Image semantic segmentation with an improved fully convolutional network

Kuo-Kun Tseng[1]

[1] School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China

Haichuan Sun[1]

[1] School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China

Junwu Liu[1]

[1] School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China

Jiaqi Li[1]

[1] School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, China

K. L. Yung[2]

[2] Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Kowloon, Hong Kong, China

W. H. Ip[2,3]

[2] Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Kowloon, Hong Kong, China

[3] Department of Mechanical Engineering, University of Saskatchewan, Saskatoon, Canada

Abstract

With the development of deep learning and the emergence of unmanned driving, fully convolutional networks are a feasible and effective for image semantic segmentation. DeepLab is an algorithm based on the fully convolutional networks. However, DeepLab algorithm still has room for improvement, and we design three improved methods: (1) the global context structure module, (2) highly efficient decoder module, and (3) multi-scale feature fusion module. The experimental results show that the three improved methods that we proposed in this paper can make the model obtain more expressive features and improve the accuracy of the algorithm. At the same time, we do some experiments on the Cityscapes dataset to further verify robustness and effectiveness of the improved algorithm. Finally, the improved algorithm is applied to the actual scene and has certain practical value.

1 Introduction

In recent years, with the rise of deep learning and the development of computer hardware, the field of computer vision has seen an explosive progress. Before that, due to the low accuracy of image algorithm and poor real-time performance, the research results of machine vision can only be applied to simple scenes. For complex scenes, although machine vision has a lot of information that other sensors do not have, it cannot be applied in practice due to its low accuracy. With the breakthrough of deep learning, artificial intelligence, and other fields, machine vision has the advantages of low-cost camera and large amount of information, so its development potential is self-evident.

Driverless technology (Yang 2014) is one of the most popular technologies in the field of artificial intelligence, and now unmanned vehicles are mainly used in military and industrial production. The traditional driverless technology mainly relies on high-precision sensors such as laser radar to sense the environment. Consider Google's driverless technology as an example, which can run safely in real time.

However, because of the high cost of laser radar, it is difficult to popularize driverless technology. But with the development of deep learning (Sun et al. 2012) and the emergence of fully convolutional networks, it is possible to use the cheap visual sensors to sense the environment.

At the same time, the image semantic segmentation algorithm (Wei and Zhao 2016) is the core algorithm of driverless technology. The general process is to input the images obtained by vehicle camera or other sensors into the convolutional neural network, and then background computer can automatically segment the entire picture, so that the smart car can avoid obstacles. Compared with object detection algorithm, image semantic segmentation algorithm is more practical in driverless technology; it can provide more spatial information.

Recently, image semantic segmentation has been an active research area. Many public datasets have been provided (Yu et al. 2018) and more and more algorithms (Liu et al. 2018) have been designed, but there is still a lack of improvement in terms of the accuracy and speed of the algorithm. Thus, the following is our motivation for this research:

1. Resrach on image semantics segmentation can provide more information to the pilotless system, which makes this application more feasible.
2. It is necessary to develop a better precision algorithm to make image semantics segmentation workable.

Since the birth of fully convolutional neural network, researchers have been innovating on the basis of it and have produced many excellent results. The DeepLab algorithm series is a relatively successful series of algorithms. When DeepLab algorithm is used for image semantic segmentation, it lacks prior information in complex environment, so it cannot avoid misjudgment. In the decoder part, the DeepLab algorithm directly uses bilinear interpolation to decode, thus losing a lot of edge information, and the output of the original algorithm is rough. The image semantic segmentation algorithm based on deep learning began to rise in 2015. It is a relatively new and rapidly developing research topic. This paper aims to study and improve image semantic segmentation by ana- lyzing recent image semantic segmentation algorithms based on deep learning algorithm. Improvements are made based on the DeepLabv2-ASPP algorithm and the DeepLabv2-largeFOV algorithm to improve the accuracy of the algorithm.

The research contribution of this paper includes the following aspects:

1. A global context structure is introduced to provide a part of the prior information for the algorithm, in order to improve the accuracy of the image semantic segmentation algorithm.
2. An effective decoder module is designed, which combines shallow semantic features with deep semantic features to recover the edge information of some objects to improve the accuracy of the algorithm.
3. The multi-scale feature fusion method is designed and introduces the additional supervision module into the multi-scale feature fusion algorithm to improve the accuracy of the algorithm.
4. The multi-reference model and multi-parameter cross-experiment verification of the improvement points proposed in this work were carried out, and the improved methods were analyzed and discussed. The idea of migration learning was used to conduct experiments on other types of data, which further proves robustness and effectiveness of the improved method.

For the sake of easy reading, we have arranged the abbreviations of this article in Table 1.

The rest of the structure of this paper is listed as follows. Section 2 describes the related work this research, and Sect. 3 presents our improved algorithm and framework. In Sect. 4, the experiments are carried out to prove the effectiveness of proposed method. Finally, our conclusion and future work are stated in last section.

2 Related work

Convolutional neural network (CNN) is the most widely used deep learning model in the field of image processing. Back propagation algorithm was proposed by Rumelhart et al. (1980), which lays the foundation for the development of neural network. In 1998, some papers were published by Lecun et al. (1998), which established the structure of CNN. The major breakthrough is to design a layered artificial neural network, called LeNet-5. It uses back propagation algorithm and stochastic gradient descent algorithm to optimize the model to classify handwritten digits. LeNet-5 has become an important milestone of CNN. However, since there was no large-scale dataset by that time, with the number of neural network layers increasing, it was inevitable in LeNet-5 that overfitting would occur. Meanwhile, there were not enough computing resources, so the training time was too long and inference time is too slow when compared to other machine learning algorithms. Hinton et al. (2006) found that training complexity can be effectively alleviated by training layer by

Table 1 Abbreviation

Abbreviation	Meaning
CNN	Convolutional neural network
MIoU	Mean intersection over union
PA	Pixel accuracy
VGG	Visual geometry group
BN	Batch normalization
CRF	Conditional random field
ASPP	Atrous Spatial Pyramid Pooling
DenseNet	Dense convolutional network
BN	Batch normalization
VOC	Visual object classes
FCN	Fully convolutional network
ResNet	Residual network
SPP	Spatial pyramid pooling

layer, which successfully solved the speed issue for training neural network.

With the development of GPU technology, neural networks have regained the attention from academia and industry sectors. Krizhevsky et al. (2012) proposed AlexNet that won the first prize from the ImageNet competition, and its error rate decreases largely by about 10% when compared to the second player. AlexNet network replaces the traditional Sigmoid function with the Relu (Glorot et al. 2011) to alleviate the gradient-vanishing problem. At the same time, it uses the dropout (Srivastava et al. 2014) regularization technique to improve the robustness of the algorithm and prevent overfitting.

Since CNNs have begun to be valued, many frameworks have been developed. In 2014, the VGG-Net structure which is simple and effective was proposed by Simonyan and Zisserman (2014). The first few layers use a 3 9 3 convolution kernel instead of a large convolution kernel, so that the receptive field of each layer is the same. At the same time, the depth of the network is increased to obtain more nonlinear expressions. The size of the feature map is reduced by using the maximum pooling method. The last three layers are two fully connected layers and one softmax layer. In that ImageNet competition, VGG-Net achieved good results. In the same year, Szegedy et al. (2015) proposed GoogLeNet. They first proposed a concept called Inception in GoogLeNet. GoogLeNet uses 1 9 1, 3 9 3, and 5 9 5 convolution kernels, respectively, to extract the feature map and learn the image information of different scales, and finally connects these convolution outputs as the input of the next layer. While increasing the depth of the network, it also increases the width of the network, reduces the parameters of the model, and improves the generalization ability of the model. All in all, it has achieved good results.

He et al. (2016) proposed ResNet. ResNet introduced a residual learning module and the depth of ResNet is 152 layers. ResNet achieved the first place in the ImageNet image dataset. The residual learning module contains multiple convolutional layers designed to add module inputs directly to the module's output by adding a hop connection. The emergence of this module effectively alleviates the problem of gradient disappearance caused by layer deepening, so that the network can still be trained normally after the model has reached more than 100 layers.

At the same time, the idea of batch normalization (BN) proposed by Ioffe and Szegedy (2015) is simple, and its effect is very obvious, which greatly improves the accuracy and convergence speed of convolutional neural networks.

The essence of deep learning is that the convolutional neural network can extract the features of objects well and is also a feature engineering in traditional algorithms. Therefore, the research on convolutional neural networks has never stopped. In 2017, Google proposed an optimized version of Inception for Xception (Chollet 2017). It introduces a residual module and uses a deep separable convolution layer. Compared to the original Inception module, the effect has been improved.

In the same year, Huang et al. (2017) proposed a novel network structure, DenseNet (dense convolutional network). The traditional network structure is either deep (such as ResNet) or wide (such as GoogLeNet), and DenseNet takes advantage of the features of each layer as much as possible. The basic idea is to connect all the layers while maintaining the structure of the convolutional neural network. In other words, the input of each layer contains the output of all the previous layers, so as to minimize the loss of shallow features. At the same time, the network structure is optimized. The number of feature map of the output from each convolution layer is very small, which guarantees less parameters and fast convergence. The paper also won the best paper of CVPR 2017.

At the same time, Wu and He (2018) proposed group normalization by replacing batch normalization, so that the normalization operation is no longer affected by the batch size, which reduces the hardware requirements of deep learning. In the traditional image semantic segmentation, the traditional methods such as NC algorithm are used to segment the image by the relationship between the pixels in the image. At the same time, the feature extraction mainly uses image features such as texture features, color features, and HOG (Dalal and Triggs 2005). Using SVM (Pai-Hsuen et al. 2005), random forest (Criminisi et al. 2012), AdaBoost (Lienhart and Maydt 2002), and other classifier algorithms for classification, the traditional algorithm has a poor prediction effect. With the rise of convolutional neural networks, the development of hardware technologies, and the release of large-scale data sets, researchers have applied deep learning techniques to the field of image semantic segmentation. Professor Li Feifei of Stanford University has created a large dataset ImageNet (Russakovsky et al. 2015), which is a dataset with 1000 objects for a variety of machine vision tasks. In addition, major research organizations have released various types of data sets, such as Pascal VOC (Everingham et al. 2010), Mircosoft COCO (Lin et al. 2014), KITTI (Geiger 2012), and Cityscapes (Cordts et al. 2016).

The current image semantic segmentation field is similar to the field of object detection based on convolutional neural networks. One focuses on improving accuracy, and the representative algorithms are FCN, SegNet, DeepLab, and so on. The other is mainly focused on speed, representing algorithms such as ENet. In the following article, we will investigate the two research directions separately. Most of the current image semantics segmentation frameworks are based on fully convolutional network (FCN). In

2014, the FCN algorithm was proposed by Long et al. (2015). The central idea is to directly segment the end-to-end image semantics at the pixel level. Its pioneering idea is to replace the traditional full connection layer fc6 and fc7 with the convolution layer, while the final fc8 is replaced with a 21-channel one, and there are 21 classifications in the VOC data set, and the convolution layer is the final output of the network. At the same time, the transposed convolution layer is used for upsampling to obtain the final result.

Badrinarayanan et al. (2017) proposed SegNet algorithm which has an encoder–decoder framework. The innovation of this framework is that when the encoder is pooled, the pooled layer index is established. When the decoder is decoding the pooled layer index is used for upsampling operation. SegNet does not need learning, and the boundary information is recovered better while reducing the training parameters.

Then the DeepLab series are coming, which is well known in the field of image semantic segmentation. In 2014, DeepLabv1 (Chen et al. 2014) added a fully connected conditional random field (CRF) based on the original full convolutional network to optimize the boundary (Krähenbühl and Koltun 2011). DeepLabv2 (Chen et al. 2018a) was proposed in 2016. On the basis of DeepLabv1, dilated convolution was added to reduce the downsampling operation. At the same time, multiple dilated convolutions (Atrous Spatial Pyramid Pooling, ASPP) were extracted in parallel with different sampling rates, which greatly improved the accuracy. On the basis of DeepLabv2, DeepLabv3 (Chen et al. 2017) proposed in 2017 continues to optimize the ASPP structure and continues to cascade multiple dilated convolution structures after the improvement in residual network to further extract features with strong representation. DeepLabv3? (Chen et al. 2018b) which is based on DeepLabv3 was proposed in 2018. DeepLabv3? optimizes the decoder part, designs a novel decoder structure, uses the Xception network structure mentioned above as the reference model, and uses more data sets in training. As the result, DeepLabv3? makes the final result more accurate. At present, it is the first in Pascal VOC 2012.

Zhao et al. (2017a) analyzed the current mainstream algorithms and found that most of the algorithms do not use the overall prior scene information, so it is easy to misjudge the problem, and it constructs the pyramid pooling module. The main idea is to make full use of global context information through context aggregation based on different regions and use global prior information to effectively improve the quality of scene resolution tasks. In multi-scale feature fusion, Kong et al. (2016) proposed a downsampling of shallow convolution features, the middle convolution feature remains unchanged, and the deep convolution feature be transposed convolution. After the normalization process, the merge operation is performed. The fused feature is used for object detection. Lin et al. (2017a) believes that the characteristics of each layer of the convolution layer are helpful to the final result, so it designs a RefineNet module that can combine deep features with shallow features, which is a complex In the fusion mode. The feature map of the final model output contains all the convolutional layer information, so the higher precision is obtained. Wang et al. (2017) innovatively introduced the superpixel concept into the decoder module. At the same time, the author also found that the dilated convolution is prone to the ''grid'' phenomenon, and the problem is solved by setting dilated convolution with different synchronization, which greatly improves the accuracy of the algorithm.

Focusing on real-time algorithms, Romera et al. (2017) proposes a new module that uses a bottleneck module similar to the ResNet network, which includes a series of downsampling operations, dimensionality reduction, and dilated convolution. The algorithm is very real time, but since the structure design is very simple, more detailed information is lost, and the accuracy of the algorithm is not very good. Dvornik et al. (2017) proposed that BlitzNet combines image semantic segmentation and object detection algorithms for joint training, and integrates the real-time object detection algorithm SSD algorithm (Liu et al. 2016) into image semantic segmentation to achieve the mutual promotion effect. Chaurasia and Culurciello (2017) proposed LinkNet to design a new deep neural network structure. The idea is to add the corresponding features of the encoder to the corresponding decoder features and design a new bottleneck structure, which makes it easier to obtain the information.

Finally, some new algorithms focus on multi-function algorithms and real-time acceleration, such as SSOD (Salscheider 2019) can use a network for semantic segmentation and object detection, and DFANet (Li et al. 2019) for network optimization to achieve real-time semantic segmentation.

3 Proposed networks

This section describes an improved the methods from the DeepLab algorithm. Firstly, some architectures and techniques used in the DeepLab algorithm are introduced. Then the overall framework of the improved algorithm is given. It also introduces the specific implementation of the algorithm in the two benchmark networks and with the fine-tuning strategy of training. Finally, the three improvements of this work are introduced in detail: global context structure, decoder module, and multi-scale feature fusion framework.

3.1 Improved fully convolutional network algorithm

3.1.1 Fully convolutional network algorithm

DeepLab image semantic segmentation algorithm is one of the most accurate algorithms. The algorithm is improved on the basis of the FCN algorithm, which introduces a dilated convolution in the encoder module, which expands the receptive field and reduces the downsampling operation. Therefore, DeepLab retains some details. The decoder module enables bilinear interpolation algorithms. The postprocessing part uses the fully connected conditional random field for boundary optimization. It designs two models. One is DeepLab-large FOV based on VGG16 network, and the other is DeepLab-ASPP based on residual network. The overall structure based on the residual network is shown in Fig. 1.

This work uses two DeepLab algorithms as the basis algorithm for improvement. The DeepLab algorithm proposes some problems in the image semantic segmentation algorithm and gives corresponding solutions.

1. The problem of reduced resolution of the feature map. In order to aggregate feature information better, the existing network structure includes a downsampling operation such as a pooling layer. These operations decrease the resolution of the feature and double the number of channels, so that most of the information is stored in multiple channels. The output requires a scale of the original image size, so the decoder module is required to perform the upsampling operation, and the lost details are unrecoverable, so the author removes downsampling operation of the last few layers of the pooling layer, and replaced the convolution layer of the latter layer with dilated convolution. As a result, the downsampling rate changed from 1/32 to 1/8, retaining the details.
2. Postprocessing using fully connected conditional random fields. Since the downsampling operation will lose some of the details, the edge of the prediction result is poor. Therefore, the author uses the fully connected conditional random field algorithm to correct the edge of the object.
3. Feature learning. The author designed the LargeFOV structure and the ASPP structure. It is possible to obtain as many multi-scale global features as possible without increasing the amount of calculation. Referring to the idea of SPP and combining with dilated convolution, the author uses different sampling rates to sample feature maps.

3.1.2 Proposed improved framework

Through the description in the previous section, it can be seen that deep learning can be regarded as feature engineering. Therefore, the expressive ability of model extraction features will directly affect the accuracy of image semantic segmentation algorithms. This work uses two classic network structures as the benchmark model, including VGG network and residual network. Improvements based on the benchmark model are designed to capture more expressive features. In this paper, the following three improvements have been made:

1. Take advantage of the global context information of the image. The global context information of an image is relative to the local information of the image. The a priori information of the image is not used in the benchmark algorithm. This leads to more misjudgment if the image is more complex or the two types of objects are similar. Therefore, the global context module is introduced into the benchmark model to extract more priori information, and then the improved algorithm will improve the accuracy of the algorithm.
2. Design an efficient and compact decoder module. The function of the decoder module is to restore the information output by the encoder to the size of the original image through a series of operations. The current benchmark algorithm uses a bilinear interpolation to the original image size at the decoder module. The original decoder module inevitably loses some of the details. So this paper improves the decoder module. The original algorithm only uses the last layer of convolutional layer features. This paper combines the features of different convolution layers, combines the shallow semantic information with the deep semantic information, adjusts the proportion of the two when

merging, and makes full use of the characteristics of different convolution layers. Therefore, this paper improves the ability of feature expression and then restores the details of the object.

3. Optimize the multi-scale model framework. The same model has different ‘‘understandings’’ for different scales of images and ultimately obtains different responses. Therefore, using the idea of ensemble learning, this paper designs two multi-scale feature-level fusion methods, and on this basis, introduces additional supervisory module to improve the accuracy of the algorithm. The proposed improved framework is shown in Fig. 2.

3.2 Proposed improved algorithms in network architecture

3.2.1 VGG network

VGG network won the second place in ILSVRC image competition in 2014. Many of its design ideas were adopted by later network structures. Its design is very concise and the network structure is very consistent. All convolutional layers were 3 9 3. The convolution kernel has a step size of 1, and the pooling layer downsamples with a step size of 2. After the pooling layer, the number of convolution kernels is doubled. The 3 9 3 size receptive field is the smallest size that can capture the neighborhood information (upper and lower, left and right, center). According to the nature of convolution, the neighborhood range of two convolution layers with 3 9 3 size can be equivalent to a 5 9 5 convolution core. Replacing the convolution kernel with a large number of small convolution kernels reduces the amount of parameters that need to be trained and also increases the nonlinear expression, which can be better to fit the data set.

Compared with another classic network structure, GoogLeNet, the VGG network has better performance in many migration learning tasks. If the downsampling operation is used for differentiation, the VGG network can be divided into five sets of convolutional layers, the last three layers are fully connected layers, and the groups are separated by pooling layer. The number of convolution kernels in each group doubles, compared to the number of convolution kernels in the previous group. The benchmark network used in this paper combines the dilated convolution, as shown in Fig. 3.

Because the benchmark algorithm takes VGG network as the basic network model and makes a series of improvements on the basis of VGG network, firstly, the step size of pool4 and pool5 is changed to 1, so the pool layer not only acts as aggregating region information, but also has no downsampling operation. Replace conv5_x with a dilated convolution of step size of 2, then replace the fully concatenated layer fc7 with a convolutional layer of 1024 1 9 1 convolution kernels, and replace fc8 with a convolutional layer that has num_class 1x1 convolution kernels. The parameters in the network layer are reduced, and the fc6 layer is replaced by 3 9 3 dilated convolution with a step size of 12. This paper takes the optimized VGG16 network as the benchmark network to continue to improve. In order to further reduce the amount of parameters, firstly, the fc7 layer is removed, and the global context structure is connected after conv5_x to obtain the overall prior information of the image, complete the components of the encoder structure, and then connect the decoder module.

In the parameter fine-tuning, since the features of conv2_x and conv3_x are used in the decoder module, all parameters are updated. Other experiments are to fix the parameters of the first three convolution layers of VGG network (conv1_x–conv3_x layer), only fine-tuning the parameters of conv4_x–conv5_x layer, the newly added global context module, the decoder module and the newly added BN layer.

3.2.2 Residual network

Before the emergence of residual network, few network structures can break through dozens of layers. The reason is that with the deepening of depth, back propagation algorithm is more and more difficult to achieve, and it is very easy to appear the problem of gradient disappearance. The introduction of the residual network makes the depth of the network structure greatly deepened. As the depth deepens, the features learned by the convolutional neural network become more and more abstract, and the distinguishing ability becomes stronger and stronger. The residual network mainly designs a module. The main idea: Define the underlying mapping HðxÞ, and adapt the superposed nonlinear layer to another map FðxÞ :¼ HðxÞ − x. So the original feature map was modified to FðxÞ þ x. This entire process can be represented by the following formula.

$$y = F(x, \{w_i\}) + x \tag{1}$$

x and y, respectively, represent the input and output of the convolution layer. Function Fðx; fwigÞ represents learnable residual mappings FðxÞ þ x can be achieved by using shortcut connections.

In the ResNet network, there are two types of shortcut connections as shown in Fig. 4. Figure 4a is mainly used in ResNet networks with less convolution layers, such as ResNet34, and Fig. 4b is mainly used in networks with more convolution layers, such as ResNet50 and ResNet101.

If the downsampling operation is a dividing line, ResNet can also be divided into five groups of convolution layers. The last layer is a fully connected layer. The convolution layer with a step of 2 is used to downsample between groups, and the number of convolution kernels in each group is twice as large as that in the previous layer. The ResNet network structure in this paper is different from traditional ResNet, as shown in Fig. 5.

In the previous research, we can know that the deeper the network structure, the more generally it has stronger representation ability. At the same time, since deep learning is a process which is similar to building blocks, the network structure can be replaced with each other, so the VGG network structure can be replaced with a deeper ResNet network structure. Similarly, DeepLab has also made some optimization in ResNet. Firstly, all convolution layers in conv4_x are replaced by dilated convolution with convolution core of 3 9 3 and step size of 2. At the same time, all convolution layers in conv5_x are replaced by dilated convolution with convolution core of 3 9 3 and step size of 4. Therefore, assume that the input image is 320 9 320 9 3, after the improved ResNet extraction feature, the output feature size is 40 9 40 9 num_class, which reduces the number of downsampling. This work is based on the improved ResNet network structure and continues to optimize.

Connect a global context information structure after conv5_x, and then connect the ASPP structure of the algorithm itself, and connect the decoder module after the ASPP structure. When the parameters are fine-tuning, since the shallow semantic information is used in the decoder module, in this experiment, only batch normalization layer parameters are fixed, and all other network layer parameters are updated. In other experiments, the parameters of the first three convolution layers of the residual network are fixed, namely fix the parameters of the conv1_x–conv3_x layer and all the batch normalization layer parameters, only fine-tune the parameters of conv4_x–conv5_x layer, the fully connected layer, the ASPP structure, the newly added global context module, the decoder module, and the newly added BN layer.

3.3 Other improvement strategies

3.3.1 Improved context structure

The context information in this paper refers to global information, which can make the model obtain more accurate scene perception, and improve the discriminative accuracy of the model. The accuracy of the final result depends greatly on the prior information of the picture scene. For the typical complex scene understanding, in order to obtain the global image level features, the spatial pyramid is widely used, and the spatial statistics provides a good proof for the overall scene interpretation. The spatial pyramid pooling network further enhances the ability of global scene understanding. Content relationships are universal and important, especially for the understanding of complex scenes. For example, an airplane may be on the runway or in the air, there may be trains on the rails, and the car may not be in the sea. In the field of object detection, the context information is a local feature extracted from the candidate region and a global feature of the entire image. In the field of image semantic segmentation, it refers to the global scene category clues of the global scene of the image.

This paper investigates the global context structure in the field of image semantic segmentation. Liu et al. (2015) proves the importance of global features, using pooling to obtain global features. At the same time, when layers and layers are joined together, since the size of each layer is different, the weights of each layer are in different ranges, so the effect of direct joining is general. The author proposes using L2 norm to normalize the convolution layer. Mehta et al. (2018) introduced spatial pyramid pooling to obtain global features, multi-scale extraction of context information, and achieved good results.

The main function of the global context structure is to take the global information of the picture as the feature, which plays a certain role in determining the final pixel category. This paper mainly conducts a series of experiments based on the literature (Liu et al. 2015; Mehta et al. 2018) and finally designs the global context structure of this paper.

As shown in Fig. 6, taking the VGG network as an example, the conv5_x of the original algorithm is used to extract image features, and then the LargeFOV structure of fc6 layer is connected to extract features. The global context information module is added after the conv5_x in this paper, and the size of the input image is 321 9 321 9 3. After extracting features through the residual network, the size of the feature image becomes 1/8 of the original image, the final conv5_x output feature size is 41 9 41 9 512. Using the mean pooling on overall output feature of conv5_x layer, the output will become 1 9 1 9 512, and then we will upsample it to 41 9 41 9 512 by using bilinear interpolation, and finally merge it with the feature map of conv5_x to get the feature map with size 41 9 41 9 1024. At the same time, in order to avoid to affect the existing structure of the model and using the trained parameters. Convolution kernels with size 512 1 9 1 are used for the dimensionality reduction operation. The feature map is the output feature map of the global context module.

After the network structure is replaced by residual network, the output of conv5_x of residual network is 41 9 41 9 2048. After pooling, perform bilinear interpolation, and merge it with the original feature image to become 41 9 41 9 4096. In order to use the parameters of the original model, 512 1 9 1 convolution kernels are used for dimensionality reduction to restore the original size, and this feature map already contains global context information.

The module introduced above is the best global context module obtained through a series of experiments. The variables of this subject are as follows: (1) Two pooling methods are selected: maximum pooling and mean pooling. (2) The global context information of different scales is selected at the same time, which is 1 9 1, 2 9 2, and 3 9 3. In view of the above two kinds of variable, this paper designs and practices a series of two-variable crossover experiments. It can be clearly seen that the global context module can effectively improve the accuracy of the algorithm, while the global context module composed of different variables has different accuracy for the algorithm. The specific experimental details are introduced in Part V.

3.3.2 Improved decoder architecture

A series of features are extracted from the encoder part, and the receptive field is enlarged by introducing dilated convolution, which reduces 1/32 downsampling to 1/8 in the original network structure and retains some details. However, if a simple decoder module such as bilinear interpolation or deconvolution is directly used, the details of object segmentation cannot be restored, resulting in rough boundaries. Therefore, this paper designs a decoder module to improve the accuracy. Firstly, a series of investigations were conducted on the current decoder module.

Badrinarayanan et al. (2017) records the index position of the pooling layer in the image semantics segmentation algorithm and restores part of the information of the pooling layer in the decoder module, which optimizes the result to a certain extent. In Chen et al. (2018b), a simple decoder module was designed, which merges four times upsampling and shallow semantics, connects two convolutional layers, then upsamples to the original image size, and finally obtains better results. Chaurasia and Culurciello (2017) proposed a real-time image semantics segmentation algorithm. In order not to affect the running speed, the encoder information is added to the corresponding decoder information. This design requires that the size and shape of the encoder and the corresponding decoder are identical. Ronneberger et al. (2015) combines shallow features with deep features. Wang et al. (2017) and Ghiasi and Fowlkes (2016) introduces the idea of superpixel, designs a complex decoder module, and achieves good results.

In this section, we investigate the decoder modules in the above-related studies and found that all decoder modules use bilinear interpolation as the basic module of the decoder module. Therefore, after considering the advantages of various types of decoder structure and the limitation of hardware resources, we design our own decoder module through a series of experiments, hoping to restore more details, and improve the accuracy of the algorithm. The concrete frame diagram is shown in Fig. 7.

As shown in the figure, taking the VGG network as an example, if the input image size is 321 9 321 9 3, the encoder output is fc7 layer, and the corresponding size is 41 9 41 9 1024. If the result is output directly, according to the nature of the full convolution neural network, the dimension of fc7 layer is reduced to 41 9 41 9 num_label by using the 1 9 1 conv layer, and then bilinear interpolation is used to change the size from 41 9 41 9 num_label to 321 9 321 9 21 by using upsampling method. The original model has a simple decoder module that does not make efficient use of shallow semantic information. Therefore, the subject redesigned the decoder module based on the original decoder model. For the VGG network, the fc7 layer is first reduced to 256 layers, and the output feature map is called deep semantic. At the same time, for shallow semantics, we will use conv2_x and conv3_x, respectively. The size of these two layers is four times and two times the size of the output layer, respectively. Therefore, we firstly upsampled the deep semantics that have reduced dimension twice, and Conv3_x reduces the dimension to 32 layers and then merges both of them. Since the shallow semantics contain certain position information, it helps the boundary optimization and controls the number of convolution kernels of shallow semantics and deep semantics to make the ratio that is 1:8 stable.

As the result of this design, the image semantics plays a major role in the final judgment of the classification, while using shallow semantics to recover some of the detail information, and then 256 convolution kernels are connected, so that it can refine these features for 3 9 3 convolutional layer. Similarly, the combined convolutional layer is upsampled twice, combines with conv2_x, then be connected to a convolutional layer, and finally the result is output. For the above newly added convolutional layer, a bn (batch normalization) layer is added to conduct normalization. To make it converge faster, connect the Relu activation function. For the residual network, since the residual network conv1_x conducts the downsampling operation, the process is slightly different from VGG. Firstly, the conv2_x is merged with the deep semantic features, then upsampled, and finally merged with conv1_x to complete the decoder module.

The decoder module described above is the best-performing decoder module obtained through a series of experiments. In this module, there are two main variables: (1) The number of convolution kernels of conv2_x (corresponding to conv1_x in residual network) that needs to reduce the number of dimension. (2) The number of convolution kernels of conv3_x (corresponding to conv2_x in residual network) that needs to reduce the number of dimension. In view of the above variables, this paper designs a series of crossover experiments, which can clearly show that the algorithm accuracy has been greatly improved after adding decoder module, and different combinations of variables will get different results. The specific experimental details are introduced in Part V.

3.3.3 Improved multi-scale feature fusion framework

According to the experience of traditional machine learning algorithm, if we want to improve the accuracy, we can not only optimize the model itself, but also integrate different models, called ensemble learning. Classical algorithms such as xgboost and random forest in traditional algorithms use the idea of integrated learning. There are two types of ensemble learning. One is to extract different features and finally integrate into the same model for training, called multi-feature fusion. The other is that the same features are trained and integrated with different algorithms. The most commonly used method of integration is voting or weighting. Ensemble learning often gets relatively good results, because different models have different learning methods and acquired different characteristics, so there will be good results in combination. Generally speaking, the greater the difference of the ensemble algorithm has, the better the effect of the ensemble model has.

For the field of image semantic segmentation, the first way to integrate is to use different network structures, such as VGG-Net, ResNet, GoogLeNet, and DenseNet, and use the method of vote or weight to get the final result. This method needs to train a variety of network structures and uses the differences between different network structures to extract the characteristics of different focuses. As the result, we should get a good result, but this method is extremely time-consuming. At the same time, by reading relevant studies, many researchers have adopted an integrated method for multi-scale prediction to improve the accuracy of algorithms in the field of image semantic segmentation and even object detection. When training models, due to batch processing problems, all input images are transformed to a specified size, so these learned parameters will be excessively coupled to the current data set, making it difficult for the model to have good results on other scale images. In order to solve this problem and increase the robustness of the model, multiple prediction results are generated by using different scales of the target, and finally the results are integrated to produce optimal results. This process is multi-scale prediction, and the research focus of this paper is How to combine these results or how to combine features acquired at multiple scales.

Firstly, a series of investigations have been made on multi-scale prediction. Multi-scale model fusion can be roughly divided into two ways. One is feature-level fusion. For example, Eigen and Fergus (2015) proposed a multi-scale prediction structure. The lower-resolution output results are upsampled to a higher-resolution image resolution and spliced with higher-resolution inputs to output the final result. It is worth noting that the paper uses multiple loss functions to get better result, which are loss function of depth estimation, normal vector estimation loss function, and multi-class loss function of image semantic segmentation. Zhao et al. (2017b) is a real-time image semantics segmentation framework. The author designs a multi-scale cascade framework, which uses lower-resolution output to modify higher-resolution output. The other is the result-level fusion method, such as the attention mechanism proposed in Chen et al. (2016). The author thinks that the weights of different scale prediction results are different for the final results. Therefore, the author designed a learnable weight to weigh the prediction results of different scales, and finally achieved good results.

In this paper, feature-level fusion strategy is adopted. Firstly, the overall block diagram of multi-scale feature fusion framework is designed.

As shown in Fig. 8, no matter how many scales are used for input, network parameters will be shared. The part of the dotted line in the figure is the parameter sharing part, but in practice, it is necessary to optimize the parameters of the network by using multiple back propagation, so the training is time-consuming.

At the same time, in order to optimize the results better, the idea of auxiliary classifier is introduced in this paper. The central idea is to add some supervisory information in order to help the model fit the data set better, which was first proposed in Lee et al. (2014). In the literature, auxiliary classifier is added in each layer of convolution layer. In Zhao et al. (2017a), the idea of auxiliary classifier is introduced, and the output loss of each classifier is weighted. Chen et al. (2016) also adds the idea of auxiliary classifiers, and the precision has been greatly improved. In this paper, the idea of auxiliary classifier is introduced, and an additional supervisory module is added to the feature fusion module. At the same time, this paper designs two simple and effective feature fusion modules. The first one uses the merging method. The specific details are shown in Fig. 9.

As shown in Fig. 9, firstly, three kinds of resolution images are input by parameter sharing, and three kinds of feature maps with different sizes which are defined as scale 1 feature map, scale 2 feature map, and scale 3 feature map are generated. Among them, the scale 1 feature map is the largest, so in the feature fusion module, we upsample the other two feature maps to the size of the scale 1 feature map and merge the three generated feature maps. And we calculated a loss independently for each feature map, and in the Fig. 9, they are represented by loss1, loss2, loss3, and loss4. The total loss value is expressed by Eq. (2).

$$\text{loss} = \sum_{i=1}^{4} \text{loss}_i. \qquad (2)$$

Since different models have certain differences, the output feature map size and number of layers are also different. If the number of layers is large, 256 or 512 1 9 1 convolution kernels are generally used to reduce the dimension, reduce the number of parameters, accelerate the convergence speed, and then conduct feature fusion. At the same time, we also tried to weight the different output loss, but the effect is not very obvious, and we need to continue experimenting to find the optimal weight corresponding to each loss. The experimental period is longer, so we did not use the weighting method in the end.

In the second way, referring to the way of feature-merging in different layers in the classical object detection paper FPN algorithm (Lin et al. 2017b), this paper migrates this feature-merging method to the multi-scale feature-merging module and makes some changes to it. At the same time, the auxiliary classifier idea is still introduced into the feature fusion module. The specific implementation details are shown in Fig. 10.

As shown in Fig. 10, the dotted line part represents the specific details of the feature fusion of the module. First, it is assumed that the F1 input is a feature map with a smaller resolution, and the F2 input is a feature map with a larger resolution. Since the module adopts summation operation, so to make the size and shape of the two identical, for F1, first use the upsampling operation to the size of F2, then use 256 convolution kernels with a convolution kernel size of 3 9 3 to extract features. For F2, use 256 convolution kernels with a convolution kernel size of 1 9 1 for dimensionality reduction, then add both of them and then use the Relu function to finally get the result, which combines the information of F1 and F2. Suppose there are three scale inputs, Scale1, Scale2, and Scale3, which are sequentially reduced in size. The inputs (F1, F2) in the figure can produce two groups (Scale3, Scale2) and (Scale3_Scale2_SUM, Scale1), after a series of feature fusions. The final result contains all the characteristic information entered by the three scales. At the same time, an additional supervision module is introduced. Compared to feature fusion module A, feature fusion module B will generate three losses, including Scale3, Scale3_Scale2_SUM (referred to as Scale32) and Scale32_Scale1_SUM. The losses are expressed as loss1, loss2, and loss3, respectively. Then the total loss size can be expressed as Eq. (3).

$$\text{loss} = \sum_{i=1}^{3} \text{loss}_i \qquad (3)$$

There are three main variables in the multi-scale feature fusion framework proposed in this section IV: (1) multi-scale input combinations, (2) feature fusion module, and (3) whether to use an additional supervision module. For the above variables, this paper designs a series of crossover experiments, which can clearly show that the algorithm accuracy has been improved after using two feature fusion modules, and different combinations of variables will get different results. The specific experimental details are introduced in Section V.

4 Comparison and analysis of experimental result

This section will introduce the experimental parameters set in the experimental process of this subject, the data set used, and the experimental results of each part of the improved algorithm, and analyze these experimental results by contrast.

4.1 Introduction of experimental conditions

4.1.1 Experimental setup and experimental environment

The accuracy of image semantic segmentation is mainly affected by the following two aspects. First, deep learning is essentially a feature engineering that can extract abstract features, which is the same as traditional machine learning algorithms, so whether convolutional neural network has good feature extraction ability will directly affect the final results. Second, the difficulty of image semantics segmentation is how to determine the category of edge pixels more accurately. The accuracy of edge judgment greatly affects the final accuracy of the algorithm. The improved algorithm is mainly measured by MIoU value and PA value.

After the introduction and description of the previous section, in this section, we mainly set up the following experiments: (1) A comparative experiment to verify the influence of the global context structure on the accuracy of the image semantic segmentation algorithm. (2) A comparative experiment to verify the effect of the decoder structure on the accuracy of the image semantic segmentation algorithm. (3) A comparative experiment to verify the influence of multi-scale feature fusion framework on the accuracy of image semantic segmentation algorithm.

All experiments in this subject are based on the python version of DeepLab and use the open-source framework of TensorFlow. The data sets used in the experiment are all open data sets. All the experiments of this subject are completed on personal PC. The main parameters of the experimental environment are shown in Table 2.

Table 2 Experimental environment

Name	Configuration
Operating system	Ubuntu14.04
CPU	i7 7700k
GPU	GTX1080Ti
Memory	11G
CUDA	8.0
python	2.7
TensorFlow	1.4

4.1.2 Data set

The Pascal VOC dataset. Pascal VOC is one of the most popular image competitions in the world. It includes a variety of contests, including object recognition, object detection, image semantics segmentation, and so on. In this paper, we mainly use the image semantics segmentation data set. For the image semantics segmentation task of Pascal VOC2012, each image of training set and verification set has corresponding label. The main task of VOC2012 data set is to recognize 20 kinds of objects, as shown in Fig. 11.

The Pascal VOC 2012 dataset includes 1464 training sets, 1449 validation sets, and 1456 test sets. The Pascal VOC 2012 data set contains the Pascal VOC 2007 data set. However, the amount of data in the VOC dataset itself is not very large. Therefore, we first expand the dataset, add the dataset with the homologous VOC dataset (Shetty 2012), and finally expand the dataset to 10,582 training sets.

In addition to the Pascal VOC dataset, this paper has also carried out some experiments on the current unmanned dataset Cityscapes dataset to verify the robustness and effectiveness of the improved algorithm. The Cityscapes dataset's label tag is similar to the VOC dataset, which includes 2975 training sets, 500 validation sets, and 1525 test sets. The specific data are shown in Table 3.

Although the research results of this subject are mainly applied to unmanned driving, if the Cityscapes data set is directly used for training, its resolution is high and the training is extremely time-consuming. Therefore, we finally use Pascal VOC 2012 data set as the main data set, select the best module according to the training results and validate it on Cityscapes data set, which can increase the iteration frequency of the experiment and more likely find out the optimal parameters of the module.

4.1.3 Evaluation index

There are a variety of indicators of accuracy in the field of image semantic segmentation, which are usually derived from pixel precision and cross-comparison indicators. Assuming that there are a total of k ? 1 categories (k categories and a background), p_{ij} represents the number of pixels that belong to class i misjudged as j-type pixels. Similarly, p_{ii} and p_{jj} represent judgments. The correct number of i-type pixels and the number of j-type pixels.

Table 3 Data sets

Data set	Train set	Validate set	Test set	Picture size	Class number
VOC2012	1464	1449	1456	500 9 500	21
Shetty (2012)	9118	–	–	500 9 500	21
Cityscapes	2975	500	1525	1024 9 2048	20

The following part is a description of several evaluation indicators that will be used later.

1. Pixel Accuracy (PA): The simplest and most intuitive evaluation indicator.

   That is to say, the proportion of the correct number of pixels to the total number. As shown in Eq. (4).

$$\mathrm{PA} = \frac{\sum_{i=0}^{k} p_{ii}}{\sum_{i=0}^{k}\sum_{j=0}^{k} p_{ij}} \tag{4}$$

2. Mean Intersection over Union (MIoU): This indicator is a standard indicator in the field of image semantic segmentation. The main process is to calculate the ratio of the intersection and union of two sets. In the field of image semantics segmentation, two sets are real labels and prediction results. As shown in Formula (5).

$$\mathrm{MIoU} = \frac{1}{k+1}\sum_{i=0}^{k}\frac{p_{ii}}{\sum_{j=0}^{k} p_{ij} + \sum_{j=0}^{k} p_{ji} - p_{ii}} \tag{5}$$

Among them, MIoU has become the standard evaluation index in the field of image semantic segmentation because of its strong representativeness, high efficiency and simplicity. The well-known competitions in the field of semantic segmentation of images all use MioU as the evaluation index. Therefore, this work mainly refers to the use of MIoU as the main evaluation index, and PA value as a supplement to the MIoU index.

4.2 Data preprocessing and model training

4.2.1 Data preprocessing

Before the training, a certain preprocessing of the input picture is performed. For VOC datasets, the size of the image is variable, so in order to be able to batch processing, the input image is cropped to a fixed size, and the processed image is randomly flipped, as well as scaled randomly from 0.5 to 1.5 times, and finally Subtract the pixel mean of the dataset image. In the test set verification, since the test set image is at most (500, 500), in order to reduce the influence of the deformation on the final result, the verification picture is uniformly filled to (513, 513). The

4.2.2 Parameter initialization and optimizing function

The improved network is fine-tuned using a stochastic gradient descent algorithm with momentum. The pretrained model parameters are imported into the improved network. If the new network layer or the structure of the network layer has changed, the weights of the new convolution layer are initialized by Glorot X.'s method (Shetty 2012), as shown in Eq. (6), while bias is initialized directly to 0. The parameter updating rules are shown in expressions (7)

$$w \sim U\left[-\frac{\sqrt{6}}{\sqrt{n_j + n_{j+1}}}, \frac{\sqrt{6}}{\sqrt{n_j + n_{j+1}}}\right] \tag{6}$$

$$v_{i+1} := \alpha v_t - \beta\lambda w_i - \lambda\frac{\partial L}{\partial w_i} \tag{7}$$

$$w_{i+1} := w_i + v_{i+1} \tag{8}$$

Among them, *i* represents the number of iterations, *m* represents the momentum term, *a* represents the momentum term coefficient, *b* represents the learning rate of the current iteration number and *k* represents the weight attenuation coefficient.

4.2.3 Learning strategy

The learning strategy has a great influence on the final model's effect. Therefore, this experiment is aiming to select a better learning strategy from Chen et al. (2018a).

Strategy 1: A ''poly''-learning rate adjustment strategy, as shown in Eq. (9).

$$\alpha_{iter} = \alpha\left(1 - \frac{iter}{max\_iter}\right)^{power} \tag{9}$$

processing of the Cityscapes data set will be specified in Sect. 4.5.

Among it, *a* is the initial set learning rate, *iter* represents the current number of iterations, a_{iter} represents the learn- ing rate when the current number of iterations is *iter*, and *max_iter* represents the maximum number of iterations.

Strategy 2: A stage learning rate adjustment strategy can be expressed as a ‘‘step’’ strategy.

Table 4 shows the common parameters among these two strategies: The initial learning rate is 0.001, the momentum coefficient is set to 0.9, the weight attenuation coefficient is set to 0.0005, and random seed 1234 and so on.

In the comparison of learning strategies, the pretraining model was used with ImageNet’s VGG16 network

Table 4 Common parameters of different learning strategies

Parameter	Value
Momentum	0.9
Weight attenuation coefficient	0.0005
Random seed	1234
Initial learning rate	0.001

structure. The input pictures are randomly cut into (433, 433), and cross-validation are conducted respectively on the different number of batch and iteration to select a better benchmark model.

The results of a series of experiments are shown in Table 5. The accuracy of step strategy in this experiment is 1.5% higher than that in Chen et al. (2018a). Compared with the poly strategy in the original paper, the benchmark model in this paper is 0.8%. The polystrategy in this paper is 1.2% higher than step strategy in this paper and 2.7% higher than the benchmark model in Chen et al. (2018a). Therefore, this work will eventually take poly strategy as the learning strategy in the later experiment. The model with MIoU value of 65.03% is set as a base model, the parameters will be adjusted according to different experiments as well.

4.3 Comparison and analysis of experimental results of improved methods

4.3.1 The influence of global context structure

This section carries out experiments and comparative analysis as in Table 6, using VOC2012 training set and Shetty (2012) data set for training, VOC2012 verification set for inference.

In order to compare the global context information of different scales, the following comparative experiments are designed in this research. In the global context structure, we first need to determine the appropriate bin size and what statistical features are used for each bin. Here we choose two pooling methods: (1) maximum pooling operation and (2) mean pooling operation.

Table 5 Effect of different learning strategies on results

Learning strategy	Number of batches	Number of iterations	MIoU (%)
Step	10	6000	62.28
Poly	10	20,000	65.88
Step	10	20,000	63.71
Poly	12	15,000	64.59
Poly	12	20,000	65.03

Table 6 Comparison of different context structures

Method	PA (%)	MIoU (%)
VGG16-Baseline	90.96	65.03
VGG16 ? ave ? B1236 ? DR	91.48	66.37
VGG16 ? max ? B1236 ? DR	91.42	66.43
VGG16 ? max ? B1	91.31	66.08
VGG16 ? ave ? B1	91.61	67.28

In order to verify the validity of the algorithm quickly, the VGG16 network structure with simpler structure and fewer parameters is used as the benchmark network. And the polylearning strategy is used as well. In addition, the batch size value is set to 16 in this series of experiments, and the maximum number of iterations is 30,000.

Here B is the bin size, the feature map is divided into corresponding bin sizes, and the statistical results are aggregated, respectively. Taking B1236 as an example, it shows that the bin size of the feature map is 1 9 1, 2 9 2, 3 9 3, 6 9 6 with *max* maximum pooling operation for each bin. And *ave* represents the average pooling operation for each bin. And DR represents the dimension reduction operation after pooling features in each bin.

It can be clearly seen from Table 6 that after adding the global context structure, the accuracy of the algorithm can be effectively improved. Among them, the use of bin size is 1 9 1, and the best result is obtained when the average pooling method is adopted for the statistical characteristics. MIoU value increased by 2.25%, PA value is also increased by 0.65%.

In order to further verify the effectiveness and robustness of the improvement, we use another network structure ResNet101 as Table 7. Because the residual network has more parameters than the VGG network, and due to the limitation of hardware resources, we cannot set it according to the hyperparameter from the Chen et al. (2018a), than finally set the batch size to 5 and iteration 40,000 times to obtain the benchmark model.

For this experiment of the global context structure, set the batch size to 2 and iteration 100,000 times. The final result is shown in Table 7. After adding the global context structure to the benchmark model, the MIoU value increased by 1.45%, and the PA value is increased by 0.18%.

Table 7 The influence of global context structure

Method	Network	PA (%)	MIoU (%)
DeepLab-Baseline	ResNet101	93.01	72.55
Ours1(Context)	ResNet101	93.19	74.00

4.3.2 Influence of decoder module on image semantic segmentation results

This section conducts experiments and comparative analysis on the decoder module. It also uses the VOC2012 training set and the Shetty (2012) data set for training, and the VOC2012 verification set for inference.

In order to evaluate different decoder modules, the following comparative experiments are designed. In this series of experiments, we need to test the number of convolution kernels which used in merging shallow semantic information with deep semantic information. Because the decoder module designed in this work has two shallow semantic layers, there will be two variables. This series of experiments are based on VGG16 network structure with using polylearning strategy, batch size is set to 10 and the maximum number of iterations is 40,000 times. The number of convolutional kernels (a_1, a_2) represents the number of channels mapped from conv2_x and conv3_x network layers into decoder modules, respectively.

It can be clearly seen from Table 8 that the proposed decoder module designed can effectively improve the accuracy of the algorithm. At the same time, the effect of different channel number combinations on the accuracy is different. Because the experiment is time-consuming, we only select three groups of channel to prove the effectiveness of the improvement. Among them, the decoder module with the channel number combination (32, 32) is obtained with the best result. Finally, compared with the original algorithm, the designed decoder module increases the MIoU value by 1.35% and the PA value by 0.68%.

In order to compare the proposed method with other works and to verify the robustness, we train model from the data of Chen et al. (2018b) and Hariharan et al. (2011), which is a total of 10,582 pictures from VOC2012 and Shetty (2012), and 1449 pictures of VOC2012 were used as the inference set. The two network structures, namely VGG16 and ResNet101, are used to verify the effectiveness in other aspect.

As shown in Table 9, after adding the decoder module and global context structure to the benchmark model 1, the average accuracy of the model was improved by 3.37% when compared to that of the baseline model, which the decoder module was improved by 1.15% only.

Table 8 Comparison of different channel numbers

Channel number combination	PA (%)	MIoU (%)
–	90.96	65.03
(32, 32)	91.64	66.38
(48, 48)	91.60	66.08
(64, 64)	91.62	66.24

On the basis of this experiment, when fully connected conditional random field is added for postprocessing operations, its improvement is 0.66%, than the overall improvement accuracy is 4.04%. When decoder module and global context module were added into benchmark model 3, the average accuracy of the model was improved by 2.71% in compared with that of the baseline model, among which the decoder module was improved by 1.26%. After postprocessing with fully connected conditional random field, the model was only improved by 0.03%, which is marginal, and the final accuracy was improved by 2.74% in total.

4.3.3 Influence of multi-scale fusion framework

This section carries out experiments on multi-scale fusion framework methods, VOC2012 and article (Shetty 2012) data set are also used for training, and VOC2012 data set is used for inference. In order to verify the effectiveness of the proposed multi-scale feature fusion framework, this experiment has set the following three variables: (1) the size of the input image scale, assuming that the original scale is 1 with referring to Chen et al. (2016). Two input scale combinations (1, 0.5) and (1, 0.75, 0.5) are set and defined as D1 and D2, respectively. (2) There are two ways to fuse output feature. One is the feature fusion method, which is defined as CCFA. Another method is defined as CCFB. (3) Using additional supervision module is defined as AL. Based on the above three variables, we set up the following experiments with VGG16 network.

Although this experiment utilizes the idea of parameter sharing, multi-scale input leads to a multiple back propagation, which leads to an extremely time-consuming training. Therefore, we made the following adjustments to the experiment as: (1) Crop the picture to (321, 321). (2) When fine-tuning the model parameters, only the parameters of last few convolution layers are adjusted. It sets the batch size to 10 and iteration 20,000 times. After this adjustment, the training time is greatly reduced. The final result is shown in Table 10.

As shown in Table 10, we set up a comparison experiment to verify the effectiveness of additional monitoring module, multi-scale fusion combination and feature fusion module. It can be clearly seen from the table that the results of multi-scale fusion combination (1, 0.75, 0.5) are about 0.2% higher than those of multi-scale fusion combination (1, 0.5). Due to hardware resources and time limitation, only two input combinations are selected for in these experiments. Generally, it will be a 0.2%–0.3%. When other parameters are fixed, the result of multi-scale fusion module A is slightly better than that of multi-scale fusion

Table 9 Final semantic segmentation with single model results

Method	Network	PA (%)	MIoU (%)
DeepLab-Baseline	VGG16	90.96	65.03
DeepLab-Baseline	ResNet101	93.01	72.55
Ours2(Context ? Decoder)	VGG16	92.07	68.40
Ours2(Context ? Decoder ? CRFs)	VGG16	92.37	69.06
Ours2(Context ? Decoder)	ResNet101	93.61	75.26
Ours2(Context ? Decoder ? CRFs)	ResNet101	93.67	75.29

Table 10 Influence of different variable combinations of multi-scale feature fusion algorithm on image semantic segmentation results

Method	PA (%)	MIoU (%)
VGG16-Baseline	90.96	65.03
VGG16-D1-CCFA	91.12	66.16
VGG16-D1-AL-CCFA	91.15	66.42
VGG16-D2-CCFA	91.14	66.31
VGG16-D2-AL-CCFA	91.13	66.51
VGG16-D2-CCFB	90.99	66.21
VGG16-D2-AL-CCFB	90.99	66.40

module B, which can improve about 0.1%. When MIoU with using multi-scale fusion combination (1, 0.75, 0.5), add auxiliary classifier and multi-scale fusion module A as the component module of the framework. The final MIoU value 1.48% is higher than the baseline model, and the PA value is increased by 0.18%.

Finally, after analyzing the factors of time-consuming, precision, and memory resources, we apply multi-scale fusion combination (1, 0.75, 0.5), additional monitoring module and multi-scale feature fusion module A to the best result as Ours2 in Table 11. Finally, MIoU increased by 1.27% and PA increased by 0.33%. However, since it is a multi-scale fusion strategy, the performance is relatively general.

4.4 Comparisons with related works

This experiment compares the improved algorithm with other related works, as shown in Table 12. We try our best to compare the algorithms with the similar conditions. Since the compared models are used with different data sets, and some algorithms do not have open source or we do not have enough hardware support to reproduce the algorithms, we directly compared the results as given in the related papers comparison as referred from Chen et al. (2018a).

Table 12 Comparison on the Pascal VOC 2012 dataset

Method	Network	PA (%)	MIoU (%)
DeepLab-Baseline	VGG16	94.26	66.52
Ours1(Context)	VGG16	94.62	67.23
Ours2(Context ? Decoder)	VGG16	95.00	69.28

Among them, ▮ represents that its network model uses both ImageNet data set and Microsoft COCO data set for pretraining. Ghiasi and Fowlkes (2016) introduces the concept of superpixel and designs a complex and effective decoder structure, which makes full use of shallow semantic information. Compared with Ghiasi and Fowlkes (2016), the decoder module in this paper is relatively simple, and only recovers some details, so the effect is relatively general. Wu and He (2018) is exactly the same as the base model used in this paper. Its best results are trained by three scale inputs (1, 0.75, 0.5). Because the benchmark model is improved in this paper, the multi-scale feature fusion model in this paper has good results.

Chen et al. (2018a) is the baseline model of this experiment, in which the multi-scale feature fusion model is adopted in the experiment with ResNet101 as benchmark model, which requires high hardware resources, so this experiment does not reproduce the results of its multi-scale model training.

Chen et al. (2017) is one of the high-precision algorithms. It uses a variety of recent high-precision algorithm modules, and single GPU cannot reproduce the results. Under the same conditions, Our network uses VGG16, which has been greatly improved the precision in compared with the original model, and the MIoU index has been improved by 4.64%. However, the training of multi-scale feature fusion model is extremely time-consuming and using of video memory resources. Therefore, this

Table 11 Effect of multi-scale feature fusion framework

Methods	Network	PA (%)	MIoU (%)
Ours2(Context ? Decoder)	VGG16	92.07	68.40
Ours3(Context ? Decoder ? MSC)	VGG16	92.40	69.67

experiment only trains VGG16 network to verify the effectiveness of the proposed algorithm.

4.5 Time analysis of improved methods

The speed of the algorithm is also important, so this experiment records the training time and inference time of different models before and after improvement, as shown in Table 13. Due to limited hardware resources, the batch size during training will be changed.

Taking the VGG network as an example, the global context module and the decoder module all add a new convolution layer to the reference model, and the multi-scale feature fusion model of this work inevitably carry out multiple back propagation, which leads to longer training time and a certain increase in inference time.

According to the single model and the benchmark model is VGG16, the inference time has increased by about 0.016 s, the speed can reach about 17 frames per second, and the accuracy has been improved by 3.4%, which basically meets the real-time requirements.

The multi-scale feature fusion algorithm can improve the accuracy of the algorithm, but it will affect the time performance. As the global context structure, it can improve the accuracy without increasing the much inference time.

Our algorithm takes more training time, while the algorithm based on the residual network increases the inference time by about 0.015 s with improving the accuracy of about 2.71%. In general, it improved algorithm can improve the accuracy while not increasing the time consumption too much.

4.6 Experiment on Cityscapes dataset

Since this work needs to be applied to the unmanned driving application, we have conducted the experiment on Cityscapes datasets. Due to the large resolution of Cityscapes dataset, all images are 1024 × 2048, the image was randomly cropped to (713, 713). In order to better cover the original image, we increased the number of training with 150 epochs. Considering the time performance and hardware resources, the VGG16 network structure is selected for this experiment.

For latest methods comparison, we added the DFAnet method as comparison method, it reports the average time from running through the all test images from Cityscapes using the best-performing networks. The baseline of the DFAnet A method achieves MIoU 71.3% When the backbone model is decreases to a simplified one, the accuracy performance of DFANet B is decreased to 67.1% with still 120 FPS inference speed.

As shown in Table 14, after adding the global context structure to the benchmark algorithm, the MIoU value increased by 0.71% and the PA value increased by 0.36%. After adding the improved decoder module based on this, the MIoU index increased by 2.05% and the PA value increased by 0.38%. The total index of MIoU increased by 2.76%, and the value of PA increased by 0.74%. Since Cityscapes data set is a city scene, its environment is complex and changeable, which is highly dependent on the prior information of the environment. Therefore, the addition of global context structure effectively improves the accuracy of the algorithm. Meanwhile, the designed decoder module can restore the edge information of objects, and the specific effect will be described in the following paragraphs.

In the final of experiments, we summarize the above experimental results as follows.

1. In the experiment of learning strategy, ‘‘poly’’strategy has better accuracy than step strategy.
2. For the proposed global context structure, MIoU value is increased by 2.25%, PA value increased by 0.65% for VGG16. MIoU value increased by 1.45%, and the PA value is increased by 0.18% for ResNet101 model.
3. The designed decoder module increases the MIoU value by 1.35% and the PA value by 0.68% with the channel number combination (32, 32).

Table 13 Time performance comparison

Methods	Network	Training time (s/picture)	Testing time (s/picture)	MIoU (%)
DeepLab-Baseline	VGG16	0.0691	0.04441	65.03
DeepLab-Baseline	ResNet101	0.1065	0.10111	72.55
Ours2(Context ? Decoder)	VGG16	0.1098	0.05950	68.40
Ours2(Context ? Decoder ? CRFs)	VGG16	0.1098	2.36849	69.06
Ours2(Context ? Decoder)	ResNet101	0.2265	0.11641	75.26
Ours3(Context ? Decoder ? MSC)	VGG16	0.5133	0.10712	69.67

Table 14 Cityscapes dataset results

Method	Network	MIoU (%)
DeepLab-CRFs-Public (Chen et al. 2018a)	VGG16	67.67
DeepLab-MSC-attention (Chen et al. 2016)	VGG16	69.08
LRR-4x[57]	VGG16	70.30
DeepLabv2-MSC I (Chen et al. 2018a)	ResNet101	76.35
DeepLabv3 (Krähenbühl and Koltun 2011)	ResNet101	78.51
DeepLab-Baseline	VGG16	65.03
DeepLab-Baseline	ResNet101	72.55
SSOD (Salscheider 2019)	–	61.2
DFANet A (Li et al. 2019)	–	71.3
DFANet B (Li et al. 2019)	–	67.1
Ours2(Context ? Decoder)	VGG16	68.40
Ours3(Context ? Decoder ? MSC)	VGG16	69.67
Ours2(Context ? Decoder)	ResNet101	75.26

4. In the proposed multi-scale fusion framework, the final MIoU value 1.48% is higher than the baseline model, and the PA value is increased by 0.18%.
5. With Pascal VOC 2012 dataset, MIoU index of our network with VGG16, has 4.64% improvement.
6. For time performance, our improved algorithm can improve can improve the accuracy without increasing the much inference time.
7. With Cityscapes dataset for improved global context structure and decoder module, the MIoU index increased by 2.05% and the PA value increased by 0.38%.

4.7 Demonstration

In this section, we show the prediction effect of the improved algorithm on the other dataset, and use the improved algorithm.

4.7.1 The experimental results with Pascal VOC dataset

In the urban scene, there are problems such as large traffic volume, many pedestrians, and complicated road conditions. From the prediction results, our improved algorithm distinguishes important obstacles in the scenes, namely vehicles, pedestrians, road surface, and road edge. However, our algorithm still has a shortcoming on identifying small objects on the roadside, such as street lights and traffic signs (Fig. 12).

4.7.2 Cityscapes dataset experimental results

In the urban scene, there are problems such as large traffic volume, many pedestrians, and complicated road conditions. From the prediction results, our improved algorithm distinguishes important obstacles in unmanned driving, namely vehicles, pedestrians, the identification of the road surface, and a good distinction between the edge of the road surface. However, our algorithm still cannot identify small objects on the roadside, such as street lights and traffic signs (Fig. 13).

4.7.3 University scene experiment results

The application example of this work is the unmanned sweeping vehicle in our campus scene. Firstly, it is feasible to be used in the campus, which the campus is not as complicated and running speed of sweeper is slow. The improved algorithm is run at a speed of 17 frames per second, and only semantically segmenting the region of interest, than, the sweeper avoid the car with other sensors. It can basically meet the requirement of slow driving sweeper in the campus scene. The application platform of this work is shown in Fig. 14a.

In practical application, this work directly uses the model which is trained by using Cityscapes dataset to predict in the actual campus scene. We collected several common road conditions and scenes on our campus, and found that if the image is directly predicted, the performance is very poor. So we made some attempts to choose the right image size and aspect ratio, which changes the original size (1440, 1080) to (800, 400).

As is seen in Fig. 14b, key information such as road surface, pedestrians, vehicles, and road edges can be well identified on the standard roads. But for the non-standard roads, the segmentation on the edge of the road is poor. At the same time, some special objects in campus scenes will be misjudged due to lack of training data, such as rubber speed hump.

5 Conclusion

This work mainly analyzes the popular image semantic segmentation of DeepLab algorithm, we discover the problems and proposed the solutions. On this basis, we designed the global context module, efficient decoder module, and multi-scale feature fusion as in this framework, which ultimately improves the accuracy of the semantic segmentation algorithm. In a conclusion, this research mainly completed the following works:

1. The global context structure is introduced into the DeepLab algorithm, and the global features of the feature map are extracted by means of the mean pooling method. The global features and the original features are combined by the combination method, which provides the overall prior information of the image. These operations effectively improve the feature extraction ability of the model.
2. Because the original decoder module of DeepLab algorithm is relatively simple, the boundary of the result is rough. Therefore, this paper designed a decoder module, which combines the shallow semantics with the deep semantics, and it can adjust the proportion of the shallow semantics and the deep semantics. Thus, the boundary of the object is optimized to improve the accuracy of the algorithm.
3. Combining the learning ideas with the multi-scale training model, we designed two effective feature-level fusion methods, and integrated an additional monitoring module. Although the efficiency of the model is reduced, the robustness is also effectively improved.
4. By setting different network structures and parameters to conduct the crossover experiments, our algorithm is verified that can effectively improve the accuracy. Furthermore, we compare the performance with the other mainstream algorithm on the Cityscapes and VOC2012 datasets, the improved algorithm greatly improves the accuracy of the algorithm without too much increase of inference time. At the same time, the improved algorithm is applied to our campus scene, which demonstrates its practical value on sweeper platform.

For the future work, we will integrate more functions into this semantic segmentation algorithm to become a multi-task deep learning network, such as visual depth prediction, which is very necessary in driverless application. In addition, in order to reduce costs, we will also research on hardware acceleration, so that semantic segmentation can also run on the low-cost edge devices.

Funding This study was funded by Shenzhen Government (Grant Nos. KQJSCX20170726104033 357, JCYJ20150513151706567 and JCYJ20160531191837793). Furthermore, this work was partially supported by the Department of Industrial and Systems Engineering of the Hong Kong Polytechnic University (Grant No. H-ZG3K).

Compliance with ethical standards

Conflict of interest All authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

References

Badrinarayanan V, Kendall A, Cipolla R (2017) Segnet: a deep convolutional encoder–decoder architecture for image segmentation. IEEE Trans Pattern Anal Mach Intell 39(12):2481–2495

Chaurasia A, Culurciello E (2017) LinkNet: exploiting encoder representations for efficient semantic segmentation.

Chen LC, Papandreou G, Kokkinos I et al (2014) Semantic image segmentation with deep convolutional nets and fully connected CRFs. Comput Sci 4:357–361

Chen LC, Yang Y, Wang J et al (2016) Attention to scale: scale-aware semantic image segmentation. In: IEEE conference on computer vision and pattern recognition, Las Vegas, pp 3640–3649

Chen LC, Papandreou G, Schroff F et al (2017) Rethinking atrous convolution for semantic image segmentation.

Chen LC, Papandreou G, Kokkinos I et al (2018a) Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs[J]. IEEE Trans Pattern Anal Mach Intell 40(4):834–848

Chen LC, Zhu Y, Papandreou G et al (2018b) Encoder–decoder with atrous separable convolution for semantic image segmentation. arXiv preprint

Chollet F (2017) Xception: deep learning with depthwise separable convolutions. Cordts M, Omran M, Ramos S et al (2016) The cityscapes dataset for semantic urban scene understanding. In: IEEE conference on computer vision and pattern recognition, Las Vegas, pp 3213–3223

Criminisi A, Shotton J, Konukoglu E (2012) Decision forests: a unified framework for classification, regression, density estimation, manifold learning and semi-supervised learning. Found Trends Comput Graph Vis 7(2–3):81–227

Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. In: IEEE conference on computer vision and pattern recognition, San Diego, pp 886–893

Dvornik N, Shmelkov K, Mairal J et al (2017) BlitzNet: a Real-time deep network for scene understanding. In: IEEE international conference on computer vision, Venice, pp 4174–4182

Eigen D, Fergus R (2015) Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In: IEEE international conference on computer vision, Santiago, pp 2650–2658

Everingham M, Gool LV, Williams CKI et al (2010) The Pascal visual object classes (VOC) challenge. Int J Comput Vis 88(2):303–338

Geiger A (2012) Are we ready for autonomous driving? The KITTI vision benchmark suite. In: IEEE conference on computer vision and pattern recognition, Portland, pp 3354–3361

Ghiasi G, Fowlkes CC (2016) Laplacian pyramid reconstruction and refinement for semantic segmentation. In: European conference on computer vision, Amsterdam, pp 519–534

Glorot X, Bengio Y (2010) Understanding the difficulty of training deep feed forward neural networks. In: Proceedings of the 13th international conference on artificial intelligence and statistics, Sardinia, pp 249–256

Glorot X, Bordes A, Bengio Y (2011) Deep sparse rectifier neural networks. In: Proceedings of the 14th international conference on artificial intelligence and statistics, Fort Lauderdale, pp 315–323

Hariharan B, Arbelaez P, Bourdev L et al (2011) Semantic contours from inverse detectors. In: IEEE international conference on computer vision, Barcelona, pp 991–998

He K, Zhang X, Ren S et al (2016) Deep residual learning for image recognition. In: IEEE conference on computer vision and pattern recognition, Las Vegas, pp 770–778

Hinton GE, Osindero S, Teh YW (2006) A fast learning algorithm for deep belief nets. Neural Comput 18(7):1527–1554

Huang G, Liu Z, Weinberger K Q et al (2017) Densely connected convolutional networks. In: IEEE conference on computer vision and pattern recognition, Hawaii, vol 1, no 2, p 3

Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International conference on international conference on machine learning, Lille, pp 448–456

Kong T, Yao A, Chen Y et al (2016) HyperNet: towards accurate region proposal generation and joint object detection. In: IEEE conference on computer vision and pattern recognition, Las Vegas, pp 845–853

Krähenbühl P, Koltun V (2011) Efficient inference in fully connected CRFs with Gaussian edge potentials. In: Advances in neural information processing systems, Granada, pp 109–117

Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. Advances in neural information processing systems, Lake Tahoe, pp 1097–1105

Lecun Y, Bottou L, Bengio Y et al (1998) Gradient-based learning applied to document recognition. Proc IEEE 86(11):2278–2324

Lee CY, Xie S, Gallagher P et al (2014) Deeply-supervised nets. In: Artificial intelligence and statistics, Reykjavik, pp 562–570

Li H, Xiong P, Fan H, Sun J (2019) DFAnet: deep feature aggregation for real-time semantic segmentation. arXiv.org

Lienhart R, Maydt J (2002) An extended set of Haar-like features for rapid object detection. In: International conference on image processing, vol 1, I-900–I-903

Lin TY, Maire M, Belongie S et al (2014) Microsoft coco: common objects in context. In: European conference on computer vision, Zurich, pp 740–755

Lin G, Milan A, Shen C et al (2017a) RefineNet: multi-path refinement networks for high-resolution semantic segmentation. In: IEEE conference on computer vision and pattern recognition, Hawaii, pp 5168–5177

Lin T, Dollar P, Girshick RB et al (2017b) Feature pyramid networks for object detection. In: IEEE conference on computer vision and pattern recognition, Hawaii, pp 936–944

Liu W, Rabinovich A, Berg AC (2015) Parsenet: looking wider to see better. arXiv preprint arXiv:1506.04579

Liu W, Anguelov D, Erhan D et al (2016) SSD: single shot MultiBox detector. In: European conference on computer vision, Amsterdam, pp 21–37

Liu X, Deng Z, Yang Y (2018) Recent progress in semantic image segmentation. Artif Intell Rev 6:1–18

Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: IEEE conference on computer vision and pattern recognition, Boston, pp 3431–3440

Mehta S, Rastegari M, Caspi A, Shapiro L, Hajishirzi H (2018) Espnet: efficient spatial pyramid of dilated convolutions for semantic segmentation. arXiv.org

Pai-Hsuen Chen, Chih-Jen Lin, Bernhard Schö lkopf (2005) A tutorial on v-support vector machines. Appl Stoch Models Bus Ind 21(2):111–136

Romera E, Alvarez J, Bergasa L, Arroyo R (2017) Efficient ConvNet for real-time semantic segmentation. In: 2017 IEEE intelligent vehicles symposium (IV). IEEE

Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: International conference on medical image computing and computer-assisted intervention, Munich, pp 234–241

Rumelhart DE, Hinton GE, Williams RJ (1986) Learning internal representations by error propagation. In: Rumenhart DE, McCelland JL (eds) Parallel distributed processing: explorations in the microstructure of cognition. MIT Press, Cambridge, pp 318–362

Russakovsky O, Deng J, Su H et al (2015) ImageNet large scale visual recognition challenge. Int J Comput Vis 115(3):211–252

Salscheider N (2019) Simultaneous object detection and semantic segmentation

Shetty S (2012) Application of convolutional neural network for image classification on pascal voc challenge 2012 dataset.

Simonyan K, Zisserman A (2014) Very deep convolutional networks for large-scale image recognition. arXiv preprint

Srivastava N, Hinton G, Krizhevsky A et al (2014) Dropout: a simple way toprevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958

Sun Z, Xue L, Xu Y (2012) A review of in-depth learning. Comput Appl Res 29(8):2806–2810

Szegedy C, Liu W, Jia Y et al (2015) Going deeper with convolutions. In: IEEE conference on computer vision and pattern recognition, Boston, pp 1–9

Wang P, Chen P, Yuan Y et al (2017) Understanding convolution for semantic segmentation. arXiv preprint arXiv:1702.08502

Wei Y, Zhao Y (2016) Review of image semantic segmentation based on DCNN. J Beijing Jiaotong Univ 40(4):82–91

Wu Y, He K (2018) Group normalization. arXiv preprint arXiv:1803. 08494

Yang F (2014) Development status and prospects of driverless cars. Shanghai Automot 3:35–40

Yu H, Yang Z, Tan L, Wang Y, Sun W, Sun M et al (2018) Methods and datasets on semantic segmentation: a review. Neurocom- puting 304:S0925231218304077

Zhao H, Shi J, Qi X et al (2017a) Pyramid scene parsing network. In: IEEE conference on computer vision and pattern recognition, Hawaii, pp 2881–2890

Zhao H, Qi X, Shen X et al (2017b) Icnet for real-time semantic segmentation on high-resolution images. arXiv preprint arXiv: 1704.08545

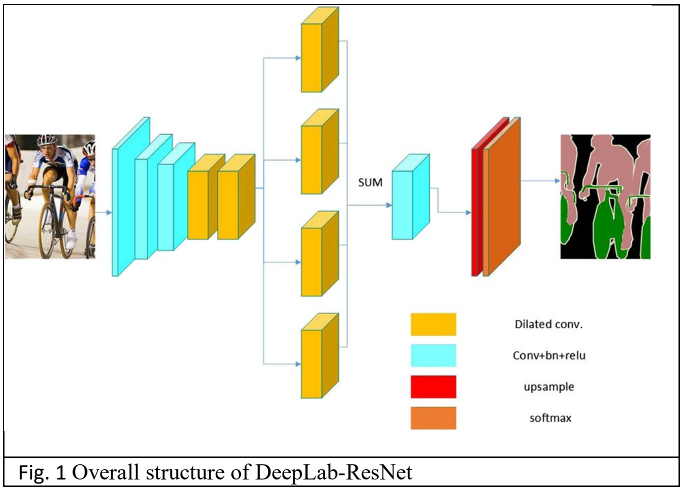


Fig. 1 Overall structure of DeepLab-ResNet

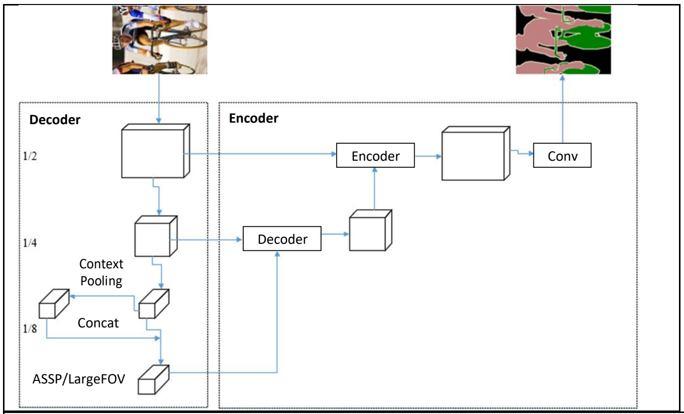


Fig. 2 Proposed improved framework

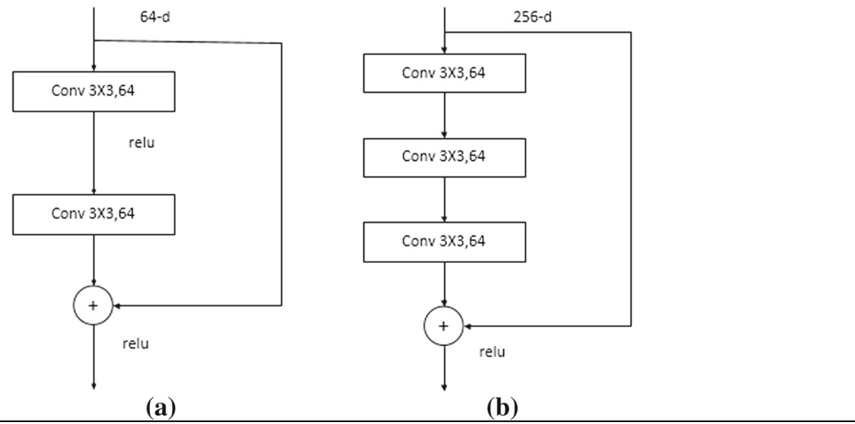


Fig. 3 VGG network structure

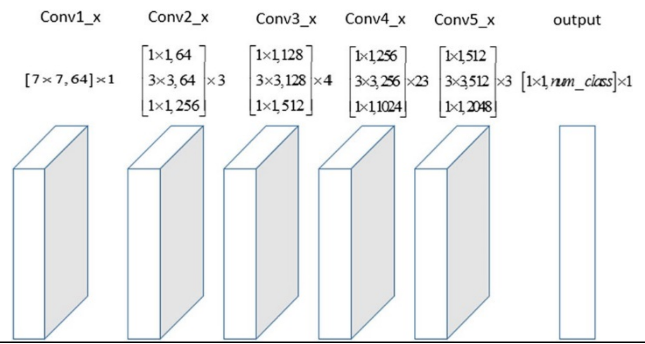


Fig. 4 ResNet’s bottleneck structure

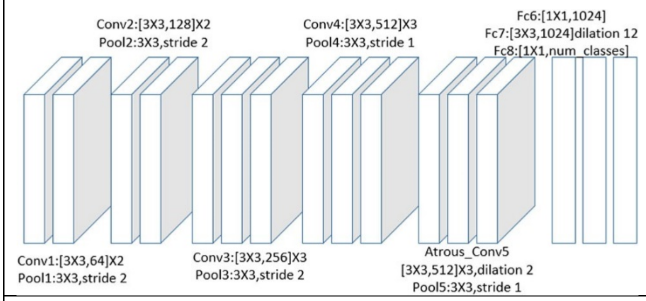


Fig. 5 ResNet101 network architecture

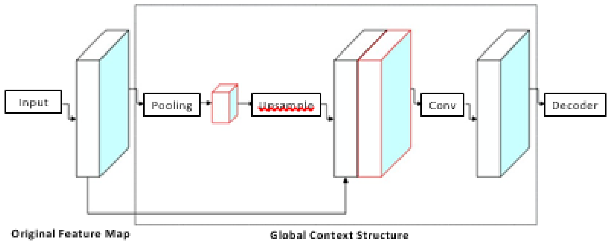


Fig.6 Proposed global context structure

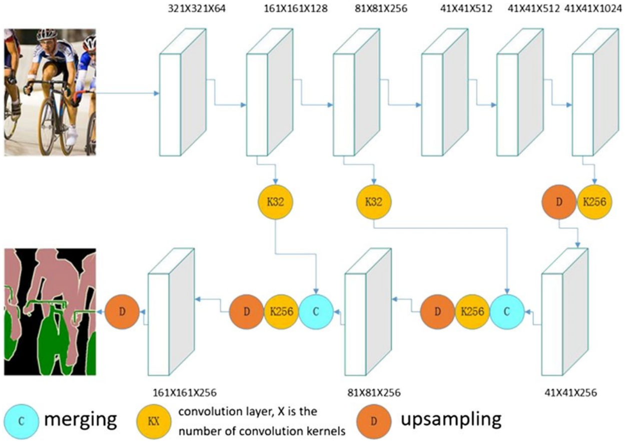


Fig. 7 Proposed decoder module

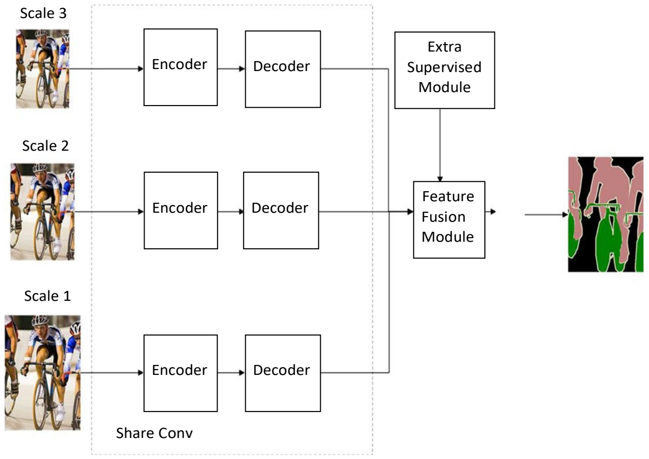


Fig. 8 Proposed multi-scale feature fusion framework

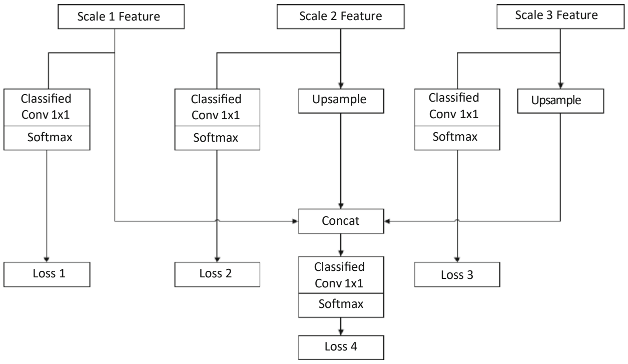


Fig. 9 Feature fusion of module A

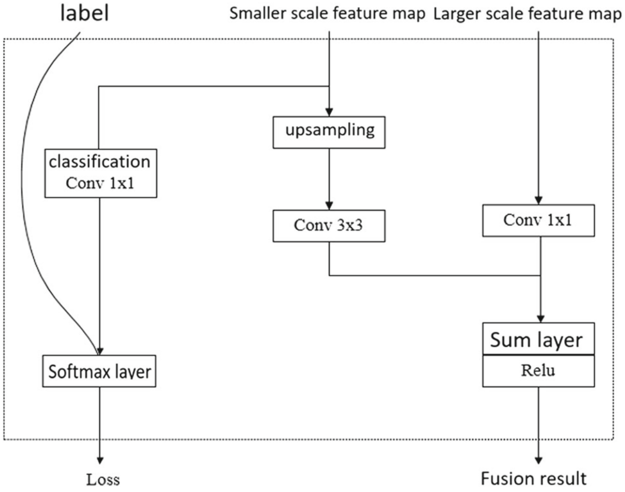


Fig. 10 Feature fusion of module B

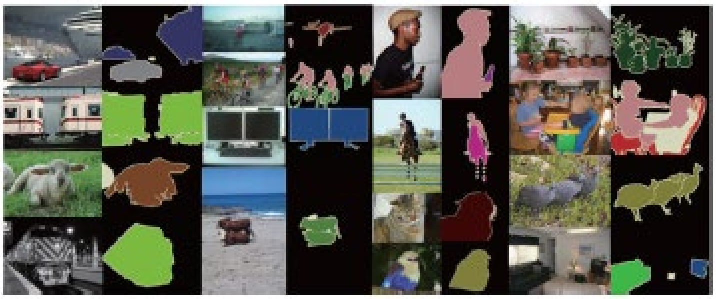


Fig. 11 Pascal VOC2012 data set example (Glorot and Bengio 2010)

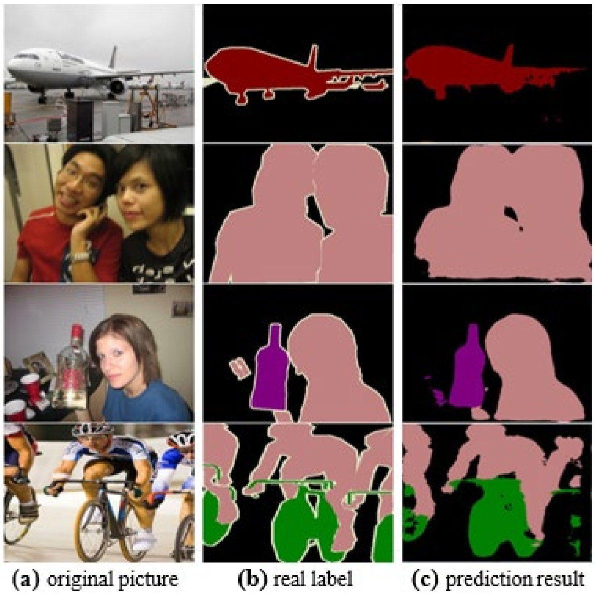
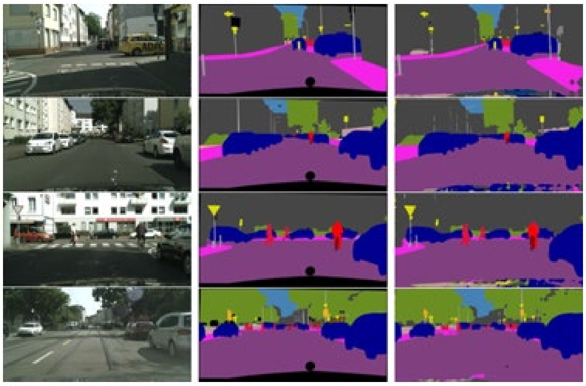
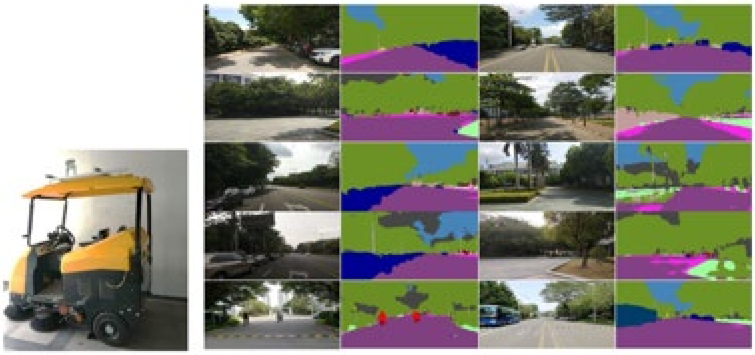


Fig. 12 Partial rendering of the Pascal VOC data set



(a) original Image **(b)** real label **(c)** prediction result

Fig. 13 Partial results of the Cityscapes dataset



(a) **(b)**

Fig. 14 Application of sweeper platform, b rendering of the campus scene of our university