Convolutional Neural Network for the Semantic Segmentation of Remote Sensing Images

Muhammad Alam^{1,2} · Jian-Feng Wang³ · Cong Guangpei⁴ · LV Yunrong⁴ · Yuanfang Chen⁵

Accepted: 25 November 2020 / Published online: 12 February 2021
© The Author(s) 2021

Abstract

In recent years, the success of deep learning in natural scene image processing boosted its application in the analysis of remote sensing images. In this paper, we applied Convolutional Neural Networks (CNN) on the semantic segmentation of remote sensing images. We improve the Encoder- Decoder CNN structure SegNet with index pooling and U-net to make them suitable for multi-targets semantic segmentation of remote sensing images. The results show that these two models have their own advantages and disadvantages on the segmentation of different objects. In addition, we propose an integrated algorithm that integrates these two models. Experimental results show that the presented integrated algorithm can exploit the advantages of both the models for multi-target segmentation and achieve a better segmentation compared to these two models.

Keywords Convolutional Neural Networks (CNN) · Deep learning · Remote sensing images · Semantic segmentation

✉ LV Yunrong
lyclyr@yeah.net

Muhammad Alam
alam52@lsbu.ac.uk

Jian-Feng Wang
e-vist@qq.com

Cong Guangpei
5414712@qq.com

Yuanfang Chen
yuanfang.chen.tina@gmail.com

- ¹ School of Engineering, London South Bank University, London, UK
- ² Instituto de Telecomunicações, Campus Universitário de Santiago, Aveiro, Portugal
- ³ Suzhou Life Intelligence Industry Research Institute, Suzhou, China
- ⁴ Guangdong Institute of Petrochemical Technology, Maoming, 525000 Guangdong, China
- ⁵ Yuanfang Chen Hangzhou Dianzi University, Hangzhou, China

1 Introduction

Semantic segmentation of images is a technique to divide images into several independent regions with their own characteristics and extract the objects of interest. Image semantic segmentation technology can segment and mark specific targets in remote sensing images so as to extract specific information in remote sensing images researches. For example, semantic segmentation technology can segment and extract buildings or vegetation in remote sensing images, which can provide basic support for other researches. Semantic segmentation of remote sensing image is an important research work, which can promote the development of military, agriculture, environmental protection and other important research areas [1]. On marine monitoring, it can monitor the situation of the sea effectively and make necessary warnings in time. On urban planning, the segmentation of buildings, roads, trees can help the government make reasonable planning for urban layout effectively. In the aspect of environmental supervision, it can classify the targets of key and large pollution enterprises [2], which can help to focus on monitoring and regulate the emission behaviors effectively. With the development

of the remote sensing technology, it can get more and more information from the remote sensing images [3]. The application of remote sensing images is expanding and has become an indispensable element of people's life.

In recent years, with the development of machine learning and deep learning, Convolutional Neural Networks (CNN) has achieved good results in the field of computer vision. It has been well used in image recognition, object detection and other areas. Based on machine learning, deep learning is a new research field. By constructing neural networks with many layers and using a large number of training samples, it learns the main features and ultimately improves the accuracy of classification or prediction [4]. Because convolution layer is very effective for image feature extraction, many researchers have applied the deep convolution neural network to the field of image semantic segmentation [5, 6]. Before deep learning, researchers made use of the texture and spectral feature of the images to do segmentation [7]. However, these methods can not get enough details of the information in the segmentation of remote sensing images and can not give the satisfied segmentation for difficult tasks [8, 9]. Recent years, deep learning algorithms have more and more advantages in image processing [10, 11]. This is of great significance in the research of remote sensing image processing. The success of deep learning in natural scene image promotes the development of deep learning in remote sensing images. As the shadow noise interference, uneven texture distribution, uneven change in illumination and other factors, it is challenge to process the complex remote sensing images [12]. So it needs higher requirements for the application of deep learning in remote sensing images. It is essential and urgent to study how to use deep learning to segment remote sensing images more effectively and accurately. In this study, we aim to apply deep learning technology to the multi-targets semantic segmentation of remote sensing images. Buildings, roads, vegetation and water are the most important information in the remote sensing images. We design the CNN models SegNet and U-net that can be used for multi-targets semantic segmentation of remote sensing images. To explore the deep learning technology applied on the semantic segmentation of remote sensing images and develop the novel technology and algorithms to get a higher accuracy.

The main contributions of the work are listed as follow:

- Based on SegNet, this work designs the Encoder-Decoder CNN structure can be used for multi-targets semantic segmentation of remote sensing images with Pooling Index. Pooling Index can store the position information of the special pixels, which can be restored when decoding. As the position of the pixels can

be restored, it can give a more clear details for the segmentation of the edges.

- This work introduces the improved Batch Normalization to CNN to make it suitable to do multi-targets semantic segmentation of remote sensing images. This introduced Batch Normalization can change the distribution of the data input to the next layer, which on one hand does not need to learn the new distribution, on the other hand can update the parameter more efficiently.
- This work Compares the results of these two above-mentioned methods especially the specific advantages and disadvantages on different classes of the objects segmentation in the remote sensing images.
- This work proposes a algorithm that can integrate the semantic segmentation results of different CNN models. In deep learning models, each has its own advantages and disadvantages in different applications or for different specific objects. With the algorithm proposed by this work, the results can be integrated the segmentation in pixels. This can make use of the advantages of different models on different objects and then produce a better results.

In the end, we compare this result with that of other researchers with other methods and discussed the advantages of our method over that of other researchers.

2 Preliminary work

2.1 Basic structure

Convolutional neural network is a hierarchical model whose input is raw data, such as RGB image and raw audio data. Convolutional neural networks extract high-level semantic information layer by layer from the input layer of raw data by stacking a series of operations such as convolution operation, pooling operation and mapping of non-linear activation functions. This process is called “feed-forward operation”. Different types of operations in the convolutional neural networks are called “layers”. Convolution operations are convolutional layers and pooling operations are pooling layers. The last layer of convolutional neural network transforms its target tasks (classification, regression, etc.) into the objective function. By calculating the error or loss between the predicted value and the real value, the error or loss back-forward layer by layer by the back-propagation algorithm to update the parameters of every layer and then back-forward again and again until the network model converges.

CNN takes the convolution operation layer as the “basic unit” and put it on the original data one layer by one layer and ends with the loss function. The data in every layer is

a three-dimensional tensor. In computer vision applications, the data layer of convolutional neural network is usually the image of RGB color space: H rows, W columns, three channels (R, G, B).

Finally, the whole network ends with the calculation of loss function. If y is the ground truth of the input x^L , the loss function is as below:

$$z = L(x^L, y)$$

In the function of $L()$, for different tasks, the form of loss function also changes. Taking regression tasks as an example, the common L loss function can be used as the objective function of convolution network. In this situation, the z loss function is as below:

$$z = L(x^L, y) = \frac{1}{2} \|x^L - y\|^2$$

For classification tasks, the objective function of the networks is cross entropy:

$$\rho_i = \frac{\exp(x_i^L)}{\sum_{j=1}^C \exp(x_j^L)} (i = 1, 2, \dots, C)$$

C is the number of classification. No matter it is the regression tasks or classification tasks, before the calculation of z , it needs suitable operation to get the x^L with the same dimensions and then to get the correct loss calculation.

2.2 Feed forward

The feed-forward operation of convolutional neural network is more intuitive whether the error is calculated in the training model or the sample prediction is obtained after the training of the model. Taking the image classification task as an example, it is assumed that the network has been trained. The prediction process is actually a feed-forward operation of the network. The test set image is sent into the network as network input x^1 , and then x^2 can be obtained after the operation. And then repeat this until it output x^L . As mentioned above, x^L is a vector that has the same dimension as the real notation. In the network trained by using the cross entropy loss function, each dimension of x^L can represent the probability that x^1 belongs to C categories respectively. Thus, the prediction mark corresponding to input image x^1 can be obtained by the following formula:

$$\arg \max(x_i^L)$$

2.3 Back forward

Like many other machine learning models (support vector machines, etc.), convolutional neural networks, including all other deep learning models, rely on minimizing loss functions to learn model parameters. However, the neural

network model is not only non-convex from the perspective of convex optimization theory. The function is extremely complex, which makes it difficult to optimize the solution. In this case, stochastic gradient descent method (SGD) and error back propagation are used to update the model parameters.

Specifically, batch stochastic gradient descent is often used to solve convolution neural networks, especially for large-scale application problems. The batch stochastic gradient descent method randomly selects n samples as a batch of samples in the training model stage. The error is calculated by feed-forward operation, and the parameters are updated by gradient descent method. Gradients are feedback layer by layer from back to front until they are updated to the first layer parameters of the network. Such a parameter update process is called a “mini-batch”. All training set samples are traversed by non-return sampling between different batches. Traversing all the training sample once is called an epoch. The size of batch samples should not be too small. Because the sample is sampled randomly, updating the model parameters according to the errors on the sample may not be globally optimal (it is only locally optimal), which will make the training process oscillate. The upper limit of batch size mainly depends on the limitation of hardware resources, such as the memory size of GPU. Of course, there are other different parameter updating strategies when updating parameters with SGD.

Assuming that the error on n samples is z after feed-forward of a batch process and the last layer L has a loss function L_2 :

$$\frac{\partial z}{\partial w^L} = 0$$

$$\frac{\partial z}{\partial x^L} = x^L - y$$

Each layer actually corresponds to two derivatives: One is the derivative of the error with respect to the parameter of the layer i $\frac{\partial z}{\partial w^i}$, the other is the derivative with respect to the input of the layer $\frac{\partial z}{\partial x^i}$. The derivative of the parameter w^i is used for updating the parameters in this layer:

$$w^i \leftarrow w^i - \eta \frac{\partial z}{\partial w^i}$$

η is the step size of each stochastic gradient descent and generally it decreases with the increase of training epochs.

The derivative $\frac{\partial z}{\partial x^i}$ of input x^i is used for the back propagation of errors to the front layer. It can be regarded as the error from the last layer to the layer i .

Take the parameters update of layer i as an example. When the error update signal (derivative) propagates back to layer i , the error derivative of layer $i + 1$ is $\frac{\partial z}{\partial x^{i+1}}$. When the parameters of the layer i are updated, the corresponding

values of partial $\frac{\partial z}{\partial w^i}$ and partial $\frac{\partial z}{\partial x^i}$ need to be calculated. According to the chain rule:

$$\frac{\partial z}{\partial (\text{vec}(w^i)^T)} = \frac{\partial z}{\partial (\text{vec}(x^{i+1})^T)} \cdot \frac{\partial (\text{vec}(x^{i+1})^T)}{\partial (\text{vec}(w^i)^T)}$$

$$\frac{\partial z}{\partial (\text{vec}(x^i)^T)} = \frac{\partial z}{\partial (\text{vec}(x^{i+1})^T)} \cdot \frac{\partial (\text{vec}(x^{i+1})^T)}{\partial (\text{vec}(x^i)^T)}$$

The vector notation “vec” is used here because tensor operations are transformed into vector operations in actual engineering implementation. As mentioned earlier, since layer $i + 1$ has been computed to get $\frac{\partial z}{\partial x^{i+1}}$, $\frac{\partial z}{\partial (\text{vec}(x^{i+1})^T)}$. It can be obtained by vectorization and transposition of the parameter in layer i when it is used to update the layer parameter. In the layer i , since x^i is directly acted by w^i to get x^{i+1} , the reverse derivative can also be directly obtained as its partial derivative $\frac{\partial \text{vec}(x^{i+1})}{\partial (\text{vec}(x^i)^T)}$ and $\frac{\partial \text{vec}(x^{i+1})}{\partial (\text{vec}(w^i)^T)}$. And then update the parameter of this layer, after which it passes $\frac{\partial z}{\partial x^i}$ as the error of this layer to the previous layer, which is layer $i-1$. Repeat this until it updates to the first layer to finish the updating of the parameters in a mini-batch

2.4 Convolution operation

Convolution is actually an operation in analytic mathematics. In convolution neural networks, it usually involves only discrete convolution. For D kernels and three-dimensional, the input tensor of convolution layer i is $x^i \in R^{H_i \times W_i \times D_i}$ and the kernel is $f^i \in R^{H_i \times W_i \times D_i}$. Convolution operation in three-dimensional input only extends two-dimensional convolution to all channels at corresponding positions. Finally, the sum of all values processed by the convolution is taken as the convolution result of this position. if there are D convolution kernels like f^i , the convolution output of $1 \times 1 \times 1 \times D$ dimensions can be obtained at the same location. D is the channel numbers of the feature x^{i+1} in the layer $i+1$. It can be showed as below:

$$y_{i,l+1,j,l+1,d} = \sum_{i=0}^H \sum_{j=0}^W \sum_{d=0}^{D^l} f_{i,j,d,l,d} \times x_{i,l+1+i,j,l+1+j,d}^i$$

In the formula, $f_{i,j,d,l,d}$ is the weight learned from the training. To the different inputs, they have the same weight. This weight is the same for all inputs at different locations, which is called “weight sharing” in convolution layer.

2.5 Activation function

Activation function is also known as non-linearity mapping. Activation function is introduced to increase the expressive ability of the whole network. Otherwise, the stacking of several linear operation layers can only play the role of linear mapping and cannot form complex functions. Here

we just introduce ReLU(Rectified Linear Unit), proposed by Nair and Hinton [13]. ReLU is one of the most popular activation function and it can be shown as the formula below:

$$y = \max\{0, x\} = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

The activation function simulates the characteristics of biological neurons: receiving a set of input signals and generating output. In neuroscience, biological neurons usually have a threshold. When the cumulative effect of input signals obtained by neurons exceeds that threshold, neurons are activated and in an exciting state; otherwise, they are in an inhibitory state. In ReLU, when $x \geq 0$, it is activated and when $x < 0$, it cannot activate the neurons. ReLU can help the stochastic gradient descent method converge quickly and has become one of the most popular.

3 Related work

Convolutional neural networks (CNN) is a kind of artificial neural networks with the convolution operators. CNN can get excellent performance in many fields especially in image processing such as image recognition, image semantic segmentation and object detection. With the more research of CNN, CNN has also achieved better results in the areas of natural language processing and data mining than other deep neural networks models [14–16].

In 1960, Canadian neuroscientists David H. Hubel and Torsten Wiesel [17] proposed the concept of “receptive field”, a single neuron in the cat’s primary visual cortex. This is the first milestone in the history of convolutional neural networks development. Then, in 1962, they discovered the existence of receptive field, binocular vision and other functional structures in the cat’s visual center, which means that the network structure was first found in the brain visual system.

Around 1980, based on the work of Hubel and Torsten, Japanese scientist Konihiko Fukushima simulated the biological vision system and proposed a hierarchical multi-layer artificial neural network(neurocognitron) [8] to deal with handwritten character recognition and other pattern recognition tasks. Neurocognitron model is also considered as the predecessor of convolutional neural network. In this neurocognitron, the most important components are S-cells and C-cells. Two types of cells alternately stack together to form the neurocognitron network. S-cells are used for extracting the local feature and C-cells are used for abstraction and fault tolerance.

Y. LeCun proposed a convolutional neural network algorithm based on gradient learning [18], and successfully applied it to handwritten digital character recognition.

Under the technical conditions at that time, the error rate could be reduced to not more than one percent. Because of this, this kind of CNN LeNet was used for the handwritten digital character recognition to sort the postal parcels for almost all the America postal system. It can be said that LeNet is the first convolutional neural network to generate practical commercial value, and it also lays a solid foundation for the development of convolutional neural network in the future [19].

In 2012, Geoffrey E. Hinton et al. beat all the other teams on the fourth anniversary of the ImageNet Image Classification Competition, known as the World Cup in computer vision. They won the championship with an accuracy of more than 12% in the second place, and the academia and industry were shocked [10]. Since then, the convolution neural network has gradually become the dominant in the field of computer vision. And after that deep convolution neural network has been the champion of the annual ImageNet competition. Until 2015, after improving the activation function of the convolutional neural network (4.94%) on the ImageNet data set exceeded the human prediction error rate (5.1%) for the first time [20]. In recent years, with the increasing number of researchers in the field of neural networks, especially convolutional neural networks, and the rapid development of technology, convolutional neural networks have become more and more complex. From the initial layer of 5 or 16, to 152 layers of Residual Net or even thousands of layers proposed by MSRA, it has been a common practice for researchers and engineering practitioners [4, 21].

However, the structure of Alex-Net almost has no difference from LeNet decades ago. But over the decades, data and hardware devices, especially gpu, have grown tremendously. They are actually the main engines of further innovation in the field of neural networks. It is this that makes the deep neural network become a practical tool and application means. Deep convolutional neural networks have become a hot topic in the field of artificial intelligence since 2012.

Based on extracting low-level features of images, early image segmentation methods like Normalized Cuts [22–24] Watersheds [25] cannot give the semantic information. These methods do not need supervised training and even some do not need training and complex calculation. These methods can perform well in some easy tasks but cannot give the satisfied segmentation for some difficult tasks [8, 9]. To get better results, how to effectively utilize the content information contained in images and enhance the effect of image segmentation by combining the intermediate and advanced semantics of images has become a research hotspot in recent years. In 2010, the CPMC (constrained parametric min-cuts) algorithm proposed by

Carreira [22] can generate high-quality candidate regions in the image, which utilized the advanced semantics of images.

With the development of deep learning technology, new semantic segmentation technologies are emerging. In 2012, when Alex won the ImageNet competition [10], CNN has been researched by many people and it performed better and better in image processing. The multi-layer structure of CNN can automatically learn the different features of multiple levels, which is a powerful tool for image classification and has also played a great role in promoting the development of semantic segmentation.

In the early researches, the deep learning methods for image segmentation are based on patch classification [26, 27]. In these methods, each individual pixel is independently classified by image patches containing a certain range around the pixel. On the one hand, it can combine the location information of pixels, on the other hand, the full connection layer can only receive images of fixed size. Zhang W et al. [28] did the research on brain image segmentation with CNN based on patch. Le H et al. [26] used patch-based CNN to segment the tumor tissue. These researches can demonstrate the effectiveness of the methods. However these have some disadvantages. They need much more memory and have low computational efficiency.

Long J et al. [6] proposed Fully convolutional network for image semantic segmentation in 2014, which can give the categories of each pixel from abstract features. Compared with patch-based CNN, FCN can process images with different size and can increase the processing speed. Long after that most of the researches are base on FCN structure. Olaf et al. [29] proposed U-net structure based on FCN which got a good good result on medical image segmentation. Zheng S et al. [6] proposed CRF-RNN which is based on FCN structure and added some other deep learning techniques. DeepLab, proposed by Chen LC [30] used deconvolution and upsampling operation and expanded deep convolutional neural network. These two methods focused on the research that how to improve FCN to get a better segmentation because the result of FCN was rough. This because when pooling, the information of the position will lose and to get a higher accuracy the position information need to be used.

Then a series of methods based on Encoder-Decoder were developed, which improved the upsampling [18]. Noh H et al. [31] proposed DeconvNet which changed the rough feature map upsampling into a deconvolution processing. Yang J et al. [32] developed CEDN, based on Encoder-Decoder structure, used for edge detection. This structure removed all the fully connection layers with deconvolution layers after the convolution layers, which is not exactly symmetrical and just simplified the layers. Badrinarayanan

V [33] gave the SegNet structure which removed all the fully connection layers and used max-pooling upsampling. This structure can output two feature maps after every pooling layer, one for the next layer and another for the decoder used for upsampling. The parameters of upsampling in SegNet are not learned through the training, which can reduce the scale of parameters in the network.

In the research of the network structure, Lin G [23, 34] proposed RefineNet with the refine structure. This structure can extra more features of the images when sampling. Through the fusion of the multiresolution, this method can get the segmentation with higher accuracy. In summary, the main methods of image segmentation includes traditional segmentation, traditional machine learning methods and deep learning methods. However, for semantic segmentation, deep learning methods or some machine learning methods are more suitable.

For the segmentation of remote sensing images, Pabitra Mitra [35] used SVM to do semantic segmentation for spectral remote sensing images and compared with unsupervised learning methods. Bilgin et al. [2] researched the semantic segmentation of multi-spectral remote sensing images with SVM. In the research, they tried the segmentation for multi-classification and get a good result. Paisitkriangkrai S [5] used artificial methods and CNN to extract features at the same time and trained the classification separately with the flowing CRF (Conditional Random Field) to optimize the details. This shew that deep the learning method could get a better result than the traditional machine learning methods. Maggiori E et al [1] used FCN model to do the semantic segmentation of buildings on the public multi-spectral remote sensing images. After that, they also used RNN to improve the results of CNN [3]. To do the multi-classification semantic segmentation, they then tried to use a model similar with DeconvNet [1]. All of these researches have got good results.

Marmanis D et.al [36] used SegNet and FCN to do boundary detection based on remote sensing image semantic segmentation with Deep Convolutional Neural Network. This improved the result of the segmentation. Since the performance of deep learning are much better than that of the traditional machine learning methods, the new researches about the semantic segmentation of remote sensing images mostly focus on training deep learning models and then do some optimized processing to get good results [37, 38].

4 Proposed method

In this research, we improve the Encoder-Decoder CNN structures SegNet with index pooling and U-net to make them suitable for multi-targets semantic segmentation of remote sensing images. In addition, we propose a model integrated algorithm that can integrate the semantic segmentation results of these two different CNN models to make use of their advantages on the different objects' segmentation.

4.1 Pooling index

In ordinary pooling, it will lose some image information and the trans-convolution operation will complements 0 on the edge of the feature map. So the result of the segmentation is not very accurate especially for the edge of the image.

This design of pooling index can store the position information of pooling operation and restore the position of the pixels when up-pooling. When encoding, this can store the coordinate information of the maximum as the pooling index in every step of pooling. And when decoding, the coordinate information can be restored to put the original position on the image. So the detail information of the image can be restored (Figs. 1, 2 and 3).

Fig. 1 Max pooling

Fig. 2 Max pooling with index operation

4.2 Batch normalization

What deep learning need is training a deep network model. The input of each layer of the model is the output of the previous layer, that is, the parameter distribution received by each layer is also the parameter distribution of the output of the previous layer. However, in convolutional neural networks, eigenvalues are activated by activation function at the last step of each layer. Now the most commonly used activation function is ReLU, which can only activate eigenvalues greater than 0, that is, the parameters output by the activation function are values greater than 0. This will lead to two problems. The first one is that each layer must

re-learn the parameter distribution according to the change of the parameter distribution of the previous layer. This does not meet our requirements for distribution stability. The second one is the parameter distribution of the previous layer changes, and this change will be amplified after going through the deep network, which will lead the value of the previous layer cannot be effectively updated in the back-propagation algorithm. This design improved the Batch Normalization with two new parameters scale and shift to solve these problems. Ioffe S [39] proposed the Batch Normalization which can help to control the distribution changes in the deep network. Assume that x is the data sample input in the current layer and this method subtracts

Fig. 3 Upsampling with the stored index

its own mean from each dimension and divides it by its own standard deviation. And then it can get a data distribution that has a mean of 0 and a variance of 1. The formula is as below:

$$\hat{x}^{(k)} = \frac{\hat{x}^{(k)} - E[x^{(k)}]}{\sqrt{\text{Var}(x^{(k)})}}$$

However, such data processing will lead to the decline of data expression ability. So we add two parameters scale and shift to enhance the ability of expression in the network. These two parameters are obtained by training in the network as below:

$$y^{(k)} = \gamma^{(k)} \hat{x}^{(k)} + \beta^{(k)}$$

In this formula, when $\gamma^{(k)}$ is equal to the standard deviation of our previous data sample, and $\beta^{(k)}$ is equal to the mean of our previous data sample, this transformation reverts the data back to its original distribution. Through this the model can get these two advantages. One is that it can increase the classification effect without using Dropout that is used to prevent over-fitting. The cause of model over-fitting is generally at the boundary of data distribution. Fitting puts the initial weight of data into the data, which effectively alleviates the over-fitting phenomenon. The other one is that the higher learning rate can be used to effectively improve the convergence speed of the algorithm. If the data distribution of each layer is inconsistent, in order to reduce the loss function better, a smaller learning rate must be used.

4.3 SegNet

In ordinary pooling, it will lose some image information and the trans-convolution operation will complements 0 on the edge of the feature map. So the result of the segmentation is not very accurate especially for the edge of the image. In this designed structure, the pooling index can store the position information of pooling operation and restore the position of the pixels when up-pooling. When encoding, this can store the coordinate information of the maximum as the pooling index in every step of pooling. And when decoding, the coordinate information can be restored to put the original position to the image. So the detail information of the image can be restored.

The kernels in convolution layers in standard SegNet are always 3×3 . With the padding of 1, they do not change the size of the images. Only the pooling layer and the up-sampling layer scale can restore the image at a ratio of 2. For the remote sensing images, the sizes are always is so large that we need to slice them. This designs the network with 4 Encoders and 4 Decoders and the softmax layer at the end. Each Encoder has 5 convolution combination layers and one pooling layer. Each convolution combination layer has

convolution layer, batch normalization layer and activation layer.

Encoder 1: Each convolution combination layer uses 64 convolution cores of 3×3 size, without padding, to perform convolution operation with step 1. After five convolution operations, the height and width of the image will be reduced by 10 pixels. Then the maximum pooling operation is performed to reduce the height and width of the image by two times.

Encoder 2: Each convolution combination layer uses 128 convolution cores of 3×3 size, without padding, to perform convolution operation with step 1. The size of the image keeps unchanged during the operation. Then the maximum pooling operation is performed to reduce the height and width of the image by two times.

Encoder 3: Each convolution combination layer uses 256 convolution cores of 3×3 size, without padding, to perform convolution operation with step 1. The size of the image keeps unchanged during the operation. Then the maximum pooling operation is performed to reduce the height and width of the image by two times.

Encoder 4: Each convolution combination layer uses 512 convolution cores of 3×3 size, without padding, to perform convolution operation with step 1. The size of the image keeps unchanged during the operation. Then the maximum pooling operation is performed to reduce the height and width of the image by two times.

Decoder 4: Enlarge the image twice with up-sampling to restore to the size before the pooling layer in the corresponding Encoder. After that, the convolution operation in each convolution combination layer is the same as that in Encoder 4.

Decoder 3: Enlarge the image twice with up-sampling. After that, the convolution operation in each convolution combination layer is the same as that in Encoder 3.

Decoder 2: Enlarge the image twice with up-sampling. After that, the convolution operation in each convolution combination layer is the same as that in Encoder 2.

Decoder 1: Enlarge the image twice with up-sampling to restore to the size before the pooling layer in the corresponding Encoder 1. The height and width of the image will be reduced by 10 pixels. Each convolution combination layer uses 64 convolution cores of 3×3 size, without padding, to perform convolution operation with step 1. After five convolution operations, the height and width of the image will be reduced by 10 pixels again.

Network structure extracts features through convolution layer and pooling layer in Encoder stage. The kernels change from large to small, with the dimensions from low to high. In the corresponding Decoder, the size of the image is gradually restored through the up-sampling and convolution layer, and the prediction result of the whole image is obtained in the end.

Since the pooling layer of each Encoder corresponds to the up-sampling layer of Decoder, the size of the image before pooling operation and that after the up-sampling operation must be consistent. So it does not need padding in the first layer of Encoder and Decoder. At each stage, a larger receptive field is obtained by stacking multiple 3×3 convolution kernels.

4.4 U-net

U-Net is based on fully convolutional networks and its main idea is to add successive layers behind conventional convolution networks, which are designed for up-sampling. In order to locate more accurately, up-sampling combines features of upstream. U-net retains a large number of characteristic channels in the up-sampling part, which propagates context information to a higher resolution layer. So the whole structure looks like an “U”.

In this study, the designed structure consists of a contraction path and an expansion path. The contraction path follows the typical architecture of a convolutional network. It includes two repetitive applications of 3×3 convolution. Each convolution is followed by a linear element (ReLU) and a 2×2 max pooling operation for the sampling. The number of feature channels is doubled in each sampling step. Every step of the extended path contains a sample up the feature of the map and then a

2×2 convolution (up-convolution). The convolution will halve the number of the feature channel, and connect the corresponding feature map in the contraction path. Then there are two 3×3 convolution and after each convolution is a ReLU. At the last layer, it uses 1×1 convolution to map the eigenvectors of each 64 component to the desired number of classes (Fig. 4).

4.5 Integrated algorithm

For the semantic segmentation, this research designs the integrated algorithm for the multi-masks of the different models with different parameters. It can make use of the advantages of the different models to do the multi-targets semantic segmentation.

With the different batch size parameters, the SegNet and U-net can get different masks for the segmentation. We create a hole initial zero mask as the original carrier. From the first pixel, each pixel can compare the result of corresponding pixels in the different masks. It calculate all the results predicted from different models and store the statistical results. With this statistical results, it can vote the best label of this pixel and consider this one to be the final prediction and put this pixel on the hole initial zero mask. Continue the next pixel until all the pixels are processed as the first one and finally the hole results can be integrated (Fig. 5). The steps can be shown in Algorithm 1.

Fig. 4 U-Net

Fig. 5 The voting process of each pixel in every mask

Algorithm 1 Integrated algorithm.

Input: masks of different models. $n = 1, \dots, n$

Output: result

```

1: for  $i = 1$  to  $n$  do
2:   add  $i$  to a fusion set  $v$ 
3: end for
4: reshape  $v$ 
5: create the hole initial zero mask  $v_0$ 
6: for  $h = 0$  to  $v(\text{height})$  do
7:   for  $w = 0$  to  $v(\text{width})$  do
8:     make a record of initial vote
9:     for  $n = 0$  to  $n(\text{numbersofmasks})$  do
10:      the pixel in the position  $(h,w)$ 
11:      update the number of pixels for each class
12:    end for
13:    calculate the maximum of the votes
14:  end for
15: end for
16: Save to the final result
17: return result

```

5 Experiments

5.1 Dataset

The data used in the experiment came from the “AI classification and recognition of satellite images” competition

in the 2017 Big Data and Computing Intelligence Contest (BDCI) [40]. It came from a high-resolution remote sensing image of a region in southern China. The spatial resolution of the image is sub-meter and the spectrum is visible light band (R,G,B). This experiment sets three of the five images as training set and two of them as test set. The size of the training data is 5664×5142 , 3357×6116 and 4011×2470 and the size of the two test images are the same: 7969×7939 . The training data is annotated as five classes: Vegetation, road, building, water, and others (Fig. 6).

As the size of the aerial images are usually very big, we can not put the images into the neural network. We should cut them into random cutting. This can randomly generate x, y coordinates, and then extract the small 256×256 small pictures under the coordinates, and do the following data enhancement operations to enlarge the training data to avoid over-fitting:

- The original and labeled images should be rotated: 90 degrees, 180 degrees, 270 degrees.
- Do the mirroring operations along the Y axis both on the original and labeled images.
- Do the fuzzy operations on the original images.
- Do the light adjustment operation on the original images.
- Do the increase noise operation on the original images (Gauss noise, Salt-and-Pepper noise).

After the operation, we will get a larger training set with 15000 images. After this 75% of these are set as training data and 25% as validation set (Fig. 7).

Fig. 6 Dataset

5.2 Training the models

In this experiment, we set the epoch as 30 and the batch size as 4 and store the best model each time and draw the loss-accuracy at the end of the training. The training process is shown in Figs. 8 and 9.

Then we set the batch size as 8 and the keep the same epochs to do the segmentation to get different masks. We train the model with these parameters and the training process is shown in Figs. 10 and 11.

5.3 Evaluation criteria

When the segmentation results are obtained, we need to analyze and evaluate the segmentation results of the model by calculating the overall segmentation accuracy and the segmentation accuracy of each category. The overall segmentation accuracy refers to the percentage of the number of pixels correctly segmented in the total number of

pixels in the sample of the whole test set. This is shown as the formula below, N is the number of overall pixels and n is that of each class:

$$OA = \frac{\sum_{i=1}^n c_{ii}}{N} \times 100\%$$

The segmentation accuracy of each class refers to the percentage of pixels accurately segmented in each segmentation category in the total number of pixels of the category. CA_i is the segmentation accuracy of class i , and c_{ij} represents the number of class i pixels wrongly segmented to class j pixels. The formula is shown in this formula:

$$CA_i = \frac{c_{ii}}{\sum_{j=1}^n c_{ij}} \times 100\%$$

Fig. 7 Data preprocessing

6 Results and discussion

When the batch size is 4, the results of SegNet and U-net are shown as Tables 1 and 2. SegNet can get a higher overall accuracy than U-Net, especially on road and building segmentation. However, for vegetation, unet is better than segnet. For other classes, there is little difference between them.

Fig. 8 The training of SegNet

We set the batch size as 8 and the keep the same epochs to do the segmentation with the integrated algorithm. The result is shown in Table 3.

With the model integrated algorithm, the overall accuracy can reach 90.2%, higher than that of the single model 88.1% and 87.0%. For the segmentation of vegetation, the accuracy of U-Net is 89.3%, higher than that of SegNet whose accuracy is 85.4%. However for the roads and building,

Fig. 9 The training of U-net

Fig. 10 The training of SegNet with BN = 8

Fig. 11 The training of U-net with BN=8

Table 1 The result of segnet

Class	Ground pixels	Correctly predicted	Accuracy
Vegetation	16730894	13467364	85.4%
Road	11690964	11602120	90.3%
Building	52165026	42202692	89.6%
Water	5341168	2903998	93.2%
Others	40603730	41352285	86.7%
Overall Accuracy			88.1%

Table 2 The result of U-net

Class	Ground pixels	Correctly predicted	Accuracy
Vegetation	16730894	14082384	89.3%
Road	11690964	11101031	86.4%
Building	52165026	40177340	85.3%
Water	5341168	2885303	92.6%
Others	40603730	41877546	87.8%
Overall Accuracy			87.0%

Table 3 The results of the integrated algorithm

Class	Ground pixels	Correctly predicted	Accuracy
Vegetation	15769747	14161233	89.8%
Road	12848416	11756301	91.5%
Building	47101219	42720806	90.7%
Water	3115878	2916462	93.6%
Others	47696522	42545298	89.2%
Overall Accuracy			90.2%

Table 4 The comparison of the results of MSVSF and integrated algorithm

Class	MSVSF	Integrated Algorithm
Vegetation	95.14%(trees)	89.8%
Road	88.52%	91.5%
Building	80.86%	90.7%
Water	—	93.6%
Overall Accuracy	87.99%	90.2%

the accuracy of SegNet can reach 90.3% and 89.6%, much higher than that of U-Net 86.4% and 85.3%. For the model integrated algorithm, the accuracy of roads and building can reach 91.5% and 90.7%. The result of model integrated algorithm can make use of the advantages of these two models. After the fusion of the advantages of the two models, the results of each class can be improved.

Rongjun Qin [41] proposed the method of mean shift vector-based shape feature (MSVSF) with multi-spectral channel to do multi-targets semantic segmentation in 2015. In the Table 4, we compare this result with that of MSVSF.

The overall accuracy of our proposed integrated algorithm for Road and Building is higher than that of MSVSF except for the vegetation segments. In MSVSF, the segmentation accuracy of vegetation is higher, because it uses spectral information. The spectral information of forest and

grassland are more obvious in remote sensing images so the segmentation accuracy is relatively high. However, MSVSF uses four channels and multi-spectral information, which is a method that needs to process the complex spectral information before the segmentation [41]. So the model integrated algorithm has more advantage on the semantic segmentation of remote sensing images (Fig. 12).

7 Conclusions

In this paper, we presented an improved CNN structure SegNet and U-net and applied to the multi-target semantic segmentation of remote sensing images. We analyzed the results and compared the advantages and disadvantages of each models on the segmentation of different objects. With

Fig. 12 The visualization of the results

pooling index and based on the structure of SegNet, we have designed the Encoder-Decoder CNN structure that can restore more information of images when up-sampling. This improved SegNet can be used for multi-targets semantic segmentation of re- mote sensing images and has got a good result for this segmentation task. With pooling index and based on the structure of SegNet, we have designed the Encoder-Decoder CNN structure that can restore more information of im- ages when up-sampling. This improved SegNet can be used for multi-targets semantic segmentation of remote sensing images and has got a good result for this segmentation task. In addition, we proposed a model integrated algorithm that can integrate the semantic segmentation results of these two different CNN models. We implemented the experiment with this algorithm and analyzed the results. The proposed integrated algorithm exploit the advantages of these two models and improved the segmentation. We compared the results with previous work and showed that the proposed integrated algorithm shows better results which further validates our method (Fig. 12).

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Maggiori E, Tarabalka Y, Charpiat G, Alliez P (2016) Fully convolutional neural networks for remote sensing image classification. In: Geoscience remote sensing symposium
2. Bilgin G, Erturk S, Yildirim T (2011) Segmentation of hyperspectral images via subtractive clustering and cluster validation using one-class support vector machines. *IEEE Trans Geosci Remote Sens* 49(8):2936–2944
3. Maggiori E, Tarabalka Y, Charpiat G, Alliez P (2017) Recurrent neural networks to enhance satellite image classification maps. *IEEE Trans Geosci Remote Sens* PP(99):1–10
4. Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A (2015) Going deeper with convolutions:1–9
5. Paisitkriangkrai S, Sherrah J, Janney P, Hengel VD (2015) Effective semantic pixel labelling with convolutional networks and conditional random fields. In: Computer Vision Pattern Recognition Workshops
6. Chen L-C, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2014) Semantic image segmentation with deep convolutional nets and fully connected crfs, arXiv:1412.7062
7. Johnson M, Shotton J, Cipolla R (2008) Semantic textron forests for image categorization and segmentation. *Proc IEEE Cvp* 5(7):1–8
8. Fukushima K (1980) Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol Cybern* 36(4):193–202
9. Ohtsu N (2007) A threshold selection method from gray-level histograms. *IEEE Trans Sys Man Cybern* 9(1):62–66
10. Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: International Conference on Neural Information Processing Systems
11. Bosch A, Zisserman A, Munoz X (2007) Image classification using random forests and ferns. In: IEEE International Conference on Computer Vision
12. Zhao W, Du S, Qiao W, Emery WJ (2017) Contextually guided very-high-resolution imagery classification with semantic segments. *ISPRS J Photogramm Remote Sens* 132:48–60
13. Nair V, Hinton GE (2010) Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th international conference on machine learning (ICML-10), pp 807–814
14. Zhang X, Zhao J, LeCun Y (2015) Character-level convolutional networks for text classification. *Adv Neural Inf Process Syst* 2015:649–657
15. Graves A, Mohamed AR, Hinton G (2013) Speech recognition with deep recurrent neural networks. In: IEEE international conference on acoustics, speech and signal processing (pp. 6645–6649)
16. Hoque MM, Quaresma P (2016) A semantic-based technique for question lassification in question answering systems — a hybrid approach. In: International Conference on Computer Information Technology
17. Wiesel TN, Hubel DH (1963) Single-cell responses in striate cortex of kittens deprived of vision in one eye. *J Neurophysiol* 26(6):1003–1017
18. Chen LC, Zhu Y, Papandreou G, Schroff F, Adam H (2018) Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European conference on computer vision (ECCV) (pp. 801–818)
19. Kim Y, Jernite Y, Sontag D, Rush AM (2016) Character-aware neural language models. In: Thirtieth AAAI Conference on Artificial Intelligence
20. He K, Zhang X, Ren S, Sun J (2015) Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision, pp 1026–1034
21. Huang G, Liu Z, Van Der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4700–4708
22. Carreira J, Sminchisescu C (2012) Cpmc: Automatic object segmentation using constrained parametric min-cuts. *IEEE Trans Pattern Anal Mach Intell* 34(7):1312–1328
23. Belongie S, Malik J, Shi J, Leung T (1998) Image and video segmentation: the normalized cut framework. In: International conference on image processing
24. Freedman D, Zhang T (2005) Interactive graph cut based segmentation with shape priors. In: IEEE Computer Society Conference on Computer Vision Pattern Recognition
25. Haris K, Efstratiadis SN, Maglaveras N, Katsaggelos AK (2002) Hybrid image segmentation using watersheds and fast region merging. *IEEE Trans Image Process* 7(12):1684–1699
26. Hou L, Samaras D, Kurc TM, Gao Y, Davis JE, Saltz JH (2016) Patch-based convolutional neural network for whole slide tissue image classification. In: Computer Vision Pattern Recognition

27. Zhou B, Zhao H, Puig X, Xiao T, Fidler S, Barriuso A, Torralba A (2019) Semantic understanding of scenes through the ade20k dataset. *Int J Comput Vis* 127(3):302–321
28. Zhang W, Li R, Deng H, Wang L, Lin W, Ji S, Shen D (2015) Deep convolutional neural networks for multi-modality isointense infant brain image segmentation. *Neuroimage* 108:214–224
29. 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. Springer, pp 234–241
30. Chen L-C, Papandreou G, Kokkinos I, Murphy K, Yuille AL (2018) Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Trans Pattern Anal Mach Intell* 40(4):834–848
31. Noh H, Hong S, Han B (2015) Learning deconvolution network for semantic segmentation. In: IEEE International Conference on Computer Vision
32. Yang J, Price B, Cohen S, Lee H, Yang MH (2016) Object contour detection with a fully convolutional encoder-decoder network. In: Computer Vision Pattern Recognition
33. Badrinarayanan V, Kendall A, Cipolla R (2017) Segnet: A deep convolutional encoder-decoder architecture for scene segmentation. *IEEE Trans Pattern Anal Mach Intell* PP(99):1–1
34. Lin G, Milan A, Shen C, Reid I (2017) Refinenet: Multi-path refinement networks for high-resolution semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1925–1934
35. Mitra P, Shankar BU, Pal SK (2004) Segmentation of multispectral remote sensing images using active support vector machines. *Pattern Recogn Lett* 25(9):1067–1074
36. Marmanis D, Schindler K, Wegner JD, Galliani S, Datcu M, Stilla U (2017) Classification with an edge: improving semantic image segmentation with boundary detection. *Isprs J Photogramm Remote Sens* 135:158–172
37. Chen LC, Papandreou G, Schroff F, Adam H (2017) Rethinking atrous convolution for semantic image segmentation, arXiv:1706.05587
38. Mas JF, Flores JJ (2008) The application of artificial neural networks to the analysis of remotely sensed data. *Int J Remote Sens* 29(3):617–663
39. Ioffe S, Szegedy C (2015) Batch normalization: Accelerating deep network training by reducing internal covariate shift, arXiv:1502.03167
40. BDCI (2017) Big data and computing intelligence contest. <https://www.datafountain.cn/competitions/270/details/>
41. Qin R (2015) A mean shift vector-based shape feature for classification of high spatial resolution remotely sensed imagery. *IEEE J Sel Top Appl Earth Obs Remote Sens* 8(5):1974–1985

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.