Journal of Combinatorial Mathematics
and Combinatorial Computing
www.combinatorialpress.com/jcmcc

Combinatorial Press

Real-Time Target Detection and Analysis of Complex Traffic Scenes Based on Image Segmentation Algorithm

Yukang Zou[1,] ✉

[1] *School of Traffic and Transportation Engineering, Central South University, Changsha, Hunan, 410075, China*

ABSTRACT

With the development of transportation systems, there is an increasing demand for real-time understanding of traffic scenes using image segmentation algorithms. Therefore, this paper carries out an in-depth study on how image segmentation algorithms for complex traffic scenes can meet the detection requirements of real-time while maintaining accuracy. The article first proposes a lightweight semantic segmentation method based on IDEL_DeepLabV3+, which lightens the IDE_DeepLabV3+ network and optimizes the loss function to improve the positive and negative sample imbalance problem. Then an improved image multi-texture detection method based on Faster RCNN is proposed to improve the detection performance of complex traffic scenes. Finally, the performance of the algorithm designed in this paper is tested through experiments. The performance of the deformable convolution, attention mechanism and feature pyramid improved model is tested and verified, the AP value of the deformable convolution is increased from 41.36 to 47.26, the mAP value of the overall model of the scSE attention mechanism is increased by 0.84%, and the final AP value of the weighted bi-directional feature pyramid network reaches 45.4. The improved DeepLabv3+ network achieves a high AP value of 75.03% in terms of the evaluation index mIOU by 75.03% is better than the original network's 72.26%, so it can be said that we experimentally verified that our improved method enhances the segmentation accuracy of DeepLabv3+ network. The experimental results show that the proposed method in this paper improves the image segmentation accuracy while guaranteeing the segmentation speed, which effectively improves the segmentation effect.

Keywords: IDE_DeepLabV3+, Faster RCNN, image segmentation algorithm, complex traffic scene

1. Introduction

With the acceleration of urbanization, urban traffic problems are increasingly becoming a hotspot for people's attention. Real-time detection of traffic conditions and data analysis can not only provide an important basis for urban traffic management, but also be able to make timely adjustments to the

✉ Corresponding author.
E-mail address: 19307411856@163.com (Y. Zou).
Received 20 February 2024; Revised 10 April 2024; Accepted 25 November 2024; Published Online 16 April 2025.
DOI: 10.61091/jcmcc127b-310

traffic conditions and optimize the operational efficiency of urban road networks and transportation systems [1-4].

In the field of transportation, the collection of real-time traffic data is a key step to realize data-driven traffic management and intelligent transportation [5]. Currently, the common real-time traffic data collection methods mainly include sensor devices, GPS positioning and traffic detection stations [6-7]. Sensor devices acquire real-time data from the road traffic scene through a variety of sensor devices, which is one of the most widely used methods at present [8-9]. For example, with the help of vehicle counters to be able to obtain through a certain section of the traffic flow, hanging on the street light pole video surveillance equipment, it is able to capture the traffic flow rate, the speed of the road operation and other aspects of the data, and other such as geomagnetic, infrared and other sensor devices, respectively, can be used to real-time collection of automobile stops, traffic light operation, sidewalks, and other aspects of the traffic data [10-13]. The use of GPS positioning technology can be used to obtain real-time location, speed and other information of the car, due to the popularity of smart phones, the collection of GPS positioning data has become easier, through the analysis of GPS data can be obtained by the passage time of the roadway, congestion and other detailed information [14-17]. Traffic detection stations are able to acquire data such as traffic signals, weather, and road conditions through online detection of traffic flow [18]. This collection method requires the construction of detection stations, which is more difficult and invested compared to the collection methods such as sensor devices and GPS positioning [19-20]. And with the further development of intelligent transportation, image segmentation algorithms are also applied in real-time target detection and analysis of traffic scenes, which is a method that uses image segmentation technology to separate the detected vehicles from the background, achieve accurate vehicle segmentation, and output in the form of binary images or masks [21-24].

Literature [25] examined a deep learning based end-to-end target detection paradigm including one and two level detectors for UAV images aimed at observing targets in the complex situation of traffic congestion and launched an evaluation effort to improve accuracy and reduce computational cost. Literature [26] describes a method for video target detection and recognition in traffic environments, which addresses the balance point between real-time and accuracy by means of improved YOLOv3 and multi-target tracking algorithms, and the results show that the method has better detection capabilities. Literature [27] proposed an improved fast R-CNN convolutional neural network, and by using it for weak target detection in complex traffic environments, it was verified that the accuracy of this method was better than other methods, and the highest accuracy rate was more than 90%. Literature [28] proposed a traffic flow parameter estimation model and an improved LSTM network for spatio-temporal counting feature recognition, the framework can estimate the traffic flow density and vehicle counts, the method can quickly and accurately count two-way traffic vehicles, which improves the speed and accuracy of the spatio-temporal information processing process. Literature [29] describes that real-time target detection in traffic scenarios is a prerequisite for realizing functions such as electronic surveillance and automatic driving, based on the existing target detection algorithms with low detection efficiency and other problems, the improved YOLOv5 target detection algorithm is used for model training, and pseudo-labeling strategy is used to optimize the training process. Literature [30] proposed a real-time automatic obstacle recognition method for traffic based on computer vision technology, aimed at detecting and recognizing road obstacles such as accident vehicles, illegally parked vehicles, etc., in order to effectively prevent traffic accidents from occurring, and verified the validity of the, with a low computational complexity, to meet the requirements of real-time applications. Literature [31] proposed an integrated multi-data source target detection network for target tracking in complex traffic environments based on the fusion of camera and LiDAR, and pointed out through comparative analysis that the method is better able to meet the accuracy requirements. Literature [32] introduced that intelligent transportation is gradually gaining attention, and traffic video surveillance as an important part of intelligent transportation, traffic environment

detection is a crucial task, and proposed a series of general strategies to optimize the convolutional neural network model for target detection, which demonstrated faster speed. Literature [33] proposed an improved multi-target detection algorithm based on improved denoising anchor box transformer end-to-end object detection, aiming to cope with the problem of low accuracy of object detection in complex scenarios in the task of autopilot perception, and tested its generalization ability in different environments. Literature [34] proposes a vehicle detection model MEB-YOLO, which is better than existing detection methods and is able to perform effective vehicle detection based on real traffic surveillance data, revealing its effectiveness in road target detection. Literature [35] proposes a method named YOLOv8-SnakeVision, which is able to perform excellently in a variety of complex road traffic environments, enhances the detection of small targets and the recognition of multiple targets, and provides strong support for the development of intelligent transportation systems. The above research explores real-time target detection and analysis in complex traffic scenarios based on deep learning, computer vision techniques, YOLOv5 and other methods, which are more superior in terms of accuracy and efficiency in traffic detection and analysis compared with traditional detection methods.

The article firstly designs a lightweight network based on IDEL_DeepLabV3+ to enhance the real-time performance of semantic segmentation network. The number of parameters is reduced by backbone feature extraction network with lighter MobileNetV2 and introducing depth separable convolution. An efficient Transformer module is introduced in parallel with ASPP in the encoder part to effectively acquire the long-distance dependency between different regions in the image. The cross-entropy loss function in the original DeepLabV3+ is replaced by the optimized Focal Loss to improve the positive and negative sample imbalance problem. Then the improved Faster RCNN detection model is proposed. The deformable convolution is used to reshape the convolutional layer in the backbone extraction network and deformable pooling is performed, the attention mechanism scSE is introduced to improve the image expression ability by enhancing the meaningful features, and the weighted bidirectional feature pyramid network is used to perform the feature fusion and then migration learning. Finally, experimental results are obtained by conducting comparative experiments to evaluate the performance of the improved DeepLabv3+ and the pre-improved DeepLabv3+ from qualitative and quantitative perspectives, respectively. Meanwhile, the improved Faster RCNN target detection algorithm proposed in this paper is experimentally validated against several major current target detection algorithms on the COCO dataset.

2. Improvement of DeepLabV3+ semantic segmentation model for complex traffic scene images

2.1. Lightweight semantic segmentation model improvement

2.1.1. Deep Separable Convolution vs. Ordinary Convolution. Depth separable convolution has significant advantages in the lightweight semantic segmentation task of complex traffic scenes, which effectively reduces the number of parameters of the model and realizes the lightweight of the model by splitting the standard convolution operation into two steps: depth convolution and point-by-point convolution. Deep separable convolution not only has advantages in lightweighting, but also can still effectively capture the semantic information in the image while keeping a relatively small model size, which makes it perform well in the semantic segmentation task of traffic scenes.

For a given convolutional layer, if the depth of the input tensor is C_{in} and the depth of the output tensor is C_{out}, the number of depth-separable convolutional parameters is shown in equation (1):

$$paremeters = h \cdot w \cdot C_{in} \cdot C_{out} + C_{in} \cdot C_{out} \tag{1}$$

Eq. (2) is the parameter ratio of the deep and standard convolution, where, denotes the proportion

of parameter reduction of the deep separable convolution with respect to the standard convolution on each output channel, $\frac{1}{C_{out}}$ denotes the proportion of the parameters of each output channel normalized with respect to the total number of parameters, and $\frac{1}{h \times w}$ for each parameter of the convolution kernel is a normalized proportion relative to the total number of parameters.

$$\frac{h \cdot w \cdot C_{in} \cdot C_{out} + C_{in} \cdot C_{out}}{h \cdot w \cdot C_{in} \cdot C_{out}} = \frac{1}{C_{out}} + \frac{1}{h \cdot w} \tag{2}$$

2.1.2. MobileNetV2 lightweight backbone feature extraction network. To enhance feature extraction, features are first upscaled by 1×1 convolution and filtered using depth-separable convolution. The features are then projected back to a lower dimensional representation by linear convolution. In neural networks, it is common to reduce the amount of computation by setting the step size to reduce the number of model parameters. In the MobileNetV2 convolutional block structure, when stride=1, the input features and output feature dimensions are kept the same and a residual structure is introduced. When stride=2, there is no residual structure [36]. The inverse residual structure of MobileNetV2 network improves the efficiency of memory usage and achieves low memory footprint computation.MovileNetV2 convolutional block structure is shown in Fig. 1.

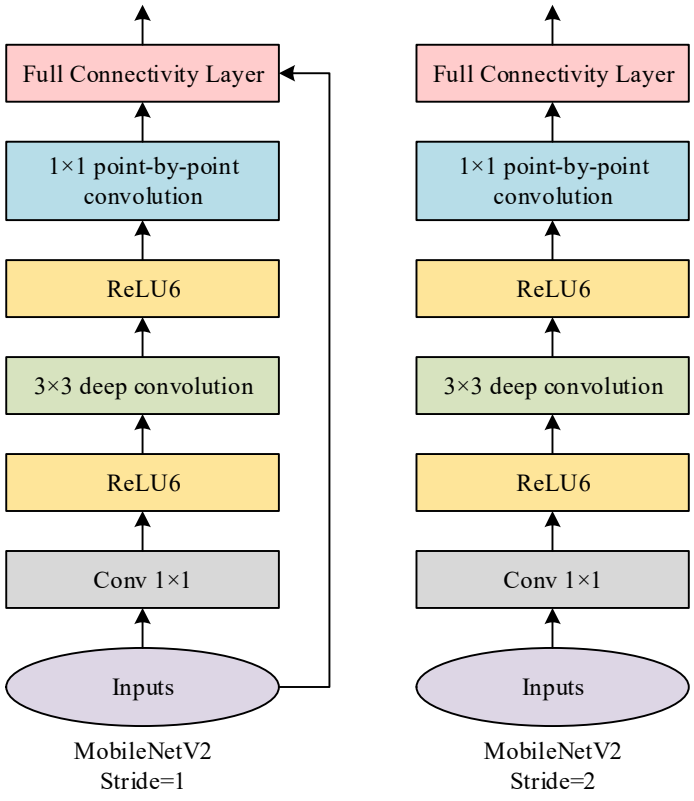


Fig. 1. Movilenetv2 convolution block structure

2.1.3. Efficient Transformer Structure:

1) Visual Transformer Structure. In order to accommodate the input requirements of the visual Transformer network, this paper serializes the input feature map $x \in \mathbb{R}^{H \times W \times C}$. The process involves

splitting the feature map into small chunks of size $p \times p$, with each image divided into $\frac{H \times W}{p^2}$ patches.This process allows the feature map to be reconstructed as a one-dimensional sequence suitable for use with the visual Transformer $\left\{x_p^i \in \square^{p^2 \times c} \mid i = 1, \ldots, N\right\}$, where i denotes the ith patch in the sequence, H and W denote the height and width of the feature maps, and C denotes the number of feature map channels.

Since the position information is lost after the image has been cut and rearranged, in order to more accurately localize the classes of pixels, the position information needs to be encoded before being imported into the network. The visual Transformer structure uses learnable vectors that are consistent with the dimension of the sequence to encode the position information, and these position encoding vectors are accumulated element by element to each patch of the sequence, thus effectively embedding the position information, as shown in equation (3):

$$z_0 = \left(x_p^1 E; x_p^2 E; \ldots; x_p^N E\right) + E_{pos} \tag{3}$$

where $x_p^i \in \square^{p \times p}$ denotes the i th patch. the positional encoding of the $E \in \square^{\left(p^2 \times C\right) \times D}$ embedding Used to linearly map each pathch to the dimension $D; E_{pos} \in \square^{N \times D}$ [37]. The $E_{pos} \in \square^{N \times D}$ denotes the embedding position, which provides positional information for each patch of the image.

The visual Transformer encoder consists of L-layer Multihead Attention Mechanism (MSA) and Multi-Layer Perceptron (MLP) as shown in Eqs. (4)-(5). The output of the L layer can be written as:

$$z_l' = MSA\left(LN\left(z_{l-1}\right)\right) + z_{l-1} \tag{4}$$

$$z_l = MLP\left(LN\left(z_l'\right)\right) + z_l' \tag{5}$$

where $LNK(\cdot)$ denotes layer normalization and z_l denotes image coding.

(1) Multihead self-attention mechanism (MSA). Multihead attention is equivalent to integrating n independent self-attention mechanisms. First, the data x is fed into each of the n self-attention mechanisms to obtain n weighted feature matrices as defined in Eqs. (6) and (7). These n feature matrices are stitched together to form a large feature matrix, and finally the final output is obtained by the operation of matrix W^O multiplication.

$$head_i = attention\left(X_i W_i^Q, X_i W_i^K, X_i W_i^V\right) \tag{6}$$

$$MSA\left(Q, K, V\right) = Concat\left(head_1, \cdots, head_i\right) W^O \tag{7}$$

where W_i^Q, W_i^K, W_i^V denote the weight matrices of the ith head, respectively. $head_i$ is the output of the ith self-attention mechanism: W^O is the model training weight matrix.

(2) Multilayer Perceptron Machine (MLP). The multilayer perceptron consists of two fully connected layers and the activation function ReLU, the multilayer perceptron is shown in equation (8).

$$MLP(X) = \max\left(0, XW_1 + b_1\right) W_2 + b_2 \tag{8}$$

where X denotes the output of the multi-head attention layer, W_1, b_1 and W_2, b_2 denote the weight matrix and bias of the two fully connected layers, respectively.

Ultimately, the generated feature maps are remapped to the order of the original space through the operation of the multi-head attention layer and the multilayer perceptron, and up-sampling is performed to obtain feature maps matching the resolution of the ASPP output. The visual Transformer

structure is schematically shown in Fig. 2.

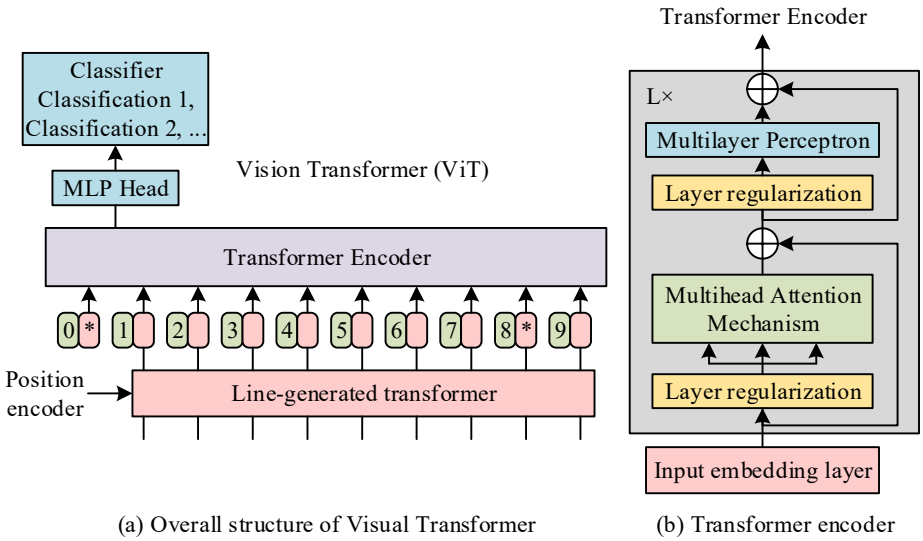


Fig. 2. Visual transformer structure

2) Efficient Transformer Structure. This section introduces the Efficient Transformer structure (ET). In order to avoid excessive memory usage and computational load, Efficient Transformer improves the multi-head attention mechanism. After layer normalization, Efficient Transformer sets up a reduction layer to halve the number of channels and reduce the dimensionality of the feature map, which reduces the memory usage. Next, the feature graph is projected into three matrices, Q (query), K (key), and V (value), by a full join operation. Specifically, in the Efficient Multi-Headed Attention (EMHA) mechanism, Q, K, and V are first partitioned into s segments, followed by the corresponding execution of Q_i, K_i, and V_i of the scaled dot product attention. Next, the obtained $O_1 \cdots O_s$ are concatenated together to obtain the entire output O. The Q_i is calculated as shown in equation (9). Where Q, K and V denote the query, key and value metrics and d is the embedding dimension. All outputs $(O_1 \cdots O_s)$ are concatenated together to generate the entire output feature O.

$$Q_i = soft\max\left(\frac{Q_i K_i^T}{\sqrt{d}}\right)V_i \tag{9}$$

2.2. Loss function

2.2.1. Focal Loss Loss Function. Focal Loss makes the model pay more attention to the categories with fewer samples and more difficult to learn during the training process by introducing the penalty weights α for positive and negative samples, so that it can solve the problem of category imbalance more efficiently and improve the classification performance of rare categories.The formula of Focal Loss loss function is shown in (10):

$$FL(p_p) = -\alpha(1-p_p)^{\gamma}\log(p_p) \tag{10}$$

α is the weight coefficient, γ is the hyperparameter, and p_p is the predicted probability of a category. In the case of $\gamma = 0$, the Focal Loss is equivalent to the traditional cross-entropy loss. As the value of γ rises, the modulation coefficient grows. When there is an error in the classification of the sample, the value of p_p is very small and the modulation coefficient is close to 1, at which point there is no significant difference compared to the original loss function value. When p_p is close to 1, it means that the samples are classified accurately and easily, while the modulation coefficient is close

to 0, which has a relatively small effect on the total loss function. Overall Focal Loss has a relatively small processing loss weight for simple samples that are easy to recognize and a large loss weight for complex samples that are not easy to distinguish.

2.2.2. Improving the Focal Loss Loss Function. According to the previous analysis, it can be seen that the closer the value of the horizontal coordinate of the Focal Loss is to 1, the better the classification effect is, so it is necessary to adjust the network's attention to the easy-to-partition samples by decreasing the value of the loss of this part of the sample. The closer the value is to 0, the worse the segmentation effect is, so it is necessary to increase the value of this part of the sample loss to enhance the network's ability to learn difficult to segment samples. Usually, common experiments choose $\alpha = 1$, $\gamma = 2$ to train the loss function, as shown in Eq. (11), which makes it possible to adjust the output of the loss function more efficiently when dealing with sample imbalance and difficult to segment samples.

$$FL\left(p_p\right) = -\left(1 - p_p\right)^2 \log\left(p_p\right) \tag{11}$$

Based on the above analysis, this thesis improves the Focal Loss loss function which has the following expression:

$$FL_N\left(p_p\right) = -\left(a - p_p\right)\left(b - p_p\right)\log\left(p_p\right) \tag{12}$$

2.3. Performance Test of Improved Scene Semantic Segmentation Algorithm for DeepLabV3+

2.3.1. Selection of data sets. In this section, Cityscapes, the most representative dataset of traffic scene, is used for experimental verification. CityScapes dataset is one of the most authoritative datasets in the field of autonomous driving, which mainly provides image segmentation data for testing the effect and performance of algorithms in the field of driving, and contains a large number of dynamic targets in the road scene, and the layout of the scene is relatively rich and complex. This dataset contains a large number of dynamic targets in the road scene, and the layout of the scene is relatively rich, and the background is also more complex, which is more suitable for the image segmentation task in the traffic scene studied in this paper.

The Cityscapes dataset defines a total of 30 classes in the annotations, and among these 30 classes, based on the number of their samples and the actual situation in the annotations, the categories with very small sample numbers are excluded in the laboratory, and 19 classes with relatively uniform distribution are used for evaluation. These 19 classes are: road, sidewalk, person, rider, car, truck, bus, motorcycle, bicycle, train, pole, vegetation, train, sky, traffic light, traffic sign. 19 The pixel number pairs of the individual classes are shown in Figure 3.

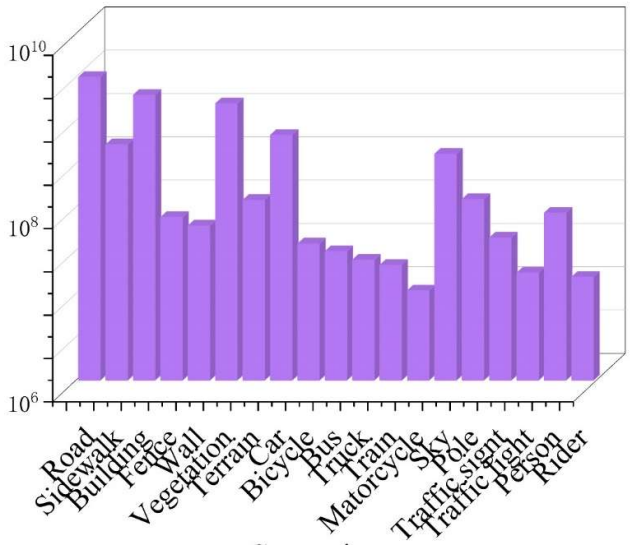


Fig. 3. The comparison of the number of colors of 19 classes

2.3.2. Experimental Procedure and Result Analysis. In this chapter, the Cityscape dataset is used as the training and testing dataset, and the DeepLabv3+ algorithm with different backbone feature extraction networks is first trained and tested, and the performance differences are compared, and then the SoftPool pooling method is used to train and test the improved DeepLabv3+ model that employs the MobileNetV2 backbone network , and evaluates its performance, and finally the improved algorithm proposed in this paper is compared with other classical algorithms and its segmentation effect is verified on real captured images. We train and test DeepLabv3+ with different backbone feature extraction networks on the Cityscapes dataset in the same environment and compare the metrics such as m IOU, FPS, and model size. The performance comparison of DeepLabv3+ with different backbone networks is shown in Table 1. From the table, it can be seen that although the accuracy of DeepLabv3+ segmentation network using MobileNetV2 network as the backbone feature extraction network has decreased, it has achieved a significant increase in detection speed, and also achieved a significant reduction in model size, i.e., the number of covariates, which is only 38.91MB, which compares with the commonly used Xception65 and ResNet101, both reduced to about one-tenth. In terms of running speed, DeepLabv3+ based on MobileNetV2 backbone network structure is far ahead of other backbone networks with 51.63fps, and the inference speed can reach the real-time level. Since the running speed is also affected by the hardware devices, the real-time performance of the model will be better if it is tested on better performance devices.

Table 1. Different main dry mesh of deeplabv3+ performance control

Main dry network	mIOU(%)	FPS	Model size(Mb)
Xception65	80.74	4.45	432.24
ResNet101	78.1	17.83	446.44
ResNet50	77.08	24.21	305.5
MobileNetV2	74.44	51.57	38.91

Next, using MobileNetV2 as the basis of the backbone feature extraction network, the pooling method in the ASPP structure of the DeepLabv3+ network was replaced with SoftPool for training and testing, and the results were compared and analyzed. The variation of mIOU and loss with the number of training rounds is shown in Fig. 4. From the figure, it can be seen that with the increasing number of training rounds, the loss first shows a fast decreasing trend, followed by a slow decreasing trend and then tends to stabilize and converge. And the trend of mIOU on the validation set is similar to the

trend of loss, which starts with a rapid increase, next shows a slow increasing trend, and finally tends to stabilize.

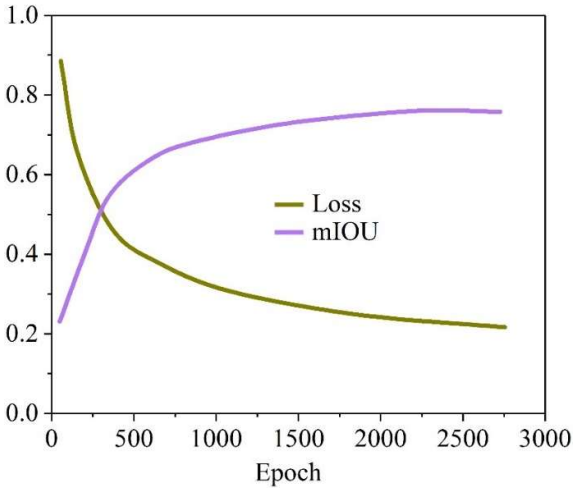


Fig. 4. The mIOU and loss changes with the training rotation

The performance of the improved DeepLabv3+ algorithm is next evaluated on the validation set from the commonly used performance metric mIOU, and the segmentation accuracy comparison is shown in Fig. 5. From the figure, it can be seen that in a total of 19 categories, the improved DeepLabv3+ shows a decrease in accuracy only in the three categories of pole, traffic light and traffic sign compared to the pre-improvement one, while the accuracy is higher than that of the original DeepLabv3+ network in the other 16 categories, among which in sidewalk, fence, terrain, truck, bus, train, and motorcycle, while the accuracy in the other 9 categories is also slightly higher than that of the original DeepLabv3+ network. Overall, the improved DeepLabv3+ network outperforms the original network's 72.26% with 75.03% on the evaluation metric mIOU presented above, so it can be said that we have experimentally verified that the improved method enhances the segmentation accuracy of the DeepLabv3+ network.

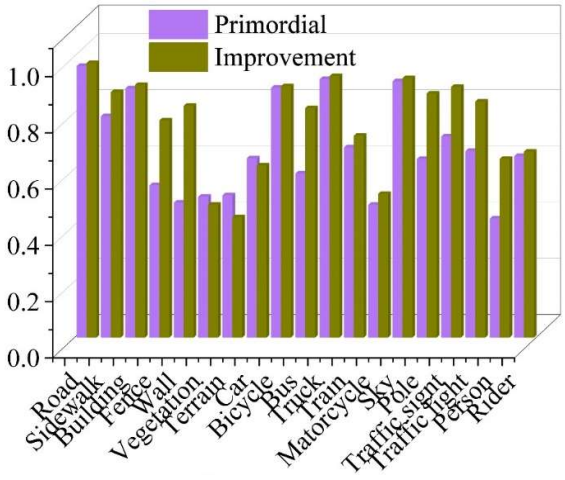


Fig. 5. Segmentation accuracy contrast

3. Improved Faster RCNN detection model

In order to improve the image detection accuracy of complex traffic scenes in an efficient and parallel manner, this section proposes a multi-texture detection model based on Faster RCNN network and

combining migration learning, deformable convolution, scSE attention mechanism and weighted bi-directional feature pyramid network.

3.1. Faster RCNN network architecture

Faster RCNN is a network model with high accuracy detection rate in the field of target recognition. Firstly, the complex traffic scene images are preprocessed with data. The processed images are passed through the backbone feature extraction network for feature extraction to obtain the feature map. A sliding window is used in the region suggestion network to generate the corresponding candidate frames, while the feature map is passed through a set of 3×3 convolutional layers, and then two parallel 1×1 convolutional layers are used to perform classification (cls) and bounding-box regression (reg) respectively, to obtain a candidate image containing the target probability. The feature maps and candidate images go to the region pooling layer (ROI Pooling) for pooling and specification of feature maps containing candidate regions [38]. Finally, the feature layer intercepted by the suggestion box is resized and further convolved to complete the classification and regression of the target object.

3.2. Network design

3.2.1. Deformable Convolution. Deformable convolution introduces a standard 3×3 convolutional layer on top of the original conventional convolution, generating N two-dimensional offset variables in the x, and y directions, which has a convolutional kernel defined as:

$$R=\{(-1,-1),(-1,0),\cdots,(0,1),(1,1)\} \tag{13}$$

The feature matrices obtained from standard convolution operation and deformable convolution are respectively:

$$y(p_0)=\sum_{p_n\in R}\omega(p_n)\cdot x(p_0+p_n) \tag{14}$$

$$y(p_0)=\sum_{p_n\in R}\omega(p_n)\cdot x(p_0+p_n+\Delta p_n) \tag{15}$$

where p_n is an enumeration of all positions in R. The Δp_n represents the position offset vector. The number of output feature channels of the ordinary convolution operation is N, and the deviation features of the deformable convolution output are kept consistent with it, so that the dimension of the stacked channels is $2N$.

Deformable convolution has the probability that the dimension of the convolved matrix is extended too large, and the situation in which the receptive field of the convolution kernel is larger than the target region occurs, which in turn introduces invalid information. In order to ensure the accurate extraction of effective information, deformable convolution second generation is newly added a weight matrix, for deformable convolution generation to find effective information in the region of the location of the weight is given to the weight of the weight of the acquisition of its weight and the offset matrix acquisition approach is the same as that of the improved formula is as follows:

$$y(p)=\sum_{p_n\in R}\omega(p_n)\cdot x(p_0+p_n+\Delta p_n)\cdot\Delta m_n \tag{16}$$

Ordinary Region of Interest Pooling layer (RoI Pooling) is to divide a feature map into a number of regions with regular shape and fixed position, which brings limitations to the feature mapping process and thus reduces the accuracy of the detection results. The pooling operation can also draw on the deformable idea, adding offset features to its original structure with the following formula:

$$y(k)=\sum\frac{x\cdot(p_n+\Delta p_n)\cdot\Delta m_n}{k} \tag{17}$$

3.2.2. scSE Attention Mechanism. In order to improve the image feature expression ability, the scSE attention mechanism is proposed to combine the spatial compression channel module (sSE) and the channel compression spatial module (cSE), with the main purpose of highlighting useful features and suppressing useless features. The specific algorithm steps are as follows:

Let the input and output feature maps be U, $\hat{U}$, respectively.

Step 1: Compress U over the channel and set the feature layer $U=\left[u^{1.1},u^{2.2},\cdots,u^{i.j},\cdots,u^{H,W}\right]$, (H,W) is the feature map size, (i,j) is the spatial location of the feature map, and the space is compressed by the 1×1 convolution of the W_{sq} with weight W_{sq} to obtain the feature map with the channel number of 1, $q=W_{sq}\times U$.

Step 2: After normalization of q by Sigmoid function, $\hat{U}_{sSE}=\left[\sigma\left(q_{1,1}\right)u^{1,1},\cdots,\sigma\left(q_{i,j}\right)u^{i,j},\cdots,\sigma\left(q_{H,W}\right)u^{H,W}\right]$ is obtained where $\sigma\left(q_{i,j}\right)$ is the significance of the spatial location coordinates of the feature map.

Step 3: Process U spatially by setting its feature layer $U=\left[u_1,u_2,\cdots,u_c\right]$, and obtain the vector z by global pooling.

Step 4: The vector z is processed through the fully connected layer with weight $\left(W_1,W_2\right)$ twice for the information, and the ReLU activation function enhances the independence of each channel to obtain the vector $\hat{z}=W_1\left(\delta\left(W_2z\right)\right)$ in C dimensions.

Step 5: Finally, after normalization of $\hat{z}$ by Sigmoid function, we get $\hat{U}_{cSE}=\left[\sigma\left(\hat{z}_1\right)u_1,\sigma\left(\hat{z}_2\right)u_2,\cdots,\sigma\left(\hat{z}_c\right)u_c\right]$, where $\hat{z}$ stands for the importance level of the ith channel.

Step 6: The scSE module mainly implements the input feature map U in the spatial and channel attention module, and sums up the output results to finally obtain the output feature map $\hat{U}_{scSE}=\hat{U}_{sSE}+\hat{U}_{cSE}$. The scSE attention mechanism implementation process is shown in Fig. 6.

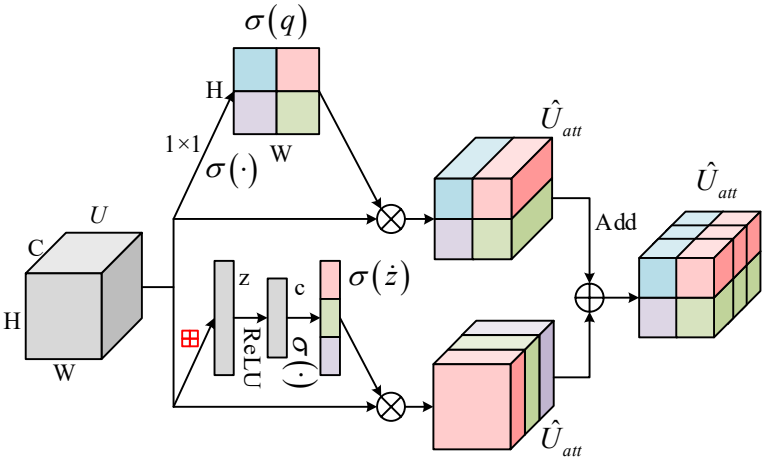


Fig. 6. Scse attention mechanism implementation process

3.2.3. Weighted Bidirectional Feature Pyramid Networks. Complex traffic scene images with different texture styles are variable, which is very likely to cause the problem of missed detection, and in order to prevent affecting the recognition effect of the model, the features need to be fused at multiple scales. The initial convolutional module used for feature fusion was FPN, which fused multilayer features by successive upsampling and using a top-down channel. Later versions of PANet based on FPN added a bottom-up channel to compensate for the unidirectionality of the information. And the latest version Weighted Bidirectional Feature Pyramid (BiFPN) adds an extra line to the simplified PANet to incorporate more features without increasing the cost.

The purpose of multiscale feature fusion is to aggregate features with different resolutions and semantics. During the feature fusion process, additional weights are added to the input features to avoid inconsistency in the output of the final features to the network. The use of weighted fusion enables the network to learn the importance of different input features, the different output formulas for adding extra weights are as follows:

$$P_6^{out} = Conv\left(P_6^{in}\right) \tag{18}$$

$$P_5^{out} = Conv\left(P_5^{in} + \mathrm{Re}\,size\left(P_6^{out}\right)\right) \tag{19}$$

$$P_2^{out} = Conv\left(P_2^{in} + \mathrm{Re}\,size\left(P_3^{out}\right)\right) \tag{20}$$

The selected weighted fusion is a fast and normalized feature fusion, after passing the RELU activation function to ensure the stability of the values [39]. The final output equations of P_5^{td} and P_5^{out} are as follows:

$$P_5^{td} = Conv\left(\frac{\omega_1' P_5^{in} + \omega_2 \,\mathrm{Re}\,size\left(P_6^{in}\right)}{\omega_1' + \omega_2' + \omega_3' + \varepsilon}\right) \tag{21}$$

$$P_5^{out} = Conv\left(\frac{\omega_1' P_5^{in} + \omega_2' P_5^{td} + \omega_3' \,\mathrm{Re}\,size\left(P_4^{out}\right)}{\omega_1' + \omega_2' + \omega_3' + \varepsilon}\right) \tag{22}$$

where P_5^{td}, P_5^{out} denote the intermediate features and final outputs of level5 in top-down and bottom-up, respectively, and all other features are constructed using the same method. Repeated bi-directional cross-scale connections with weights are fast normalized feature fusion combined to form the final weighted bi-directional feature pyramid network.

3.2.4. Fine-tuning transfer learning. Transfer learning is learning by applying the knowledge learned from a source dataset to a new target. Fine-tuning is one of the commonly used techniques for this, and is necessary to improve the generalization ability of the model when the target dataset has fewer samples. The specific process is: add a fully connected layer after the existing base model, in order to make it effective, so that the existing model is not trained at this time, the existing network structure is frozen, only the custom added fully convolutional network is trained, and then after training, the existing model is thawed and added to the fully convolutional network for training together.

3.3. Model structure

In order to more efficiently carry out the multi-texture study of complex traffic scene images, the FasterRCNN model is improved accordingly, and in the feature extraction network, the conventional convolution block is used with Resnet50, which has a better performance at present, with a total of 5 stages. Deformable convolution is introduced after the backbone network, and the Conv2-Conv5 regular convolution in the original feature extraction network is reshaped to adjust its own shape when extracting features, while deformable RoI Pooling is used to make the model more accurately fit the shape and positional features of the texture. The scSE attention mechanism is introduced after the original Conv1 and the replaced Conv2~Conv5 convolutional blocks, while the weighted bi-directional pyramidal network model is used to perform feature fusion on the convolutional blocks with the added attention mechanism. All layers before the last fully connected layer are migrated, the classification information of the last fully connected layer is modified, and random weights are re-assigned to the last layer to maximize the performance of the neural network as much as possible in the final feature extraction. The overall structure of the network model used is shown in Figure 7.

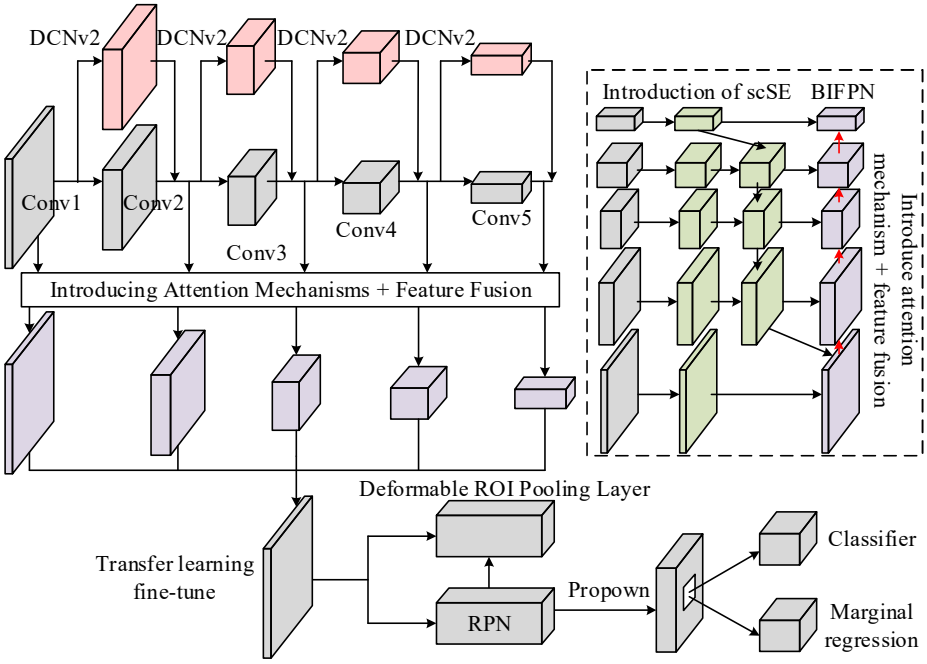


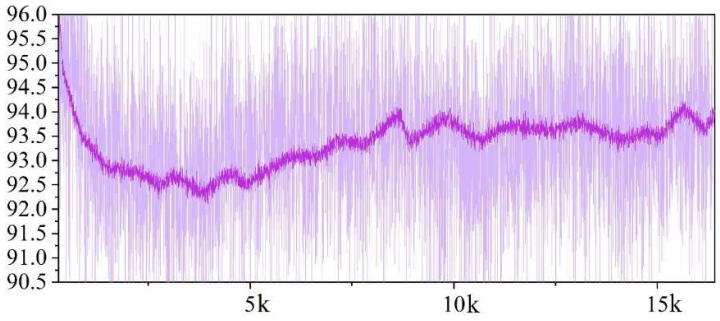
Fig. 7. Overall structure of the proposed network model

4. Performance test of the improved Faster RCNN detection algorithm

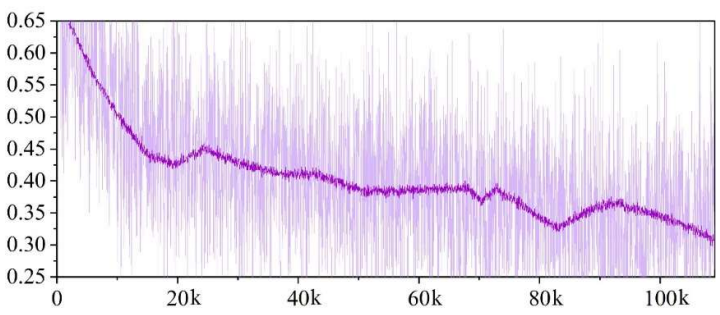
4.1. Improved model validation and analysis based on feature pyramid network

This section verifies whether the weighted bidirectional feature pyramid network can effectively improve the performance of the target detection network and further obtains the effectiveness of the weighted bidirectional feature pyramid network.

The training accuracy and loss plot based on weighted bidirectional feature pyramid network is shown in Fig. 8 (Fig. a shows the accuracy and Fig. b shows the loss value).



(a) Acc_train



(b) Loss_train

Fig. 8. Training accuracy and loss of the network based on weighted two-way characteristics

The performance of the ten detection algorithms is compared on the COCO dataset and the performance comparison of the ten detection algorithms on the COCO dataset is shown in Table 2. In the table, the last four rows are the improvement results made in this subsection for the weighted bidirectional feature pyramid network algorithm. The use of weighted bidirectional feature pyramid network makes a very significant improvement in the performance of the detection algorithms, with the AP value increasing from 35.13 to 41.86, and the APSS value increasing from 16.94 to 25.16 for small target detection, so the addition of the FPN network module is very necessary. Finally, the weighted bidirectional feature pyramid network is further improved by introducing PANet and BFP network, and the final AP value reaches 45.4. Compared with the first five rows of mainstream algorithms, the performance is better than that of SSD513, RetinaNet, and the performance is comparable to that of the newest YOLO v4 algorithm, and the performance is even better in small target detection.

Table 2. COCO data concentration 10 detection algorithm performance comparison

	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
YOLO v4	CSPDarknet53	47	65	47.6	28.8	48.8	53.8
SSD512	ResNet101	30.8	50.3	36.9	9.1	32.4	45.7
RetinaNet	ResNet101	38.9	60.9	43.3	20.5	42.9	51.8
Cascade RCNN	ResNet101	41.1	58.7	43.7	25.4	48.4	54.1
Faster RCNN	VGG-16	34.8	55.3	40.4	18	38.1	50.3
Faster RCNN+ResNet50	ResNet50	35.13	63.05	40.49	16.94	39.51	49.53
Faster RCNN+ResNet50+FPN	ResNet50	41.86	65.82	45.24	25.16	43.25	61.61
Faster RCNN+ResNet50+PANet	ResNet50	44.08	63.42	46.38	28.92	48.84	64.87
Faster RCNN+ResNet50+ BiFPN	ResNet50	45.4	69.7	46.62	26.23	46.02	59.94

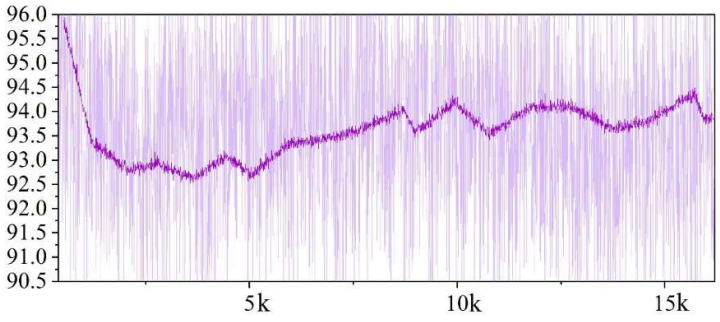
The results of eight classes of traffic target detection based on weighted bidirectional feature pyramid network are shown in Table 3. From the table, it can be seen that the weighted bidirectional feature pyramid network model has a more significant improvement in each class compared to the benchmark Faster RCNN model, and the mAP value of the overall model is improved by 9.66%. Overall, the weighted bi-directional feature pyramid network improvement strategy based on weighted bi-directional feature pyramid network can effectively improve the accuracy of target detection. Relative to the weighted bidirectional feature pyramid network model, the improvement strategies based on PAnet and BFP also have a small amount of improvement, due to the less obvious improvement rate, weighed against the impact of training time and detection speed brought about by the complexity of the model. Therefore, the improvement strategy based on weighted bidirectional feature pyramid network is used in all subsequent experiments in this paper.

Table 3. The eight kinds of traffic target detection results based on the characteristics pyramid

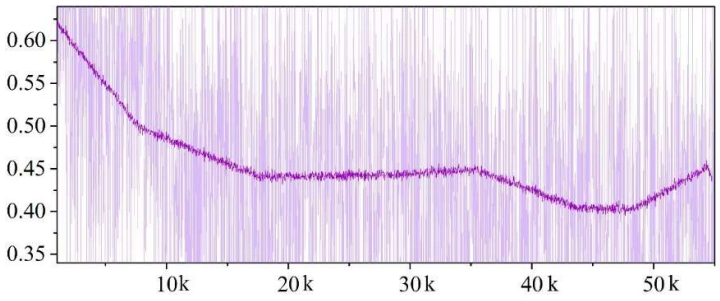
Categories	Model frame			
	Faster RCNN	Faster RCNN+FPN	Faster RCNN+PANet	Faster RCNN+BiFPN
Man	45	49.2	53.8	53.3
Bicycle	28.1	27.3	27.5	29
Car	34.4	44.2	40.3	42.1
Motorcycle	32.7	37.5	39.3	36.9
Bus	53.9	58.1	57.8	59.8
Truck	25.3	28	31.9	29.3
Traffic Light	19.7	29.5	27.1	29.3
Traffic Mark	50	65.4	64.9	64.4
mAP	32.9	42.56	45.98	46.1

4.2. Improved model validation and analysis based on the attention mechanism

The spatial attention mechanism makes the candidate frames more convergent to the target, and the candidate frames generated by the weighted bidirectional feature pyramid network are made more convergent to the detection target by the scSE attention mechanism. In this subsection, the scSE attention mechanism is experimentally validated and analyzed to verify the improvement on the performance of the detection network. The training accuracy and loss diagram of the improved model is shown in Fig. 9.



(a) Acc_train



(b) Loss_train

Fig. 9. Based on the spatial attention mechanism, the accuracy and loss

In this subsection, a comparison experiment of eight target detection models is designed. The performance of the eight detection algorithms is compared on the COCO dataset, and the performance comparison of the eight detection algorithms on the COCO dataset is shown in Table 4. In the table, the last row is the result of the improvement made to the Faster RCNN algorithm based on the scSE attention mechanism in this subsection. Adding the scSE network to the Faster RCNN-ResNet50-FPN resulted in an increase in the AP value from 46.06 to 46.4, and the detection algorithmicity was slightly improved, so the addition of the spatial attention mechanism has a role in the model performance improvement. Compared with the first five rows of mainstream algorithms, the performance is better than SSD513, RetinaNet, and slightly weaker than YOLO v4 algorithm.

Table 4. The performance of eight detection algorithms was compared

	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
YOLO v4	CSPDarknet53	43.8	63.4	48.8	27.4	50.5	56.9
SSD512	ResNet101	31.3	50.1	35.6	8.7	30.7	48.9
RetinaNet	ResNet101	42.1	56.8	43.5	24.4	43.5	51.3
Cascade RCNN	ResNet101	40.2	62.8	45.7	23.4	46	56.6
Faster RCNN	VGG-16	36.1	57.8	39.3	15.6	39.8	53.2
Faster RCNN+ResNet50	ResNet50	35.23	61.25	41.89	18.04	37.51	47.13
Faster RCNN+ResNet50+FPN	ResNet50	46.06	67.12	46.14	23.16	43.85	61.01
Faster RCNN+ResNet50+FPN+scSE	ResNet50	46.4	67.75	41.47	29.19	49.19	58.68

The results of the detection experiments of common targets in eight categories of traffic scenarios are shown in Table 5. From the table, it can be seen that the scSE attention mechanism improves on each category compared to the benchmark Faster RCNN model, and the overall model's m AP value is improved by 8.6%. The scSE attention mechanism improves in all categories except for a slightly lower AP value in the category of traffic signs, and the mAP value of the overall model improves by 0.84%,

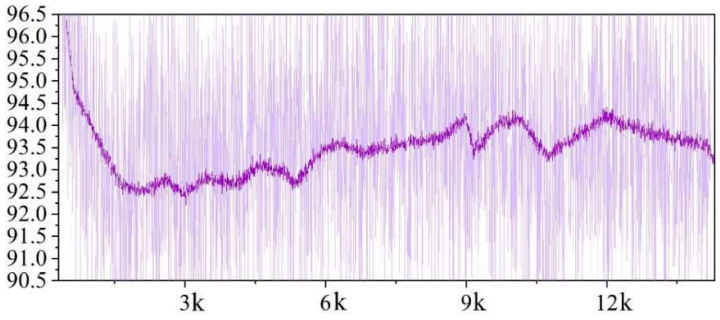
from which it can be concluded that the improvement of introducing spatial attention mechanism in the model is very effective.

Table 5. Eight kinds of traffic target detection results based on spatial attention mechanism

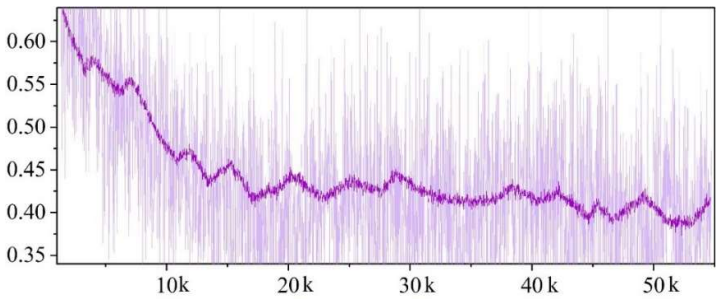
Categories	Model frame		
	Faster RCNN	Faster RCNN+FPN	Faster RCNN+FPN+scSE
Man	43.4	53.3	54.8
Bicycle	24	27.8	24.9
Car	30.9	44.9	43.6
Motorcycle	31	35.1	39.5
Bus	50.6	58.3	60.5
Truck	23.9	33.3	33.4
Traffic Light	17	28.5	24.4
Traffic Mark	49.9	62.7	59
mAP	37.2	44.96	45.8

4.3. Improved model validation and analysis based on deformable convolutions

In this section, the deformable convolutional network in the improved model is validated based on the improved Faster RCNN-ResNet-FPN detection model with the addition of the feature pyramid. The training accuracy and loss map of the improved model based on deformable convolution is shown in Fig. 10.



(a) Acc_train



(b) Loss_train

Fig. 10. The training accuracy and loss of the model

In this subsection, a comparison experiment of eight target detection models is designed: five mainstream target detection algorithms, the benchmark experiment Faster RCNN, Faster RCNN-ResNet50 improved based on FPN, and Faster RCNN-ResNet50-FPN improved based on deformable convolution. The performance of the eight detection algorithms is compared on the COCO dataset, a

comparison of the performance of eight detection algorithms on the COCO dataset is shown in Table 6. In the table, the last row is the result of the improvement made to the Faster RCNN algorithm based on DCN deformable convolution in this subsection. The addition of DCN network to Faster RCNN-ResNet50-FPN increases the AP value from 41.36 to 47.26, and the detection algorithmic performance is significantly improved, so the addition of deformable convolution network can effectively improve the detection algorithm performance. Compared with the YOLO v4 algorithm, which has excellent detection performance, the performance index is better than YOLO v4.

Table 6. COCO data set eight detection algorithm performance comparisons

	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
YOLO v4	CSPDarknet53	47.1	63.1	49.3	27	48.8	56.1
SSD512	ResNet101	35.2	53.6	34.1	6	36.2	48.6
RetinaNet	ResNet101	36.6	59	45.9	22	42.7	51.7
Cascade RCNN	ResNet101	43.7	63.4	46.4	26.5	44.9	56
Faster RCNN	VGG-16	34.2	53.7	34.1	15.8	38	53.5
Faster RCNN+ResNet50	ResNet50	36.93	61.95	39.69	17.44	40.71	50.63
Faster RCNN+ResNet50+FPN	ResNet50	41.36	68.02	44.54	28.06	45.65	64.61
Faster RCNN+ResNet50+FPN+c3-c5DCN	ResNet50	47.26	70.37	46.53	27.8	50.69	66.78

The results of the eight categories of traffic target detection based on deformable convolutional improvement are shown in Table 7. From the table, it can be seen that the scSE attention mechanism improves on each category compared to the baseline Faster RCNN model, and the mAP value of the overall model is improved by 8.36%. The improved Faster RCNN based on FPN and c3-c5 multilayer DCN improves in all the categories compared to the FPN-based scSE attention mechanism, and the mAP value of the overall model improves by 1.2%. Deformable convolution adapts to different detection targets by allowing the convolution kernel to learn an offset at each location point to undergo a shape change, which can effectively improve the model detection performance.

Table 7. The eight kinds of traffic target detection results

Categories	Model frame		
	Faster RCNN	Faster RCNN+FPN	Faster RCNN+FPN+c3-c5 DCN
Man	43.8	51.1	57.9
Bicycle	24.6	26.7	33.2
Car	31.6	39.2	44.9
Motorcycle	30.5	36.1	42.4
Bus	55.7	61.7	65.2
Truck	22.2	29.4	34.1
Traffic Light	18.8	25.7	25.5
Traffic Mark	53.3	69.2	63.1
mAP	35.6	42.76	43.96

5. Conclusion

In the field of traffic safety, image segmentation, as a key technology, is playing an increasingly important role in real-time target detection in complex traffic scenes. The article draws the following conclusions:

In the controlled experiments on the performance of DeepLabv3+ with different backbone networks. In terms of running speed, DeepLabv3+ based on MobileNetV2 backbone network structure is far

ahead of several other backbone networks with a running speed of 51.63fps, from which the real-time performance of the model is good.

In the experiments of eight types of traffic target detection improved based on attention mechanism, feature pyramid and deformable convolution. The performance of the algorithm is improved by 0.34 and 6.73 AP values based on the scSE attention mechanism and using weighted bidirectional feature pyramid network respectively. The Faster RCNN model is improved in each category compared to the baseline Faster RCNN model, and the mAP value of the overall model is improved by 8.36%. This shows the effectiveness of the improved Faster RCNN detection algorithm performance.

References

[1] Ghahremannezhad, H., Shi, H., & Liu, C. (2020). A real time accident detection framework for traffic video analysis. Machine Learning and Data Mining in Pattern Recognition, MLDM, 77-92.

[2] Zhang, X., Hu, S., Zhang, H., & Hu, X. (2016). A real-time multiple vehicle tracking method for traffic congestion identification. KSII Transactions on Internet and Information Systems (TIIS), 10(6), 2483-2503.

[3] Xia, W., Li, P., Huang, H., Li, Q., Yang, T., & Li, Z. (2024). TTD-YOLO: A real-time traffic target detection algorithm based on YOLOV5. IEEE Access.

[4] Wu, Y., Li, Z., Chen, Y., Nai, K., & Yuan, J. (2020). Real-time traffic sign detection and classification towards real traffic scene. Multimedia Tools and Applications, 79, 18201-18219.

[5] Jacob, S., Rekh, A., Manoj, G., & Paul, J. (2018). Smart traffic management system with real time analysis. Int. J. Eng. Technol.(UAE), 7, 348-351.

[6] Janecek, A., Valerio, D., Hummel, K. A., Ricciato, F., & Hlavacs, H. (2015). The cellular network as a sensor: From mobile phone data to real-time road traffic monitoring. IEEE transactions on intelligent transportation systems, 16(5), 2551-2572.

[7] D'Andrea, E., & Marcelloni, F. (2017). Detection of traffic congestion and incidents from GPS trace analysis. Expert Systems with Applications, 73, 43-56.

[8] Fernandez-Sanjurjo, M., Bosquet, B., Mucientes, M., & Brea, V. M. (2019). Real-time visual detection and tracking system for traffic monitoring. Engineering Applications of Artificial Intelligence, 85, 410-420.

[9] Park, S., Han, H., Kim, B. S., Noh, J. H., Chi, J., & Choi, M. J. (2018). Real-time traffic risk detection model using smart mobile device. Sensors, 18(11), 3686.

[10] Barthélemy, J., Verstaevel, N., Forehead, H., & Perez, P. (2019). Edge-computing video analytics for real-time traffic monitoring in a smart city. Sensors, 19(9), 2048.

[11] Nellore, K., & Hancke, G. P. (2016). A survey on urban traffic management system using wireless sensor networks. Sensors, 16(2), 157.

[12] Tippannavar, S., & SD, Y. (2023). Real-time vehicle identi-fication for improving the traffic management system-a review. Journal of Trends in Computer Science and Smart Technology, 5(3), 323-342.

[13] Kashinath, S. A., Mostafa, S. A., Mustapha, A., Mahdin, H., Lim, D., Mahmoud, M. A., ... & Yang, T. J. (2021). Review of data fusion methods for real-time and multi-sensor traffic flow analysis. IEEE Access, 9, 51258-51276.

[14] Kaklij, S. P. (2015). Mining GPS data for traffic congestion detection and prediction. Int. J. Sci. Res, 4(9), 876-880.

[15] Kan, Z., Tang, L., Kwan, M. P., Ren, C., Liu, D., & Li, Q. (2019). Traffic congestion analysis at the turn level using Taxis' GPS trajectory data. Computers, Environment and Urban Systems, 74, 229-243.

[16] Sane, N. H., Patil, D. S., Thakare, S. D., & Rokade, A. V. (2016). Real time vehicle accident detection and tracking using GPS and GSM. International Journal on Recent and Innovation Trends in Computing and Communication, 4(4), 479-482.

[17] Li, S., Li, G., Cheng, Y., & Ran, B. (2020). Urban arterial traffic status detection using cellular data without cellphone GPS information. Transportation research part C: emerging technologies, 114, 446-462.

[18] Gupta, M., Miglani, H., Deo, P., & Barhatte, A. (2023). Real-time traffic control and monitoring. e-Prime-Advances in Electrical Engineering, Electronics and Energy, 5, 100211.

[19] Hilmani, A., Maizate, A., & Hassouni, L. (2020). Automated real-time intelligent traffic control system for smart cities using wireless sensor networks. Wireless Communications and mobile computing, 2020(1), 8841893.

[20] Bidwe, S., Kale, G., & Bidwe, R. (2022). Traffic monitoring system for smart city based on traffic density estimation. Indian Journal of Computer Science and Engineering, 13(5), 1388-1400.

[21] Faisal, F., Das, S. K., Siddique, A. H., Hasan, M., Sabrin, S., Hossain, C. A., & Tong, Z. (2020). Automated traffic detection system based on image processing. Journal of Computer Science and Technology Studies, 2(1), 18-25.

[22] Chen, C., Wang, C., Liu, B., He, C., Cong, L., & Wan, S. (2023). Edge intelligence empowered vehicle detection and image segmentation for autonomous vehicles. IEEE Transactions on Intelligent Transportation Systems, 24(11), 13023-13034.

[23] Gulati, I., & Srinivasan, R. (2019). Image processing in intelligent traffic management. International Journal of Recent Technology and Engineering, 8(2S4), 213-218.

[24] Farooq, M. U., Ahmed, A., Khan, S. M., & Nawaz, M. B. (2021). Estimation of Traffic Occupancy using Image Segmentation. Engineering, Technology and Applied Science Research, 11(4), 7291-7295.

[25] Iftikhar, S., Asim, M., Zhang, Z., Muthanna, A., Chen, J., El-Affendi, M., ... & Abd El-Latif, A. A. (2023). Target detection and recognition for traffic congestion in smart cities using deep learning-enabled UAVs: A review and analysis. Applied sciences, 13(6), 3995.

[26] Song, S., Li, Y., Huang, Q., & Li, G. (2021). A new real-time detection and tracking method in videos for small target traffic signs. Applied Sciences, 11(7), 3061.

[27] Zheng, H., Liu, J., & Ren, X. (2022). Dim target detection method based on deep learning in complex traffic environment. Journal of Grid Computing, 20(1), 8.

[28] Zou, S., Chen, H., Feng, H., Xiao, G., Qin, Z., & Cai, W. (2022). Traffic flow video image recognition and analysis based on multi-target tracking algorithm and deep learning. IEEE Transactions on Intelligent Transportation Systems, 24(8), 8762-8775.

[29] GU, D. Y., LUO, Y. L., & LI, W. C. (2022). Traffic target detection in complex scenes based on improved YOLOv5 algorithm. Journal of Northeastern University (Natural Science), 43(8), 1073.

[30] Lan, J., Jiang, Y., Fan, G., Yu, D., & Zhang, Q. (2016). Real-time automatic obstacle detection method for traffic surveillance in urban traffic. Journal of Signal Processing Systems, 82, 357-371.

[31] Shen, Z., He, Y., Du, X., Yu, J., Wang, H., & Wang, Y. (2024). YCANet: Target detection for complex traffic scenes based on camera-LiDAR fusion. IEEE Sensors Journal, 24(6), 8379-8389.

[32] Zhang, J., Wang, R., Liu, R., Guo, D., Li, B., & Chen, S. (2022). DSP-based traffic target detection for intelligent transportation. IEEE Transactions on Intelligent Transportation Systems, 24(11), 13180-13191.

[33] Jia, J., Guo, W., Zhang, Z., & Cai, Y. (2025). Research on multi-target detection in complex traffic environment based on improved DINO. Journal of Electronic Imaging, 34(1), 013032-013032.

[34] Song, Y., Hong, S., Hu, C., He, P., Tao, L., Tie, Z., & Ding, C. (2023). MEB-YOLO: An Efficient Vehicle Detection Method in Complex Traffic Road Scenes. Computers, Materials & Continua, 75(3).

[35] Liu, Q., Liu, Y., & Lin, D. (2023). Revolutionizing target detection in intelligent traffic systems: YOLOv8-SnakeVision. Electronics, 12(24), 4970.

[36] Weiwei Gao ,Bo Fan ,Yu Fang ,Mingtao Shan& Haifeng Zhang. (2024). Lightweight and real-time semantic segmentation of UAV traffic videos based on siamese network for keyframe recognition. Multimedia Tools and Applications,(prepublish),1-19.

[37] Van Thong Huynh,Soo-Hyung Kim,Gueesang Lee & Hyung-Jeong Yang. (2019). Eye Semantic Segmentation with a Lightweight Model..CoRR,abs/1911.01049.

[38] Chuan Li,Nianbiao Cai,Tong Pu,Xi Yang,Hao Liu & Lulu Wang. (2025). Rebar Recognition Using Multi-Hyperbolic Attention in Faster R-CNN. Applied Sciences,15(1),367-367.

[39] Qinghua Zhao & Yaqiu Liu. (2024). Design of apple recognition model based on improved deep learning object detection framework Faster-RCNN. Advances in Continuous and Discrete Models,2024(1),49-49.