Keywords: vehicle tracking; speed estimation; mixed traffic conditions; YOLO; SORT

Xuan Can VUONG[1]*, Rui-Fang MOU[2], Trong Thuat VU[3]

VEHICLE TRACKING AND SPEED ESTIMATION UNDER MIXED TRAFFIC CONDITIONS USING YOLOV4 AND SORT: A CASE STUDY OF HANOI

Summary. This paper presents a method to estimate vehicle speed automatically, including cars and motorcycles under mixed traffic conditions from video sequences acquired with stationary cameras in Hanoi City of Vietnam. The motion of the vehicle is detected and tracked along the frames of the video sequences using YOLOv4 and SORT algorithms with a custom dataset. In the method, the distance traveled by the vehicle is the length of virtual point-detectors, and the travel time of the vehicle is calculated using the movement of the centroid over the entrance and exit of virtual point-detectors (i.e., region of interest), and then the speed is also estimated based on the traveled distance and the travel time. The results of two experimental studies showed that the proposed method had small values of MAPE (within 3%), proving that the proposed method is reliable and accurate for application in real-world mixed traffic environments like Hanoi, Vietnam.

1. INTRODUCTION

Vehicle speed is a fundamental parameter of traffic flow in both macroscopic and microscopic traffic analyses. Vehicle speed has defined as the rate of movement of vehicles in distance per unit of time and is indirectly or directly related to the traffic state and road conditions. Speed measurement will help policymakers to estimate assessments of the traffic state or other anomalous events, as well as to implement solutions to deal with the over-speeding behavior of drivers. Besides, it can be combined with vehicle counting to assist in traffic planning and signal identification, as well as pollutant emission assessment.

Vehicle speed can be measured automatically from various sources, such as magnetic inductive loop detectors, radar detectors, infrared detectors, video-based systems, and so on. In comparison to other sources, video-based systems have a variety of benefits, including portability, affordability, ease of installation, and operation [1]. Regarding the estimation approaches, the estimation of vehicle speed from the video-based systems can be conducted with traditional approaches and deep convolutional neural networks (CNNs). Traditional approaches commonly used include background subtraction [2], frame differencing [3], and optical flow [4, 5]. Background subtraction works pretty well in real-time, it is quite sensitive to changes in lighting, weather, and other environmental factors. The frame differencing approach has a good ability to adapt to changes in light, but it is unable to obtain all of the object's boundary information. Optical flow may detect moving objects without any prior scene knowledge, but it is inapplicable for real-time detection due to its intricate and time-consuming

[1] University of Transport and Communications, Faculty of Environment and Transport Safety; Hanoi, Vietnam, 100000; e-mail: vuongcan@utc.edu.vn; orcid.org/0000-0002-1174-1416
[2] Southwest Jiaotong University, School of Transportation and Logistics; Chengdu, China, 611756; e-mail: mrfswjtu@126.com; orcid.org/0000-0002-9191-4355
[3] University of Transport and Communications, Faculty of Electrical-Electronic Engineering; Hanoi, Vietnam, 100000; e-mail: vutrongthuat@utc.edu.vn; orcid.org/0000-0002-2363-3519
* Corresponding author. E-mail: vuongcan@utc.edu.vn

calculations. Generally, traditional approaches are sensitive to noise, so they are difficult to improve accuracy in complex traffic conditions.

Because of their accuracy, CNNs are commonly used to overcome the shortcomings of the abovementioned traditional approaches in intelligent traffic analytics, such as vehicle detection, tracking, and speed estimation. Typical CNNs include region-based CNNs (R-CNN [6], Fast R-CNN [7], Faster R-CNN [8]), You Only Look Once (YOLO) [9], and Single Shot MultiBox Detector (SSD) [10]. For example, Kumar et al. [11] applied Mask-RCNN [12] and SORT (Simple Online and Real-time Tracking) [13] to estimate vehicle speed and achieved a root mean square error (RMSE) of 9.54 (mph) for estimating vehicle speed in movies of unrestricted traffic. Biswas et al. [14] applied a system of vehicle speed detection based on a Faster R-CNN [8], the feature-based image alignment, the channel and spatial reliability tracking, and the structural similarity index measurement. Their system achieved a speed accuracy of 96.80%. However, the representations in R-CNN and its associated approaches take a long time to run completely on an image. To overcome that restriction, SSD and YOLO predict objects in images using local information. Hence, they accurately identify objects in real-time image processing while maintaining a high mAP (mean Average Precision) [15]. Besides, YOLO has several advancements (e.g., batch normalization, dimension clusters, and anchor boxes), leading to high mAP in the MS. COCO dataset [16]. Therefore, YOLO has received a lot of attention recently. For instance, Liu et al. [17] used YOLOv2 and FlowNet to estimate the real-time speed of an object. Liu and Juang [18] used YOLOv4 [19] and DeepSORT [20] for the detection and tracking of vehicles on the freeway. Gauttam and Mohapatra [21] used YOLO to estimate the speed of an approaching vehicle when this vehicle is to cross the ROI (region of interest) using a moving camera. Their system achieved an average accuracy of 90%.

YOLO has been employed successfully for vehicle detection in homogeneous traffic with 4-wheelers vehicles used predominantly [17, 18, 21-25]. However, most of the research does not focus on vehicle detection and tracking in mixed traffic flow like in Vietnam where on-road motorcycles are used frequently. For example in Hanoi City, the capital of Vietnam, motorcycle traffic accounts for 86% of traveling in the urban area [26, 27]. In contrast to homogeneous traffic like that found in developed countries, traffic flow is quite complicated due to the non-lane-based movements of vehicles on roads. Hence, the detection, tracking, and speed estimation of vehicles in mixed traffic flow like Vietnam remain a major challenge.

In this study, we provide a method to estimate the speed of the vehicles, including cars and motorcycles under mixed traffic flow using a sequence of video images from an un-calibrated camera. The proposed method has used the state-of-the-art object detector and tracking, YOLOv4 [19] and SORT [13] with a custom dataset for vehicle detection, classification, tracking, and speed estimation. In addition, the average speed of mixed traffic flow is also estimated based on an individual vehicle and vehicle type. The following are the primary.

- Proposing a method for speed estimation under mixed traffic flow.
- Designing a custom dataset for YOLOv4 and SORT to track and estimate vehicle speed.
- Experimental studies show that the proposed method is practical and easy to use.

The remainder of this paper is organized as follows. The proposed method employing YOLOv4 and SORT is described in Section 2. The experimental study to estimate vehicle speed in Hanoi City, Vietnam is described in Section 3. The conclusion and future work are provided in Section 4.

2. PROPOSED METHODS

2.1. Workflow of proposed methods

The proposed method of vehicle tracking and speed estimation includes main three components: (1) vehicle detection and classification (2) vehicle tracking, and (3) speed estimation. In the following sections, we provide a detailed explanation of each component. A flowchart of the proposed method is shown in Fig. 1.

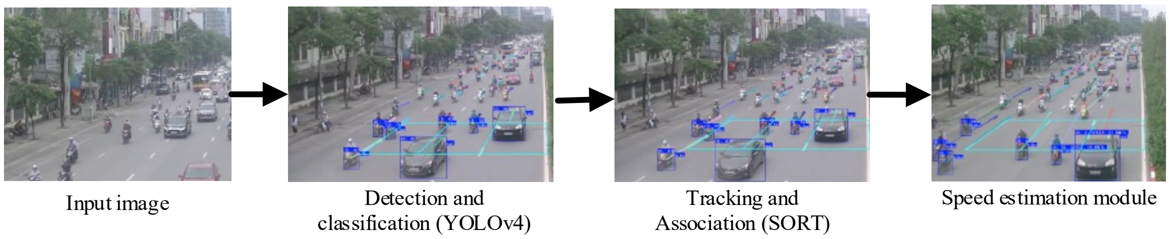


Fig. 1. Flowchart of the proposed method for vehicle speed estimation from traffic videos

2.2. Vehicle detection and classification

To detect and classify vehicles in traffic videos, we apply YOLOv4 [19] with a custom dataset. To implement real-time vehicle detection, YOLOv4 offers improved accuracy and quicker results in comparison with the abovementioned CNNs. The architecture of the vehicle detector using YOLov4 consists of four components as follows: input image, backbone network, head, and neck. The backbone network in YOLOv4 is the pre-trained network from the input images and extracts features based on the CSPdarknet-53 architecture [28]. The head of the YOLOv4 consists of Spatial Pyramid Pooling (SPP) [29], Path Aggregation Network (PAN) [30], and YOLOv3 [31] and it is utilized to predict the classes and bounding boxes of the vehicles, and enhance the training speed and the model accuracy. The neck is to increase strength by collecting feature maps from the intermediate stages. Besides, the Greedy Non-maximum Suppression (NMS) [32] is utilized to ensure that there is no overlap in the proposed area.

In YOLOv4, the input image is split into a grid S×S, with each grid cell in charge of vehicle detection. We take B bounding boxes and place one in each grid cell. The network then outputs an offset value for the bounding box and class probability. The bounding boxes are chosen, and the vehicle in the image is then located using those that have a class probability with a particular threshold (see Fig. 2).

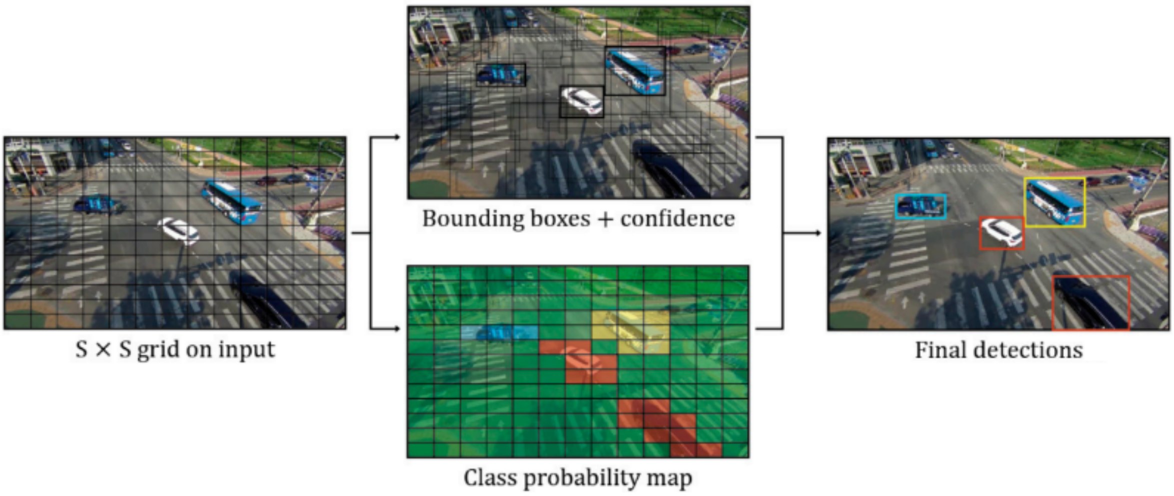


Fig. 2. Illustration of YOLOv4 framework [25]

YOLOv4 trained on the MS.COCO dataset [16] provides a fair inference speed and relatively high performance. Bicycle (class 2), motorcycle (class 4), car (class 3), bus (class 6), and truck (class 8) in the MS.COCO dataset are road transport modes. In a practical application with a complex traffic context like Vietnam, YOLO is capable of identifying and classifying large and medium-sized vehicles (i.e., cars, buses, trucks), but is not stable when detecting small-sized vehicles (bicycles and motorcycles) due to a lack of their training data. To overcome that drawback, we have built a real training dataset of vehicles that is taken from real traffic situations in Vietnam. Cameras are used to collect most of the images for the training dataset placed in front of or behind vehicles and are collected from different real-life scenes on different traffic routes. Some of the images are also taken from videos available on YouTube, Google map's street view, and other sources. We have considered two main types of vehicle categories for vehicle detection, including motorcycles and cars, which account for 95% of on-road vehicles. Because other vehicles (bicycles, buses, trucks, etc.) account for a very small percentage of traffic flow in urban areas, they should not be considered in this study.

Our dataset includes 2543 images of motorcycles, cars, and other vehicles (bicycles, buses, trucks, etc.). The vehicle dataset is divided into a training set and a validation set with a ratio of 7:3 (see Tab. 1 and Fig. 3).

Table 1

Vehicle information about our dataset

	Objects		
	Total	Training	Validation
Motorcycle (motorbike)	11446	8439	3007
Car	4508	3350	1158
Others (bus, truck, bicycle, etc.)	3382	2563	819

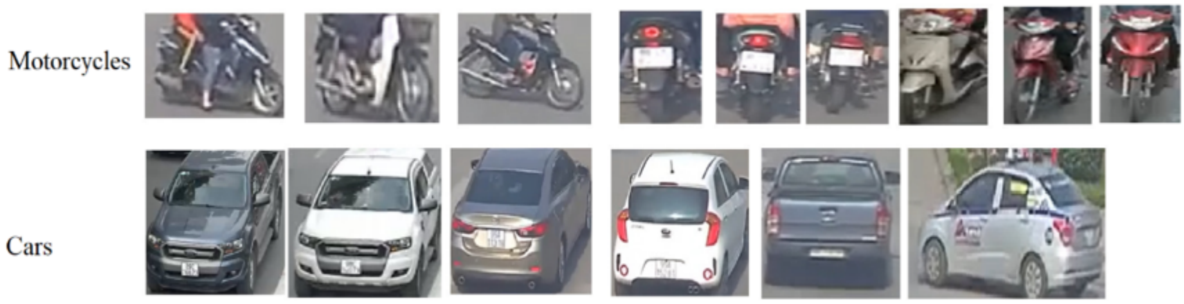


Fig. 3. Cars and motorcycles in our dataset

The process of vehicle labeling in each image is done with the help of the LabelImg tool developed by Tzutalin [33], and the diagram of the training process YOLOv4 on our dataset is shown in Fig. 4. The YOLOv4 layers are trained via "yolov4-custom.cfg" file to optimize configurations. Besides, the "yolov4. conv.137" file as a pre-trained network is used for the training model of YOLOv4. The process of training and validation is done on Google Colab [34], equipped with a Tesla T4 GPU. The result of the average loss of YOLOv4 with 33000 iterations equates to 1.1941. The overall mAP is 96.55%. The values of average precision (AP) of cars and motorcycles are 90.34% and 90.16%, respectively. These values are quite high, reflecting the capacity of our dataset in vehicle detection in the Vietnam transport context.

2.3. Vehicle tracking

In this section, we briefly describe the Simple Online and Real-time Tracking (SORT) [13] algorithm to track an individual vehicle. SORT is a real-time and online tracking algorithm integrated with the Kalman filter and the Hungarian algorithm, which has an accuracy comparable to the state-of-the-art online trackers while supporting higher update rates. The state space of each vehicle is modeled as follows:

$$[x, y, s, r, \dot{x}, \dot{y}, \dot{s}]^T \tag{1}$$

Where (x , y) is the center location of the bounding box of the vehicle; s and r represent the scale (area) and the aspect ratio of the bounding box, respectively; $\dot{x}$, $\dot{y}$, and $\dot{s}$ are the velocity elements. When detection is associated with a vehicle, the detected bounding box is used to update the state of the vehicle. The location of the new bounding box for each vehicle is predicted based on the Kalman filter.

Regarding data association, SORT uses the intersection-over-union (IOU) between each incoming detection and all predicted bounding boxes of the existing vehicle. These values are used to populate a cost matrix, which is then solved by a Hungarian algorithm to assign each track an appropriate detection. In addition, a minimum value of IOU (IOU_{min}) is applied to reject assignments when the overlap between the detection and the vehicle is less than the IOU_{min}. The vehicle tracking procedure of SORT is shown in Fig. 5.

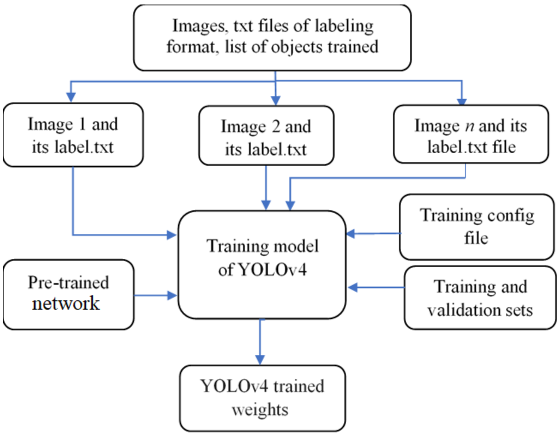


Fig. 4. Diagram of training YOLOv4 on our dataset

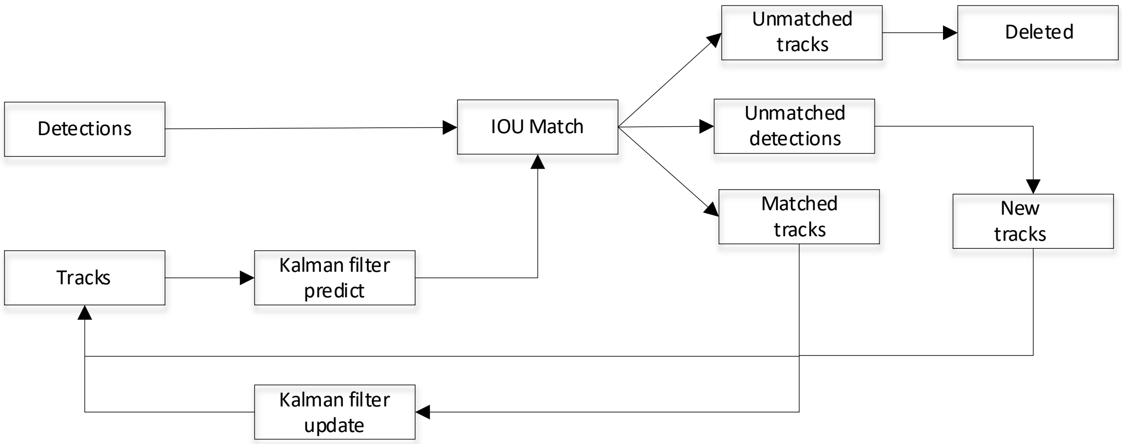


Fig. 5. Vehicle tracking procedure of SORT [13, 18]

2.4. Speed estimation

For each detected vehicle, the bounding box is given to identify and determine the centroid point of the vehicle, which is the most important thing for tracking and its speed estimation. While the vehicle is in ROI (region of interest), the vehicle will be tracked and calculated the traveled distance using Euclidean distance. After that, we can estimate vehicle speed between some consecutive frames by its centroid point. To simplify the calculation, in this paper, ROI as a virtual point-detector, vehicle speed in ROI is a constant, and the traveled distance by the vehicle is equal to the length of the virtual detector, i.e., the size of the ROI along the length of the road segment, as shown in Fig. 6.

The speed of the vehicle between the two scanlines (i.e., the entrance and exit of ROI) can be calculated by a simple equation as follows.

$$v = \frac{d}{t} \times 3.6 \quad , \tag{2}$$

where v is the speed of the vehicle that traveled in ROI (km/h); d is the traveled distance of the vehicle in ROI (m). In this study, it is the length of the virtual detector, i.e., the length of the ROI (see Fig. 1), and we take $d = 5$ meters; t is the travel time (in seconds) for the vehicle to run the traveled distance, and it can be calculated as:

$$t = (f_2 - f_1) \times (1 / fps) \quad , \tag{3}$$

where f_1, f_2 are the frame indexes when a vehicle goes across scanline 1 (the entrance of the virtual point-detector) and scanline 2 (the exit of the virtual point-detector), respectively; fps is the frame rate of the input video (i.e., number of frames per second).

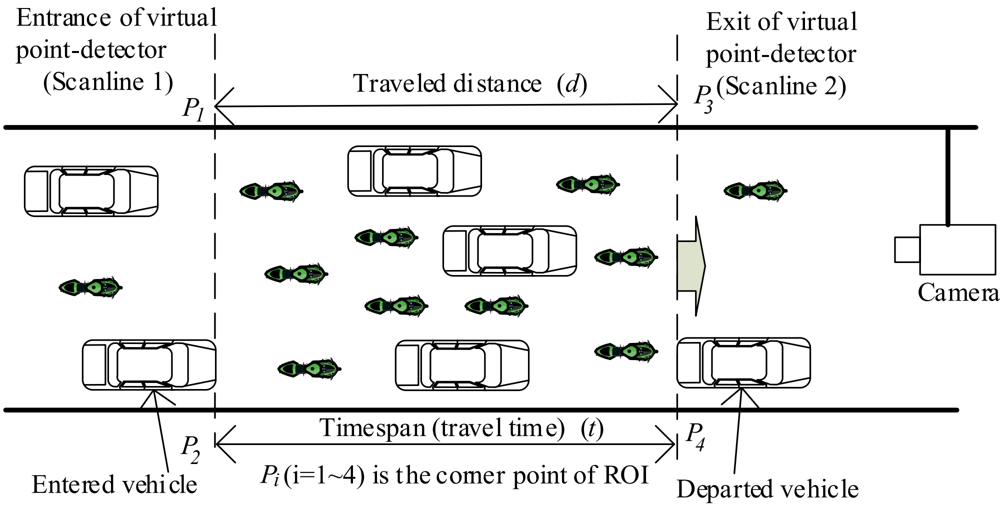


Fig. 6. ROI setup and speed estimation module

Due to large variations in the speed of different types of vehicles, including motorcycles, cars, trucks, buses, and so on, we can determine the weighted average speed as the speed of mixed traffic flow in the observation period using the given equation [35]:

$$v_t = \frac{\sum_{i=1}^{n} q_i v_i}{\sum_{i=1}^{n} q_i} , \tag{4}$$

where n is the total number of vehicle types, which present in mixed traffic flow; v_i is the average speed of the i-th vehicle type that is calculated based on Equation (2) and the number of vehicles of the i-th type; v_t is the average speed of mixed traffic flow; q_i is the number of vehicles of the i-th type.

In this paper, the performances of speed estimation of the vehicles (motorcycles and cars) are calculated using the mean absolute percentage error (*MAPE*) and the root mean square error (*RMSE*), which are determined as follows.

$$MAPE = \frac{1}{m}\sum_{j=1}^{m}\left|\frac{v_j - \hat{v}_j}{v_j}\right| , \tag{5}$$

$$RMSE = \sqrt{\frac{1}{m}\sum_{j=1}^{m}(v_j - \hat{v}_j)^2} , \tag{6}$$

where $\hat{v}_j$ is the estimated value; v_j the measurement value; m is the number of estimated vehicles.

3. EXPERIMENTAL ANALYSIS AND RESULTS

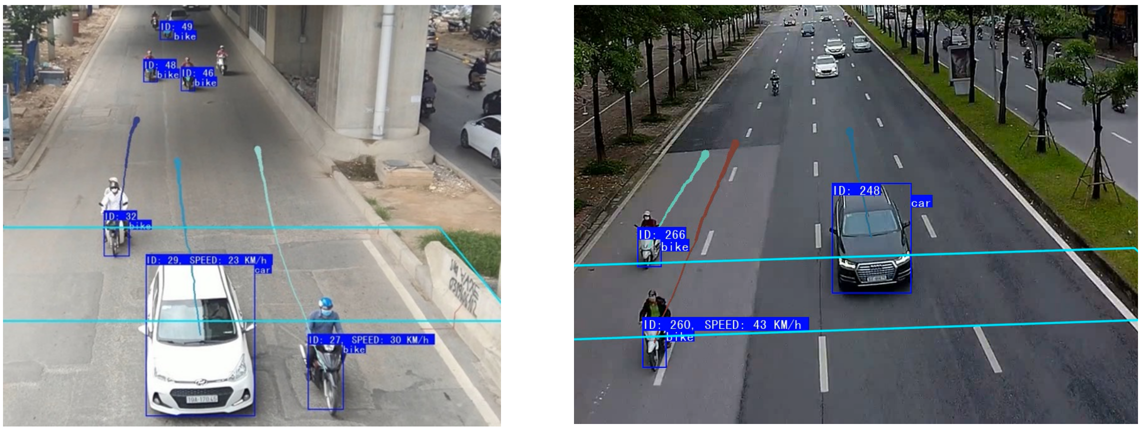
The experimental data of traffic videos are to detect the vehicle's motion and estimate its speed. To evaluate the proposed method, we have carried out two experimental studies corresponding to two locations in Hanoi City of Vietnam, where they have certain differences in terms of geometry and traffic conditions as well as average speed. The experimental data have been acquired by stationary cameras located at pedestrian bridges on Xuan Thuy road (Coordinates: longitude = 21.036687, latitude = 105.781923, see Fig. 7a) and on Me Tri road (Coordinates: longitude =21.013213, latitude =105.776168, see Fig. 7b). The video data were 1920x1080 resolution with 29 frames per second. The first experiment was a 1-minute video clip, and the other was a 3-minute video clip.

(a) Pedestrian bridge on Xuan Thuy road

b) Pedestrian bridge on Me Tri road

Fig. 7. Experimental locations (taken by google maps)

The experimental studies were established using PYTHON and were run on the Google Colab environment [34]. Vehicle counting calculated with our method and manual method was the number of vehicles passing the ROI (see Fig. 8). The results of vehicle counting were shown in Tab. 2.



(a) The first experiment on Xuan Thuy road

(b) The second experiment on Me Tri road

Fig. 8. Cars and motorcycles detected and tracked in the experimental studies

Table 2

Results of vehicle detection and counting

Location	Vehicle type	Manual count	Our method	Accuracy (%)
Xuan Thuy road	Motorcycle	34	34	100
	Car	11	11	100
Me Tri road	Motorcycle	86	83	96.5
	Car	36	35	97.2
Total		167	163	97.6

Tab. 2 showed that there were 45 vehicles (motorcycles and cars) extracted correctly, and no vehicles were missed detections in the first experiment on Xuan Thuy road. In the second experiment on Me Tri road, there was missed vehicle counting. However, the error was quite low (within 5%). The average accuracy achieved in the experimental studies was 97.6%. This was quite a high value and it allowed us to estimate the speed of the vehicles (motorcycles and cars). The results of speed estimation based on our method and manual method were described in Tab. 3.

Table 3

Speed estimation results

Location	Number of vehicles estimated	Average speed from manual observed (km/h)	Average speed from our method (km/h)	*MAPE* (%)	*RMSE* (km/h)
Xuan Thuy road	41	26.122	26.385	2.90	0.85
Me Tri road	112	45.785	46.509	2.95	1.71

Tab. 3 showed that the number of vehicles estimated in terms of vehicle speed was lower than the number of vehicles counted in experimental studies, accounting for about 91% of the total number of vehicles counted. This was because vehicle matching and tracking errors occurred due to many noises influencing the order of the vehicle exist. When determining the average speed of the vehicle flow, it is not necessary to determine the speed of the whole number of vehicles through an observation cross-section, but only a sufficiently large sample size is acceptable. Therefore, this value is consistent with reality. Besides, the results found that *MAPE* and *RMSE* of the proposed method were quite small, within 3% and 2.0 km/h, respectively. This result can allow us to use the proposed method for other related studies (e.g., traffic status analysis, traffic jams, and so on) in future works.

4. CONCLUSIONS

In this paper, we have presented a computationally efficient vehicle detection and speed estimation method for traffic videos under mixed traffic flow in Hanoi City of Vietnam. The proposed method was based on the YOLOv4 algorithm in combination with the SORT algorithm with a custom dataset. The speed estimation results in experimental studies showed that the accuracy of the proposed method was quite high with MAPE within 3%. However, the proposed method still has some limitations that need to be overcome, such as limitations on the type of vehicles considered, limitations on traffic and weather conditions, and other aspects. In the next study, we will increase the number of images for the training dataset to increase the type of vehicles, such as buses, trucks, etc. At the same time, to improve the algorithms to increase the accuracy of the proposed model in different traffic conditions in Vietnam. Besides, smartphone applications and real-time processing will also be of interest.

Acknowledgment

This research is supported by the Haiphong Department of Science and Technology, Vietnam, under the project:" Research on the implementation of the traffic signal control system at intersections in Haiphong City". We are very grateful to our students (Vu Van Duy and Nguyen Duy Noi) at the University of Transport and Communications for their great support of this research. The authors also thank two anonymous reviewers for their very helpful comments.

References

1. Wang, J.M. & Chung, Y-C & Lin, S. & Chang, S-L. & Cherng, S. & Chen, S-W. (eds.). Vision-based traffic measurement system. In: *Proceedings of the 17th International Conference on Pattern Recognition. 2004 ICPR 2004*. 2004: IEEE.
2. Ibrahim, O. & ElGendy, H. & ElShafee, A.M. Speed Detection Camera System using Image ProcessingTechniques on Video Streams. *International journal of computer and electrical engineering*. 2011. Vol. 3. No. 6. P. 771-778.

3. Wang, J-X (ed.). Research of vehicle speed detection algorithm in video surveillance. In: *2016 International Conference on Audio, Language and Image Processing (ICALIP).* 2016: IEEE.
4. Lan, J. & Li, J. & Hu, G. & Ran, B. & Wang, L. Vehicle speed measurement based on gray constraint optical flow algorithm. *Optik*. 2014. Vol. 125. No. 1. P. 289-295.
5. Qimin, X. & Xu, L. & Mingming, W. & Bin, L. & Xianghui, S & editors. A methodology of vehicle speed estimation based on optical flow. In: *Proceedings of 2014 IEEE International Conference on Service Operations and Logistics, and Informatics.* 2014: IEEE.
6. Girshick, R. & Donahue, J. & Darrell, T. & Malik, J. (eds.). Rich feature hierarchies for accurate object detection and semantic segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014.
7. Girshick, R. (ed.). Fast r-cnn. In: *Proceedings of the IEEE international conference on computer vision.* 2015.
8. Ren, S. & He, K. & Girshick, R. & Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*. 2015. Vol. 28. P. 1-9.
9. Redmon, J. & Divvala, S. & Girshick, R. & Farhadi, A. (eds.). You only look once: Unified, real-time object detection. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
10. Liu, W. & Anguelov, D. & Erhan, D. & Szegedy, C. & Reed, S. & Fu, C-Y & et al. (eds.). Ssd: Single shot multibox detector. In: *European conference on computer vision.* 2016: Springer.
11. Kumar, A. & Khorramshahi, P. & Lin, W-A. & Dhar, P. & Chen, J-C. & Chellappa, R. (eds.). A semi-automatic 2D solution for vehicle speed estimation from monocular videos. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops.* 2018.
12. He, K. & Gkioxari, G. & Dollár, P. & Girshick, R. (eds.). Mask r-cnn. In: *Proceedings of the IEEE international conference on computer vision.* 2017.
13. Bewley, A. & Ge, Z. & Ott, L. & Ramos, F. & Upcroft, B. (eds.). Simple online and realtime tracking. In: *2016 IEEE international conference on image processing (ICIP).* 2016: IEEE.
14. Biswas, D. & Su, H. & Wang, C. & Stevanovic, A. Speed estimation of multiple moving objects from a moving UAV platform. *ISPRS International Journal of Geo-Information*. 2019. Vol. 8. No. 6. No. 259. P. 1-15.
15. Pham, P. & Nguyen, D. & Do, T. & Ngo, T.D. & Le, D-D (eds.). Evaluation of deep models for real-time small object detection. In: *International conference on neural information processing.* 2017: Springer.
16. Lin, T-Y & Maire, M. & Belongie, S. & Hays, J. & Perona, P. & Ramanan, D. & et al. (eds.). Microsoft coco: Common objects in context. In: *European conference on computer vision.* 2014: Springer.
17. Liu, K. & Ye, Y. & Li, X. & Li, Y. (eds.). A Real-Time Method to Estimate Speed of Object Based on Object Detection and Optical Flow Calculation. In: *Journal of Physics: Conference Series*; 2018: IOP Publishing.
18. Liu, C-M. & Juang, J-C. Estimation of lane-level traffic flow using a deep learning technique. *Applied Sciences*. 2021. Vol. 11. No. 12. No. 5619.
19. Bochkovskiy, A. & Wang, C-Y & Liao, H-YM. Yolov 4: Optimal speed and accuracy of object detection. *arXiv: Computer Vision and Pattern Recognition*. 2020. P. 1-17.
20. Wojke, N. & Bewley, A. & Paulus, D. (eds.). Simple online and realtime tracking with a deep association metric. In: *2017 IEEE international conference on image processing (ICIP)*. 2017: IEEE.
21. Gauttam, H.K. & Mohapatra, R.K. (eds.). Speed prediction of fast approaching vehicle using moving camera. In: *International Conference on Computer Vision and Image Processing.* 2019: Springer.
22. Li, X. & Liu, Y. & Zhao, Z. & Zhang, Y. & He, L. A deep learning approach of vehicle multitarget detection from traffic video. *Journal of Advanced Transportation*. 2018. Article ID 7075814.
23. Sang, J. & Wu, Z. & Guo, P. & Hu, H. & Xiang, H. & Zhang, Q. & et al. An improved YOLOv 2 for vehicle detection. *Sensors*. 2018. Vol. 18. No. 12. Article No. 4272.

24. Ćorović, A. & Ilić, V. & Đurić, S. & Marijan, M. & Pavković, B. (eds.). The real-time detection of traffic participants using YOLO algorithm. In: *2018 26th Telecommunications Forum (TELFOR)*; 2018: IEEE.
25. Seong, S. & Song, J. & Yoon, D. & Kim, J. & Choi, J. Determination of vehicle trajectory through optimization of vehicle bounding boxes using a convolutional neural network. *Sensors*. 2019. Vol. 19. No. 19. Article No. 4263.
26. Nguyen, H.V. & Vuong, X.C. & Vu, T.T. A Fuzzy Traffic Signal Control Method for a Single Intersection Under Mixed Traffic Conditions. *Technology*. 2020. Vol. 11. No. 10. P. 1715-1723.
27. Vuong, X.C. & Phan, X.V. & Mou, R.F. & Vu, T.T. & Vu, V.D. & Nguyen, D.N. Vehicle Detection and Counting Under Mixed Traffic Conditions in Vietnam Using Yolov 4. *International Journal Advanced Research Engineering a Technology*. 2021. Vol. 12. No. 2. P. 722-730.
28. Wang, C-Y & Liao, H-YM & Wu, Y-H & Chen, P-Y & Hsieh, J-W & Yeh, I-H (eds.). CSPNet: A new backbone that can enhance learning capability of CNN. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops.* 2020.
29. He, K. & Zhang, X. & Ren, S. & Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*. 2015. Vol. 37. No. 9. P. 1904-1916.
30. Liu, S. & Qi, L. & Qin, H. & Shi, J. & Jia, J. (eds.). Path aggregation network for instance segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2018.
31. Redmon, J. & Farhadi, A. Yolov 3: An incremental improvement. *arXiv.* 2018. No 1804.02767. P. 1-6.
32. Neubeck, A. & Van Gool, L. (eds.). Efficient non-maximum suppression. In: *18th International Conference on Pattern Recognition (ICPR'06).* 2006: IEEE.
33. Tzutalin. labelImg 2020. Available at: https://github.com/tzutalin/labelImg.
34. Google. Welcome To Colaboratory 2022. Available at: https://colab.research.google.com/.
35. Chu, C.M. & Kazushi, S. & Shoji, M. The speed, flow and headway analyses of motorcycle traffic. *Journal of the Eastern Asia Society for Transportation Studies*. 2005. Vol. 6. P. 1496-1508.

Received 11.07.2021; accepted in revised form 20.11.2022