Traffic Flow Detection Based on Improved YOLOX and DeepSORT

Zewen Du [*1], Ying Jin[*2], Hongbin Ma[*3] and Ping Liu[*4]

[*1]School of Automation, Beijing Institute of Technology, Beijing 100081, China
E-mail: dzw1114@163.com
[*2]School of Automation, Beijing Institute of Technology, Beijing 100081, China
E-mail: jinyinghappy@bit.edu.cn
[*3]School of Automation, Beijing Institute of Technology, Beijing 100081, China
E-mail: mathmhb@bit.edu.cn
[*4]School of Automation, Beijing Institute of Technology, Beijing 100081, China
E-mail: lppusr@163.com

Abstract. Aiming at the shortcomings of low detection rate, low tracking accuracy and poor robustness in traditional traffic flow detection algorithms in surveillance video, we introduce a traffic flow detection method, which divides the traffic flow detection task into three modules, including vehicle detection, vehicle tracking and vehicle counting. Firstly, the improved YOLOX model is used to detect the vehicles in a single frame image. Secondly, the DeepSORT algorithm is used to track the detected vehicles. Finally, the two-way traffic flow statistics are completed by the improved virtual coil method. In order to make full use of the feature information of different scales, we use the ASFF module to perform adaptive feature fusion on the output feature maps of different scales. In addition, the convolutional attention modules are also used in the detection algorithm to improve the classification and localization performance of the network. Through experimental tests, the traffic flow detection method proposed in this paper has achieved high accuracy and adaptability in a series of traffic surveillance videos under different weather and different perspectives.

Keywords: Vehicle Detection, Vehicle Tracking, Traffic Flow Detection, YOLOX, DeepSORT

1. INTRODUCTION

Traffic congestion is a common problem in many cities today. A large number of private cars not only cause traffic congestion but also pollute the surrounding environment. In order to ease the congestion of urban traffic and reduce the pollution to the environment, the intelligent transportation system (ITS) emerges with the needs of society. ITS currently plays an active role in urban transportation network which can guide vehicles and traffic management departments in real time, accurately and efficiently, and can solve related traffic problems in smart cities[1].

The ITS integrates sensor technology, information technology and computer technology. It obtains the current road status by collecting signals form sensors, and then uses computer technology to analyze the road condition. As an important part of the ITS, traffic flow detection can assist traffic management departments to efficiently guide vehicles and reduce the traffic congestion[2].

In recent years, due to the rapid development of computer vision and deep learning, vehicle detection and traffic flow statistics based on deep learning have been rapidly applied[3][5]. Vehicle detection based on deep learning has high detection accuracy and excellent environmental adaptability. It can effectively solve the detection problems caused by scale transformation, illumination transformation and occlusion between vehicles. Based on the above content, this paper builds a traffic flow detection system based on deep learning technology, which divides the traffic flow detection task into vehicle detection, vehicle tracking and vehicle counting. And this model can be easily extended to the traffic flow detection of different vehicles, such as motorcycles, vans, trucks, buses and so on.

2. RELATED WORK

2.1. Vehicle Detection

Currently, vehicle detection methods can be divided into traditional methods and deep learning methods. The traditional methods mainly use the hand-designed feature extraction algorithm to extract features from the region of interest (ROI) in the image, then use a classification algorithm to determine whether the ROI has a target object. Among them, the design of the feature extractor has a great influence on the detection accuracy. With the development of image processing algorithms, many feature extraction methods have been proposed, such as SIFT[6], SURF[7], HOG[8], Haar-like[9], etc.

Haar-like features are often used in many vehicle detection studies[10] [11] due to its excellent horizontal and vertical structure detection capabilities and real-time computing speed. There are also some researchers using other image feature extraction algorithms for vehicle detection. For example, reference [12] used SIFT features for vehicle detection. Reference [13][15] used HOG features for vehicle detection. However, because the traditional object detection algorithms rely too much on the experience of the feature designer, and the extracted features also have poor generalization performance, so

the traditional detection algorithms encounter some problems in the development process.

Different from traditional object detection algorithms, convolutional neural networks (CNN) have a deeper network architecture and can learn to extract more complex and abstract features by itself. In 2014, the Region CNN (RCNN) algorithm was first proposed for object detection[16]. Later, many excellent object detection algorithms based on deep learning were proposed, such as Fast RCNN[17], Faster RCNN[18], SSD[19], etc., they have achieved excellent accuracy and performance that is difficult to reach by traditional detection algorithms in different tasks.

For the vehicle detection task, reference [20] used the Faster RCNN method and achieved good results. The R-CNN series performs object detection tasks with high detection accuracy but slow detection speed, and it is difficult to meet real-time requirements. Therefore, in order to balance speed and accuracy in the object detection algorithm, Redmon et al. transformed the detection problem into a regression problem and proposed YOLO[21]. In 2021, researchers from Megvii proposed YOLOX[22]. Compared with the original YOLO, the detection accuracy and speed of YOLOX have been greatly improved.

2.2. Traffic Flow Detection

Traditional traffic flow detection methods such as geomagnetic coil method, infrared sensor detection method, etc. are easily affected by the environment and have high maintenance costs. Compared with these methods, the traffic flow detection methods based on computer vision have high detection accuracy and strong environmental adaptability. Hilal et al. used CNN to regress the spatial density map of vehicles for the problem of vehicle counting under the aerial images[23]. This method has high recall and accuracy, but it requires a large amount of calculation. In 2019, Wang et al. proposed a vehicle counting method using a compressed domain[24]. Finally, the computation time is reduced, but the accuracy still needs to be improved.

Although the traffic flow detection algorithms based on traffic surveillance video have been well developed, the trade-off between speed and accuracy is still a problem. So, this paper integrates deep learning technology into the traffic flow detection task. Firstly, the improved YOLOX algorithm is used to detect all vehicles in the video frame. Secondly, the DeepSORT algorithm[25] is used to track the detected vehicles. Finally, we use the improved virtual coil method to realize the two-way counting of vehicles in the traffic surveillance video. Through the final test, the method introduced in this paper can accurately and efficiently detect traffic flow information at intersections in different environments.

3. METHODOLOGY

The traffic flow detection task is divided into three parts in this paper, namely vehicle detection, vehicle tracking and vehicle counting. Flow chart of traffic flow detection method is shown in **Fig. 1**.

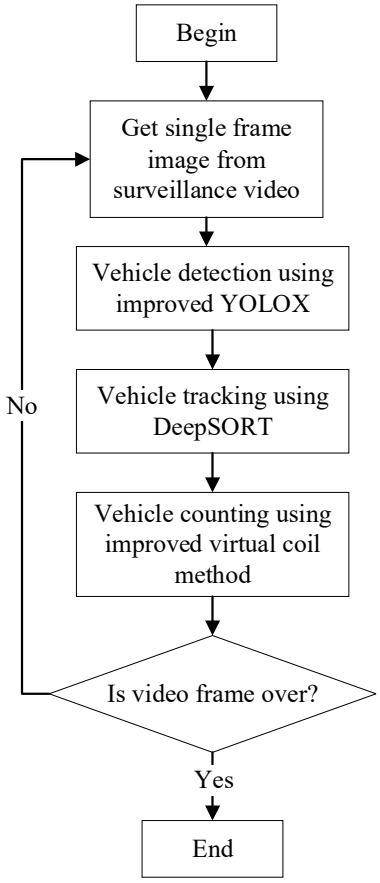


Fig. 1 Flow chart of traffic flow detection.

3.1. Vehicle Detection

Due to the poor robustness of the traditional vehicle detection algorithm. We use the improved YOLOX algorithm. YOLOX is a high-performance object detector proposed by Megvii researchers in 2021. The researchers combined outstanding progress in object detection fields such as decoupled head, strong data enhancement, Anchor Free, and label assignment with YOLO, and finally achieved excellent detection accuracy and speed.

Attention mechanism is a popular direction currently. It can simulate the mechanism by which the human eye observes objects, allowing the network to autonomously learn the important parts of the input image for the current task. Therefore, CBAM module[26] is integrated into YOLOX in this paper, and feature maps of different scales output by feature pyramid network are successively passed through channel self-attention module and spatial self-attention module. Through the above structure, the network can learn that it should pay more attention to a certain channel and a certain region for the current feature map. Part of the structure after adding the CBAM modules is shown in **Fig. 2**.

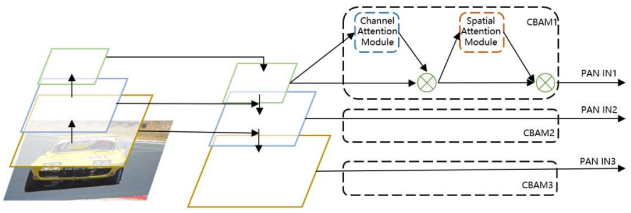


Fig. 2 Network structure after adding CBAM module.

In addition, in order to effectively solve the inconsistency within the feature pyramid, we apply the ASFF module[27] to the outputs of PAN structure, and perform adaptive filtering and fusion on the output feature maps of three different scales. This allows the network to retain more useful information.

In the training process, the training loss is defined as follows:

$$L = L_{cls} + \lambda_1 L_{reg} + \lambda_2 L_{obj} \quad (1)$$

where L_{cls} and L_{obj} are cross entropy loss of categories classification and object classification between foreground and background respectively. The L_{reg} is a linear combination of L_1 loss and $DIoU$ loss used to optimize the predicted position of the bounding boxes.

The final structure of vehicle detection network is shown in **Fig. 3**.

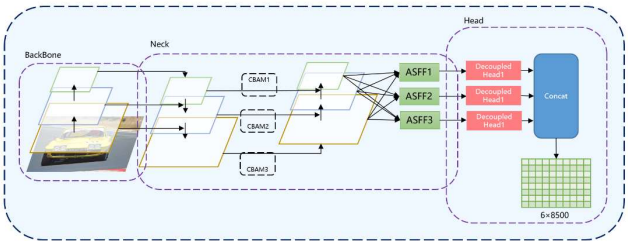


Fig. 3 Network structure of vehicle detection.

3.2. Vehicle Tracking

Since it is difficult for vehicle detection algorithm to accurately detect all vehicles in each frame of the video, so that the traffic flow detection results will be inaccurate if the vehicle detection results are directly counted. Therefore, it is necessary to track the position of the detected vehicles to achieve continuous localization targets.

For the vehicle tracking task, we use the DeepSORT algorithm. DeepSORT is a typical tracking-by-detection (TBD) algorithm which uses Kalman filter to predict the position of the tracking target in the next frame.

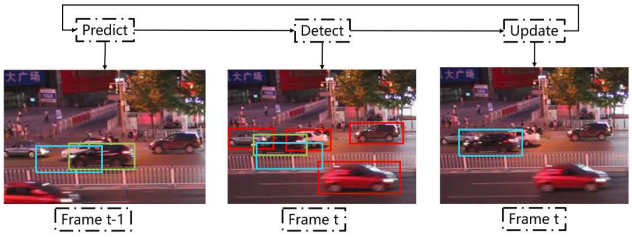


Fig. 4 Core process of DeepSORT.

The core process of DeepSORT is shown in **Fig. 4**. The blue rectangle is the tracking bounding box, the red rectangles are the detection bounding boxes, and the green rectangle is the detection box corresponding to the current tracking target. The algorithm mainly includes three steps. Firstly, the Kalman filter is used to predict the position of the tracked target in the current frame. Secondly, the vehicle detection model is used to complete the vehicle detection in the current frame. Finally, the detection bounding boxes and the tracking bounding boxes of the vehicle are matched and the parameters of the Kalman filter are updated to reduce the cumulative error of the filter.

3.3. Traffic Flow Statistics

We use the improved virtual coil method in traffic statistics. First, a data list consists of ID value and two-dimensional position of each vehicle is generated according to the outputs of the vehicle tracking algorithm, and the position of each vehicle is defined as the center point of the predicted bounding box. Then, we realize two-way counting of vehicles by judging the order of each vehicle passing through two virtual coils in different video frames. The statistical process is shown in **Fig. 5**.



Fig. 5 Flow chart of traffic flow statistics.

As shown in **Fig. 6**, two virtual coils, yellow and blue, are set at different positions in the image. When the center point of the tracked vehicle falls into the blue rectangle and the yellow rectangle in turn, the number of downward traffic flow will be incremented, on the contrary, the number of upward traffic flow will be incremented. Through this method, the counting of traffic flow can be well completed.

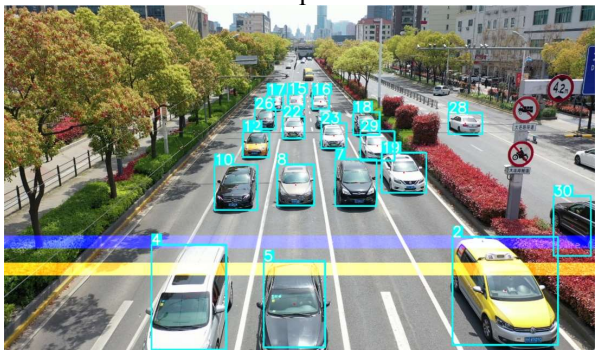


Fig. 6 Virtual coils in traffic flow statistics (yellow and blue coils).

4. EXPERIMENT AND VERIFICATION

4.1. Datasets

We use UA-DETRAC dataset for the model training of vehicle detection. UA-DETRAC is a large-scale dataset for vehicle detection. The dataset consists of ten hours of videos shot in 24 different locations in Beijing and Tianjin, with a video resolution of 960 × 540 pixels. Vehicles in this dataset are divided into four categories: cars, buses, vans, and other vehicles. Weather conditions are also divided into cloudy, night, sunny and rainy.

In order to dynamically evaluate the generalization performance of the model during the experiment, we divide the dataset into training set and validation set. Ultimately, the training set contains 55,818 images and the validation set contains 9,852 images. In addition, in order to prevent the network from overfitting, we use some data augmentation methods include image random horizontal flip, image random HSV enhancement, image random affine transformation, Mosaic, MixUP and so on.

4.2. Parameter Setting

Some hyperparameters during the training process of vehicle detection network are shown in **Table. 1**.

Table. 1 Experimental parameter table.

Parameter Name	Parameter Value
Epoch	100
Batch Size	8
Initial Learning Rate	0.001
Weight Decay	0.0005
Warm-up Epoch	5
Optimizer	SGDM
Momentum	0.9

During the update phase of the learning rate, warm-up strategy is used to accelerate the network convergence. The calculation formula of the learning rate in the warm-up stage is as follows:

$$LR = ILR * \left(\frac{Iter}{WTI}\right)^2 \tag{2}$$

$$WTI = Iters\ Per\ Epoch * Warmup\ Epoch \tag{3}$$

where $Iter$ is the current number of training iterations. WTI is the total number of iterations for warm-up stage. ILR is the initial learning rate and LR is the learning rate after the update.

After the warm-up stage, the learning rate decreases with number of iterations according to the cosine decay mechanism, and the formula is as follows:

$$LR = MLR + 0.5 * (ILR - MLR) * \left(1 + \cos\left(\frac{\pi * (Iter - WTI)}{TI - WTI}\right)\right) \tag{4}$$

$$TI = Total\ Epoch * Iters\ Per\ Epoch \tag{5}$$

where MLR is the minimum learning rate and is set to 0.00001. TI is the maximum number of iterations during network training.

The change curve of the learning rate with the number of iterations is shown in **Fig. 7**.

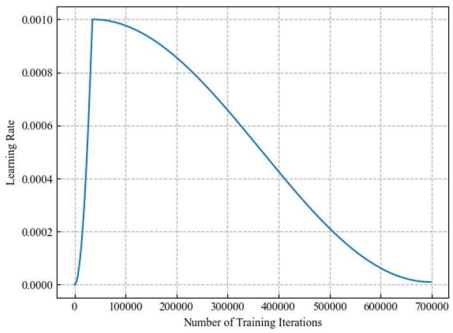


Fig. 7 The change curve of the learning rate.

In order to improve the performance of the vehicle detection model. We add the CBAM module and the ASFF module to the basic network. The comparison curves of average precision (AP) value between improved network and the basic network in the training process is shown in **Fig. 8** and **Fig.9**.

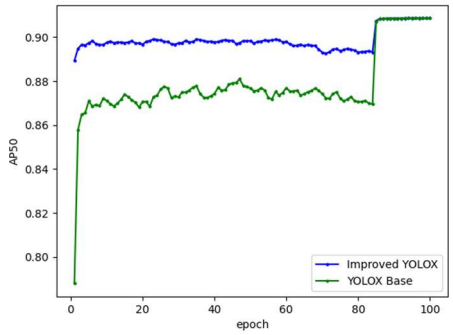


Fig. 8 AP50 performance between basic model and improved model.

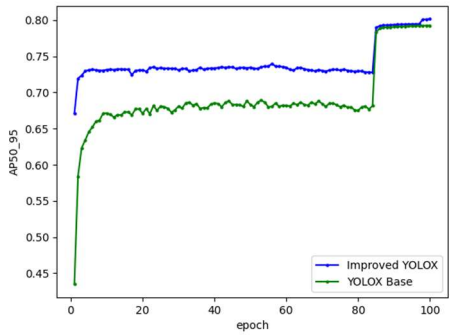


Fig. 9 AP(0.5:0.95) performance between basic model and improved model.

As can be seen from **Fig. 8** and **Fig.9**, the AP and convergence speed of the network after adding the ASFF and CBAM modules are improved. Since the Mosaic and MixUP data enhancements are turned off in the last 15 epochs of training, and the learning rate of the model is reduced, so the AP curves has a certain improvement after 85 rounds.

In the vehicle tracking task, confidence filtering threshold is selected as 0.3, so that detection frames with a confidence level below 0.3 will not be tracked. The maximum distance in the Mahalanobis distance is

selected to be 9.4. If the Mahalanobis distance between the detection bounding boxes and the prediction bounding boxes is higher than 9.4 in the cost matrix, it will be directly considered that the matching fails. The maximum cost value is set to 0.2 in the cost matrix of cosine distance in appearance space. The feature extraction network structure of appearance information is shown in **Table. 2**.

Table. 2 Network structure table.

Layer Type	Kernel Size/Stride	Output Size
Convolution	3×3/1	32×128×64
Convolution	3×3/1	32×128×64
Max Pooling	3×3/2	32×64×32
Residual	3×3/1	32×64×32
Residual	3×3/1	32×64×32
Residual	3×3/2	64×32×16
Residual	3×3/1	64×32×16
Residual	3×3/2	128×16×8
Residual	3×3/1	128×16×8
Dense		128
Batch and L2 Normalization		128

4.3. Testing Results

In order to assess the accuracy of the traffic flow detection system. We use some surveillance videos of different intersections downloaded from the Internet for system testing.

Some of the detection results of the vehicle detection network are shown in the following figures.

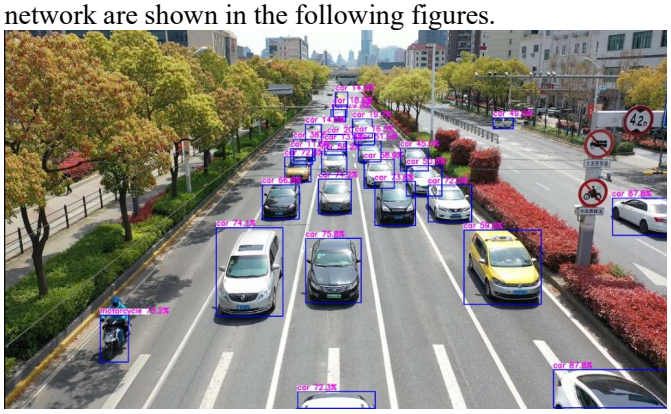


Fig. 10 Vehicle detection at intersection 1 (sunny day).



Fig. 11 Vehicle detection at intersection 2 (rainy day).

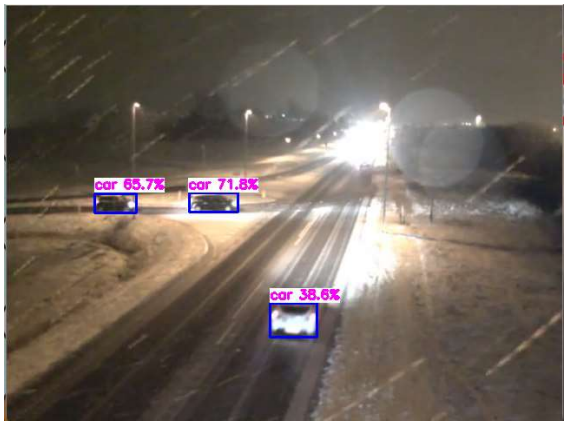


Fig. 12 Vehicle detection at intersection 3 (snowy night).

In order to verify the accuracy of the traffic flow detection system, we conduct several experiments on the traffic flow in different weather and different intersection monitoring perspectives. The accuracy at different intersections and environments is shown in **Table. 3**. Some pictures are shown in the following figures.

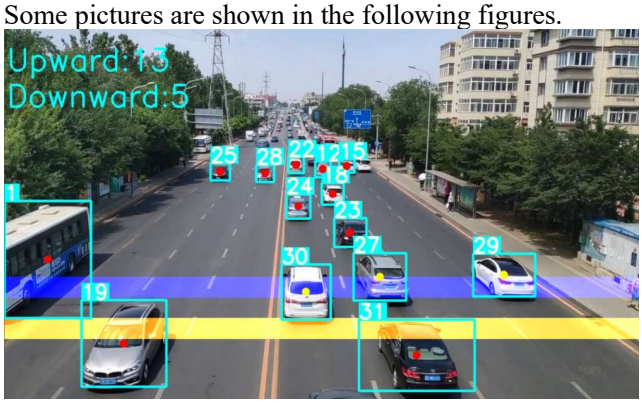


Fig. 13 Traffic flow detection at intersection 1 (sunny day).



Fig. 14 Traffic flow detection at intersection 2 (rainy day).



Fig. 15 Traffic flow detection at intersection 3 (rainy night).

Table. 3 Traffic flow detection results at different intersections.

	Video Duration (Second)	Weather	Day/Night	Actual Number of Upward Vehicles	Actual Number of Downward Vehicles	Tracking Number of Upward Vehicles	Tracking Number of Downward Vehicles	Accuracy
1	30	Sunny	Day	0	32	0	32	100%
2	20	Sunny	Day	26	11	26	10	97%
3	300	Rain	Day	39	38	39	38	100%
4	300	Snow	Day	16	22	14	22	95%
5	300	Rain	Night	17	0	16	0	96%

Through experimental tests, the traffic flow detection system proposed in this paper can adapt to traffic flow detection in different environments, different weather and different resolutions of surveillance video.

5. CONCLUSION

In this paper, a traffic flow detection system is proposed for the problem of traffic flow detection at intersections. We divide the traffic flow detection task into vehicle detection, vehicle tracking and vehicle counting. Specifically, we use the improved YOLOX as the vehicle detection model to detect the vehicles in the different intersections. Then, the detected vehicles are tracked using the DeepSORT algorithm to achieve continuous positioning of the target vehicle between frames. Finally, we use the improved virtual coil method to realize two-way traffic statistics. Through the experimental test under the surveillance video at different intersections, our method achieves high detection accuracy and can detect traffic flow information in real time.

Acknowledgements

This work was partially funded by the National Key Research and Development Plan of China (No. 2018AAA0101000) and the National Natural Science Foundation of China under grant 62076028.

REFERENCES:

[1] Zear A, Singh P K, Singh Y. Intelligent transport system: A progressive review[J]. Indian Journal of Science and Technology, 2016, 9(32): 1-8.

[2] Boukerche A, Tao Y, Sun P. Artificial intelligence-based vehicular traffic flow prediction methods for supporting intelligent transportation systems[J]. Computer networks, 2020, 182: 107484.

[3] C. Chen, B. Liu, S. Wan, P. Qiao and Q. Pei, "An Edge Traffic Flow Detection Scheme Based on Deep Learning in an Intelligent Transportation System," in IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 3, pp. 1840-1852, March 2021, doi: 10.1109/TITS.2020.3025687.

[4] Peppa M V, Bell D, Komar T, et al. Urban traffic flow analysis based on deep learning car detection from CCTV image series[C]//SPRS TC IV Mid-term Symposium "3D Spatial Information Science–The Engine of Change". Newcastle University, 2018.

[5] Lou L, Zhang Q, Liu C, et al. Vehicles detection of traffic flow video using deep learning[C]//2019 IEEE 8th Data Driven Control and Learning Systems Conference (DDCLS). IEEE, 2019: 1012-1017.

[6] Lowe D G. Object recognition from local scale-invariant features[C]//Proceedings of the seventh IEEE international conference on computer vision. IEEE, 1999, 2: 1150-1157.

[7] Bay H, Tuytelaars T, Gool L V. Surf: Speeded up robust features[C]//European conference on computer vision. Springer, Berlin, Heidelberg, 2006: 404-417.

[8] Dalal N, Triggs B. Histograms of oriented gradients for human detection[C]//2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05). Ieee, 2005, 1: 886-893.

[9] Viola P, Jones M. Rapid object detection using a boosted cascade of simple features[C]//Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001. IEEE, 2001, 1: I-I.

[10] Han S, Han Y, Hahn H. Vehicle detection method using Haar-like feature on real time system[J]. World Academy of Science, Engineering and Technology, 2009, 59: 455-459.

[11] Haselhoff A, Kummert A. A vehicle detection system based on haar and triangle features[C]//2009 IEEE intelligent vehicles symposium. IEEE, 2009: 261-266.

[12] Ukani V, Garg S, Patel C, et al. Efficient vehicle detection and classification for traffic surveillance system[C]//International Conference on Advances in Computing and Data Sciences. Springer, Singapore, 2016: 495-503.

[13] Li X, Guo X. A HOG feature and SVM based method for forward vehicle detection with single camera[C]//2013 5th International Conference on Intelligent Human-Machine Systems and Cybernetics. IEEE, 2013, 1: 263-266.

[14] Han F, Shan Y, Cekander R, et al. A two-stage approach to people and vehicle detection with hog-based svm[C]//Performance Metrics for Intelligent Systems 2006 Workshop. 2006: 133-140.

[15] Cao X, Wu C, Yan P, et al. Linear SVM classification using boosting HOG features for vehicle detection in low-altitude airborne videos[C]//2011 18th IEEE International Conference on Image Processing. IEEE, 2011: 2421-2424.

[16] Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2014: 580-587.

[17] Girshick R. Fast r-cnn[C]//Proceedings of the IEEE international conference on computer vision. 2015: 1440-1448.

[18] Ren S, He K, Girshick R, et al. Faster r-cnn: Towards real-time object detection with region proposal networks[J]. Advances in neural information processing systems, 2015, 28.

[19] Liu W, Anguelov D, Erhan D, et al. Ssd: Single shot multibox detector[C]//European conference on computer vision. Springer, Cham, 2016: 21-37.

[20] Fan Q, Brown L, Smith J. A closer look at Faster R-CNN for vehicle detection[C]//2016 IEEE intelligent vehicles symposium (IV). IEEE, 2016: 124-129.

[21] Redmon J, Divvala S, Girshick R, et al. You only look once: Unified, real-time object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2016: 779-788.

[22] Ge Z, Liu S, Wang F, et al. Yolox: Exceeding yolo series in 2021[J]. arXiv preprint arXiv:2107.08430, 2021.

[23] H. Tayara, K. Gil Soo and K. T. Chong, "Vehicle Detection and Counting in High-Resolution Aerial Images Using Convolutional Regression Neural Network," in IEEE Access, vol. 6, pp. 2220-2230, 2018, doi: 10.1109/ACCESS.2017.2782260.

[24] Z. Wang, X. Liu, J. Feng, J. Yang and H. Xi, "Compressed-Domain Highway Vehicle Counting by Spatial and Temporal Regression," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 29, no. 1, pp. 263-274, Jan. 2019, doi: 10.1109/TCSVT.2017.2761992.

[25] Wojke N, Bewley A, Paulus D. Simple online and realtime tracking with a deep association metric[C]//2017 IEEE international conference on image processing (ICIP). IEEE, 2017: 3645-3649.

[26] Woo S, Park J, Lee J Y, et al. Cbam: Convolutional block attention module[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 3-19.

[27] Liu S, Huang D, Wang Y. Learning spatial fusion for single-shot object detection[J]. arXiv preprint arXiv:1911.09516, 2019.