. Accordingly, the tracker is terminated and vehicle count is incremented. The flowchart of the proposed method is given in the Figure 2.

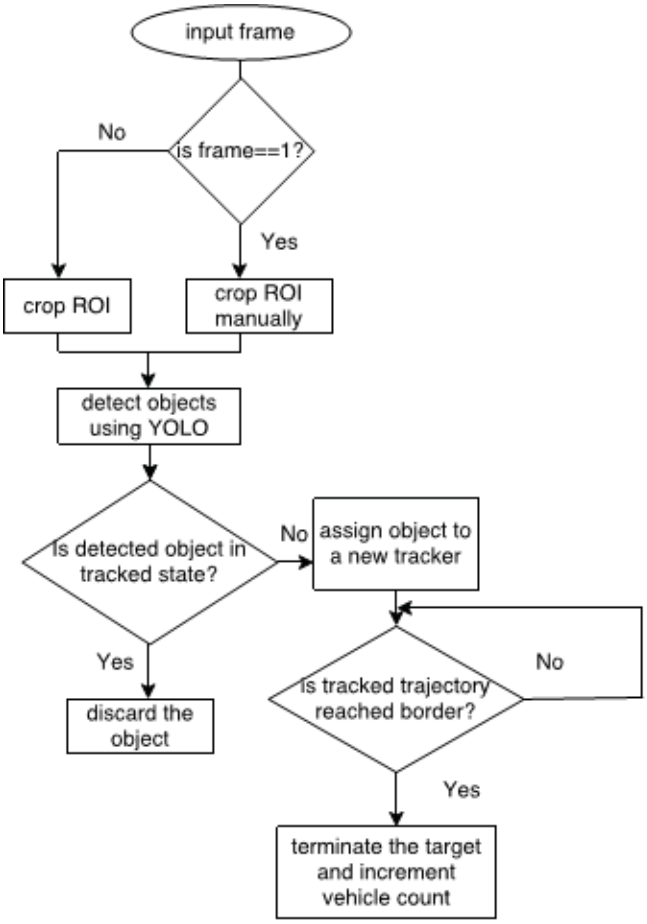

Figure 2: Flowchart of the proposed vehicle counting process.

IV. EXPERIMENTAL ANALYSIS

A. Experimental setup

The proposed algorithm is implemented using OPENCV 3.2 and PYTHON in a machine with Intel(R) Core i5-5200U, CPU of 2.20GHz processor with 8GB RAM. The parameters of the YOLO [15] and CF tracker [16] are selected as provided by the respective papers.

B. Dataset preparation

There is a shortage of standard road video dataset for counting vehicles. Therefore, we prepared the video datasets using the mobile device with 13MP camera. Hence, to test the efficiency of the proposed method, the videos are acquired from the over-bridge in a highway facing downwards with different illumination conditions and shadow effects. The videos are not stable, since they have been captured by hand (not fixed). All videos have 1920 x 1080 resolution in RGB .mp4 format. Additionally, the videos contain complex background with moving plants and crossing pedestrians, surrounding buildings, trees, birds etc. To initiate the counting process, a small road section is manually cropped in the first frame. A simple background subtraction algorithm often fails to extract the moving vehicles accurately. In addition, shadows and illumination variation degrade the performance of the background subtraction algorithm. Therefore, we employed a robust object detector, YOLO to detect and classify the moving vehicles in the entry window. Thus, the cropped region acts as an entry window where the tracking trajectory is initiated. In contrast, the border of the frame acts as an exit line for all vehicles where tracking trajectory is terminated by incrementing the count. The processing time of YOLO for image shown in Figure 5 using the above mentioned CPU based machine is 1.5 sec. However, real time speed is reported in GPU based machines. Also, that of correlation filter based tracker is 0.013 sec for 2 objects. In spite of high detection rate, YOLO takes high processing time. Hence, to reduce the time, we run YOLO after every M frames. (in this work M=5, depends on traffic density)

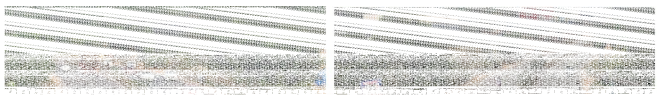

Figure 3: Sample locations of video used for vehicle counting. The videos are acquired using the handheld mobile cameras taken from the over-bridge. Four different locations are chosen to test the accuracy of the proposed method

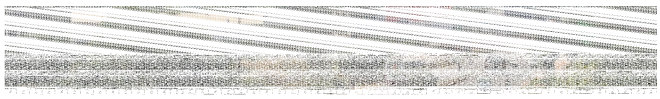

Figure 4: Manually selected region of interest (ROI)

Figure 5: Cropped ROI

Figure 6: Object detection using YOLO framework on entry window (ROI)

C. Quantitative and qualitative analysis

The experimental results consist of comparing vehicle count using the proposed method with the manual count. To measure quantitatively, precision and recall evaluation metrics used for object detection are considered [19]. The precision (P) and recall (R) are defined as follows:

$$P = \frac{\text{No of correctly detected bounding box}}{\text{No of ground-truth bounding box}} \quad [7]$$

$$R = \frac{\text{No of correctly detected bounding box}}{\text{No of detected bounding box}} \quad [8]$$

Thus, recall gives the information about how many of detected bounding boxes are correct while precision tells about false alarms. High value, close to 1 is expected for ideal systems. The vehicle is said to be detected if the overlap of detected bounding box and the ground-truth bounding box is greater than 0.5. F-score measures harmonic mean of precision and recall as

$$F = \frac{2PR}{P+R} \quad [9]$$

Similarly, counting accuracy is computed as

$$\text{Accuracy} = \frac{\text{No of correct detections}}{\text{No of ground-truth detections}} \quad [10]$$

The detail of videos, precision, recall, F-measure, ground-truth count and vehicle count obtained using the proposed method are given in Table 1.

To test the efficiency of the proposed method, 7 videos have been considered with low, medium and high traffic conditions. From Table 1, it is clear that the proposed method achieves greater accuracy. Thus, mean 95.9 % accuracy is achieved. Figure 7. shows the sample 575th frame of video 2.mp4. The vehicles currently in tracking state are displayed using red bounding box.

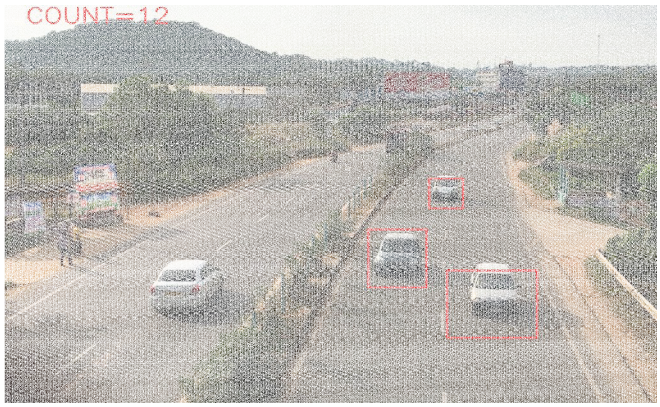

Figure 7: 575th frame of video 2.mp4. Total no of vehicles passed are displayed at the top corner. Vehicles under tracked state are shown in red bounding box.

Table 1: Results of vehicle counting by the proposed method using the road videos

Video Sequence	No of Frames	Total No. of Vehicles in the ground-truth	Total No of Vehicles detected using the proposed method	Missing/Multiple detection/error	Precision %	Recall %	F-score	Counting accuracy %
1.mp4	899	10	10	0/0/0	100	100	**100**	**100%**
2.mp4	715	16	15	1/0/1	93.7	100	**96.7**	**93.7%**
3.mp4	845	17	18	0/1/1	100	94.4	**97.1**	**94.4%**
4.mp4	3598	58	55	3/0/3	94.8	100	**97.3**	**94.8%**
5.mp4	2100	25	23	2/0/2	92	100	**95.8**	**92.0%**
6.mp4	5528	67	67	1/1/2	98.5	98.5	**98.5**	**97.0%**
7.mp4	2999	28	28	0/0/0	100	100	**100**	**100%**

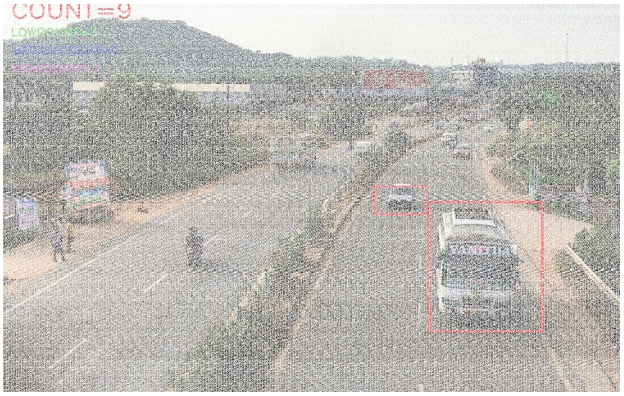

Figure 8: 860th frame of video 1.mp4. Total no of vehicles passed are displayed at the top corner. Traffic count is also classified as small (LOWCOUNT), medium (MEDIUMCOUNT) and large (HIGHCOUNT) categories. Vehicles in tracked state are shown in red bounding box.

V. CONCLUSION

In this paper, the detection and counting of vehicles in a mixed traffic condition is proposed. We exploited YOLO framework to detect the vehicles and correlation filters to track precisely. The traffic density of selected videos varies from low to high and the proposed method counts the vehicles accurately. The advantage of the proposed method is that it can be generalized to any kind of road video captured using handheld mobile camera. Moreover, YOLO can also distinguish the vehicle classes, counting can also be accomplished for different classes to analyze the count of each vehicle type in a traffic video. The limitations of the proposed work are as follows: Although the tracking of multiple vehicles can be performed in real time, the processing time required for the object detection is high. In this work, single lane is considered for counting the vehicles, which will be extended to 2 lanes in the future. In addition, an automatic ROI extraction through lane detection will be considered in the future work.

REFERENCES

[1] Pornpanomchai, C., Liamsanguan, T., & Vannakosit, V., Vehicle detection and counting from a video frame. In Wavelet Analysis and Pattern Recognition, ICWAPR'08. International Conference on, vol. 1, pp. 356-361, 2008

[2] Xia, Y., Shi, X., Song, G., Geng, Q., & Liu, Y, Towards improving quality of video-based vehicle counting method for traffic flow estimation. Signal Processing, vol. 120, pp. 672-681, 2016.

[3] Barcellos, P., Bouvié, C., Escouto, F. L., & Scharcanski, J., A novel video based system for detecting and counting vehicles at user-defined virtual loops. Expert Systems with Applications, vol. 42 no. 4, pp. 1845-1856, 2015.

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

[5] Salvi, G., An automated nighttime vehicle counting and detection system for traffic surveillance. In Computational Science and Computational Intelligence (CSCI), 2014 International Conference on , vol. 1, pp. 131-136, 2014.

[6] Bhaskar, P. K., & Yong, S. P., Image processing based vehicle detection and tracking method. In Computer and Information Sciences (ICCOINS), 2014 International Conference on, pp. 1-5, 2014.

[7] Chen, T. H., Lin, Y. F., & Chen, T. Y., Intelligent vehicle counting method based on blob analysis in traffic surveillance. In Innovative Computing, Information and Control, 2007. ICICIC'07. Second International Conference on, pp. 238-238, 2007.

[8] Quesada, J., & Rodriguez, P., Automatic vehicle counting method based on principal component pursuit background modeling. In Image Processing (ICIP), 2016 IEEE International Conference on , pp. 3822-3826, 2016.

[9] Jang, H., Won, I. S., & Jeong, D. S., Automatic Vehicle Detection and Counting Algorithm. International Journal of Computer Science and Network Security (IJCSNS), vol. 14, no. 9, pp. 99, 2014.

[10] Moranduzzo, T., & Melgani, F., Automatic car counting method for unmanned aerial vehicle images. IEEE Transactions on Geoscience and Remote Sensing, vol. 52, no. 3, pp. 1635-1647, 2014.

[11] Viola, P., & Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. In Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on (Vol. 1, pp. I-I). IEEE.

[12] Dalal, N., & Triggs, B. (2005, June). Histograms of oriented gradients for human detection. In Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on (Vol. 1, pp. 886-893). IEEE.

[13] Felzenszwalb, P. F., Girshick, R. B., & McAllester, D., Cascade object detection with deformable part models. In Computer vision and pattern recognition (CVPR), 2010 IEEE conference on, pp. 2241-2248, 2010.

[14] Ren, S., He, K., Girshick, R., & Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In Advances in neural information processing systems (pp. 91-99).

[15] Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (pp. 779-788).

[16] Danelljan, M., Häger, G., Khan, F., & Felsberg, M. (2014). Accurate scale estimation for robust visual tracking. In British Machine Vision Conference, Nottingham, September 1-5, 2014. BMVA Press.

[17] Yang, Y., & Bilodeau, G. A. (2016). Multiple Object Tracking with Kernelized Correlation Filters in Urban Mixed Traffic. arXiv preprint arXiv:1611.02364.

[18] Van Pham, H., & Lee, B. R. , Front-view car detection and counting with occlusion in dense traffic flow. International Journal of Control, Automation and Systems, vol. 13, no. 5, pp. 1150-1160, 2015.

[19] Wolf, C., & Jolion, J. M., Object count/area graphs for the evaluation of object detection and segmentation algorithms. International Journal of Document Analysis and Recognition (IJDAR), vol. 8, no. 4, pp. 280-296, 2006.