IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received June 12, 2018, accepted July 10, 2018, date of publication July 13, 2018, date of current version August 7, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2856088

An End-to-End Neural Network for Road Extraction From Remote Sensing Imagery by Multiple Feature Pyramid Network

XUN GAO[1,2], (Student Member, IEEE), XIAN SUN[1], YI ZHANG[1], MENGLONG YAN[1], GUANGLUAN XU[1], HAO SUN[1], JIAO JIAO[1,2], (Student Member, IEEE), AND KUN FU[1,2,3]

[1]Key Laboratory of Technology in Geo-Spatial Information Processing and Application System, Institute of Electronics, Chinese Academy of Sciences, Beijing 100190, China
[2]School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 100049, China
[3]Institute of Electronics, Chinese Academy of Sciences, Suzhou 215123, China

Corresponding author: Xian Sun (sunxian@mail.ie.ac.cn)

This work was supported in part by the National Natural Science Foundation of China under Grants 61725105 and 41501485, and in part by the Gusu Innovation Talent Foundation of Suzhou under Grant ZXT2017002.

ABSTRACT Unlike single geospatial objects extraction from high-resolution remote sensing images, the task of road extraction faces more challenges, including its narrowness, sparsity, diversity, multiscale characteristics, and class imbalance. Focusing on these challenges, this paper proposes an end-to-end framework called the multiple feature pyramid network (MFPN). In MFPN, we design an effective feature pyramid and a tailored pyramid pooling module, taking advantage of multilevel semantic features of high-resolution remote sensing images. In the optimization stage, a weighted balance loss function is presented to solve the class imbalance problem caused by the sparseness of roads. The proposed novel loss function is more sensitive to the misclassified and the sparse real labeled pixels and helps to focus on the spare set of hard pixels in the training stage. Compared with the cross-entropy loss function, the weighted balance loss can reduce training time dramatically for the same precision. Experiments on two challenging datasets of high-resolution remote sensing images which illustrate the performance of the proposed algorithm have achieved significant improvements, especially for narrow rural roads.

INDEX TERMS Multiple feature pyramid network (MFPN), feature pyramid, pyramid pooling, weighted balance loss.

I. INTRODUCTION

In recent years, road extraction from high-resolution remote sensing images has been applied in many domains, e.g., urban planning, Geographic Information System (GIS) data updating, and traffic navigation. Ideally, roads have regular shape in object extraction since they have obvious geographical features [1], including strip-like distribution, uniformity of gray distribution, geometric shapes of fixed width, and interconnected network topologies. However, as shown in Fig. 1, the main difficulties in road extraction from remote sensing images are as follows: (1) Diversity. Types of roads include highways, urban trunk roads, and country roads, resulting in multiscale characteristics. (2) Narrowness. Compared with massive objects such as buildings, roads are narrow, likely to cause discontinuous extraction (Fig. 8, columns 3 and 4). (3) Sparsity. In rural areas, roads are a sparse target compared to vegetation and farmland (Fig. 1c), leading to the challenge of class imbalance. (4) Easily disturbed. The texture of roads in remote sensing images is easily obscured by trees (Fig. 1d) or confused with rivers (Fig. 1c), causing feature variation in different imaging conditions. Therefore, extracting roads from remote sensing images automatically and precisely is rather tough work.

In order to handle the extraction task, different methods have been proposed to cope with these challenges. Some primary traditional methods based on unsupervised learning, such as [2]–[12], try to use the inherent information of the images, including color, texture, and boundary. Nevertheless, the spectral or inherent properties of roads in remote sensing images are usually mixed up with other disturbances, such as shadows, traces of water, and light. Recently, there is a growing body of literature that recognizes the importance of

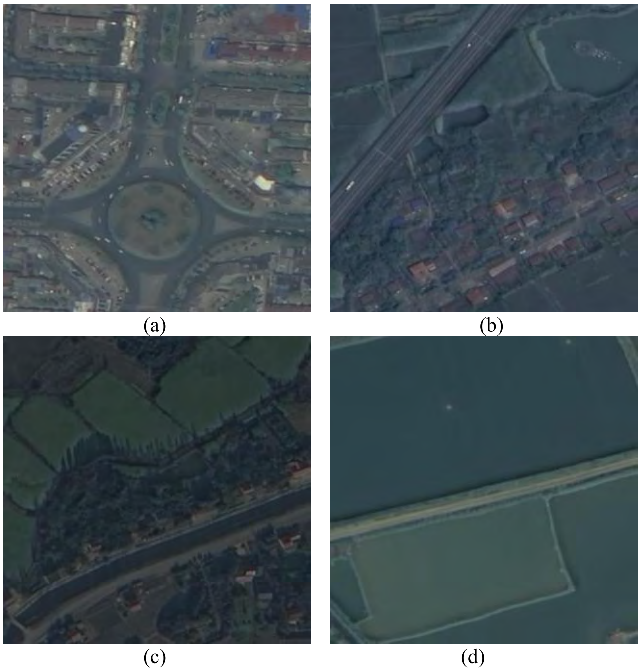


FIGURE 1. Samples of roads in different scenarios. (a) roads in urban; (2) roads in rural; (c) roads is confused with rivers; (d) roads is obscured by trees.

robust features. Many researchers are paying more attention to the use of modern deep convolutional neural networks (DCNNs) [13], which tend to significantly improve performance. Some commonly used models have been achieving state-of-the-art performance not only in computer vision, such as fully convolutional network (FCN) [14], deconvolutional network [15], SegNet [16], DeepLab [17]–[19], integrated CNN with conditional random fields (CRFs) [17], [20], [21], and pyramid scene parsing network (PSPNet) [22], but also in remote sensing road extraction, e.g. [23], [24] and Road Structure Refined CNN (RSRCNN) [30].

Among the above prevailing methods of DCNN, common to these architectures is the use of all convolutional layers, replacing the last fully connected layers in classification with the convolution layer, which is more conducive to assigning a category label to each pixel. However, road extraction based on DCNN has a couple of critical limitations. First, the network has a too-large receptive field for smaller objects. With deepening of the network, the spatial size of the receptive field on feature maps is gradually increasing. Although theoretically this indicates how much we use context information, in fact, an object that is larger or smaller than the receptive field may be mislabeled, especially for narrow roads, as shown in Fig. 1c. Meanwhile, for narrow and sparse roads, there is less target information after several pooling layers. Even though the high-level semantic information is abundant, there is a lack of low-level location information for sparse targets. Second, the traditional cross entropy (CE) loss function is not suitable for optimizing sparse scenes. Different from the object detection task, road extraction mainly focuses on assigning a category label to each pixel, while the detection task is only needed to identify several different objects. So, we argue that the different category distribution will result in different optimization spaces. Assuming CE loss is still used, it pays equal attention to all pixel points. After several iterative optimizations, the vast number of background pixels will gradually lead the center of gravity to the background instead of roads. Therefore, we need an efficient loss function to deal with the problem of unbalanced categories caused by road sparsity.

To overcome such limitations, we propose a new end-to-end multiple feature pyramid network (MFPN) based on PSPNet. Unlike PSPNet, we add a novel module called feature pyramid and customized a pyramid pooling structure for road extraction, which has the ability to address the first limitation Feature pyramid [41] is a top-down multi-scale feature fusion structure combining low-level location information and high-level abundant semantic features. It can make up for the poor effect of PSPNet on multi-scale road extraction. Inspired by PSPNet, the tailored pooling pyramid module (TPPM), which is designed according to the strip-like shapes of roads, can fuse the contextual information of different subregions with different scales. These two improvements in the network architecture have excellent prediction capability suitable for the extraction of country roads. In addition, our weighted balance loss function focuses on pixels that are misclassified and sparse real labels. It can save computing resources and training time in maintaining prediction accuracy. Experiments on public datasets and our own datasets prove the competitive performance of our MFPN.

The remainder of this paper is organized as follows: We first review some related works in Section 2 and describe in detail the architecture of the proposed network in Section 3. Experimental datasets and evaluations are described in Section 4. The experimental results are presented in Section 5. Finally, Section 6 summarizes the findings.

II. RELATED WORK

Study of road extraction from remote sensing images has lasted for 30 years so far. In this section, we discuss the technological innovations from traditional methods related to machine learning as applied to deep learning.

Traditional road extraction methods tried to use the inherent features of roads. Tupin *et al.* [2] used a liner detector to extract road candidate segments, using the Markov random field (MRF) classifier to select and connect the real road segments. Wang and Luo [3] proposed a semiautomatic road extraction method using the Markov random texture and support vector machine (SVM) classifier. Based on spatial and spectral information, the SVM classifier was used in [4] and [5]. All of these methods are based on manually designed features of roads, relying on human intervention. Therefore, there is a need for a practical algorithm that can automatically extract robust features from the whole remote sensing image, not just the characteristics of the road itself.

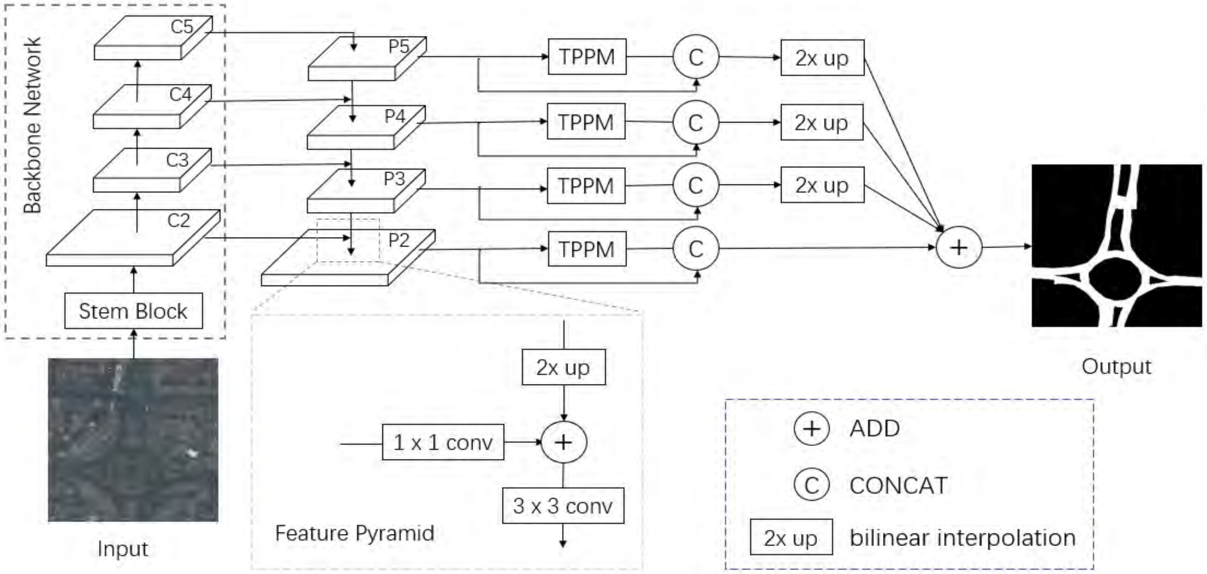


FIGURE 2. Overall framework of our model. TPPM, tailored pyramid pooling module; shown in Fig. 3.

Many recent works have already paid attention to the DCNN technique, which tends to extract robust features. In [23] and [24], FCN was also incorporated to automatically learn features of roads, and then to make decisions on road regions. Mnih [25] proposed a method based on patch-based CNN. It used principal component analysis (PCA) features as the CNN input, which were trained by the restricted Boltzmann machine (RBM). By using SegNet followed by two postprocessing techniques, landscape metrics (LMs) and conditional random fields (CRFs), Panboonyuen *et al.* [26] presented an enhanced DCNN framework. Alshehhi *et al.* [27] also presented an effective model utilizing patch-based CNN. In order to acquire fine prediction, simple liner iterative clustering (SLIC) [28] is used to obtain the initial image and region adjacent graph (RAG) [29] is applied to facilitate merging between super-pixels. By changing the architecture of the network, the above methods could obtain satisfactory performance. However, they are still multistage and depend on preprocessing or postprocessing.

The most related work for us is RSRCNN [30], which also built an end-to-end framework and designed a novel loss function. Different from the methods mentioned above, RSRCNN was an end-to-end framework that is able to advance the state-of-the-art road extraction for aerial images in the Massachusetts Roads Dataset by using deconvolutional and fusion layers in DCNN. The fusion layers of RSRCNN directly combined the convolutional and deconvolutional layers with pixel-wise summing. It may result in falsely detected roads due to a strong edge in the early layer of CNN. To relieve this problem, Wei *et al.* [30] proposed a road structure–based loss function, which incorporated the geometric information of road structure in cross-entropy loss. Note that this is not a direct solution to the problem and leaves behind the problem of class imbalance imposed by road sparsity. Therefore, we used feature pyramid to mitigate the edge effects of fusion and propose a novel weighted balance loss function to cope with the huge class imbalance.

III. METHODS

In this section, the proposed end-to-end framework for road extraction from high-resolution remote sensing images is illustrated. We first introduce the MFPN architecture and each component, followed by our efficient loss function. Then we describe the training setting.

A. *MFPN ARCHITECTURE*

1) OVERALL FRAMEWORK

As shown in Fig. 2, the proposed MFPN method is a single end-to-end network composed of a backbone network and two feature-processed subnetworks. The backbone network is responsible for overcoming the problem of road diversity, computing a robust convolutional feature map by a modified 101-layer ResNet [40]. The first subnet is a feature pyramid that can be well suited for multiscale scenarios by effectively integrating multilevel semantic information. The second subnet uses a tailored pooling pyramid module able to aggregate contextual information, to get over the disturbance of external factors on the road. We will elaborate on each component below.

2) BACKBONE NETWORK

To cope with the diversity of roads, we need feature extractors with generalization capabilities. The model should be adaptable to different types of roads, to acquire a robust feature map through the backbone network. Here, different from the original ResNet, which included a stem block and

TABLE 1. The architecture of backbone network.

Layers		Backbone Network	Output Size (Input 3 x 320 x 320)
Stem Block	Convolution	3 x 3 conv, stride 2	64 x 160 x 160
	Convolution	3 x 3 conv, stride 1	64 x 160 x 160
	Convolution	3 x 3 conv, stride 1	128 x 160 x 160
	Pooling	3 x 3 max pool, stride 2	256 x 80 x 80
C2		$\begin{bmatrix}1\times 1, \text{stride } 1, 64\\ 3\times 3, \text{stride } 1, 64\\ 1\times 1, \text{stride } 1, 256\end{bmatrix}\times 3$	256 x 80 x 80
C3		$\begin{bmatrix}1\times 1, \text{stride } 2, 128\\ 3\times 3, \text{stride } 1, 128\\ 1\times 1, \text{stride } 1, 512\end{bmatrix}$ $\begin{bmatrix}1\times 1, \text{stride } 1, 128\\ 3\times 3, \text{stride } 1, 128\\ 1\times 1, \text{stride } 1, 512\end{bmatrix}\times 3$	512 x 40 x 40
C4		$\begin{bmatrix}1\times 1, \text{stride } 1, 256\\ 3\times 3, \text{hole rate } 2, 256\\ 1\times 1, \text{stride } 1, 1024\end{bmatrix}\times 23$	1024 x 40 x 40
C5		$\begin{bmatrix}1\times 1, \text{stride } 1, 512\\ 3\times 3, \text{hole rate } 4, 512\\ 1\times 1, \text{stride } 1, 2048\end{bmatrix}\times 3$	2048 x 40 x 40

four residual blocks, C2, C3, C4, and C5, we use a refined ResNet to extract robust features from remote sensing images. First, inspired by Inception v3 [45] and v4 [44], we modify the stem block with a stack of three 3 × 3 convolution layers followed by a 3 × 3 max pooling layer. The stride of the first convolution layer is 2 and of the other two layers is1, so the size of the output feature maps is one-quarter of the input images. Compared with the original stem block in ResNet (7 × 7 convolution layer with stride 2 followed by 3 × 3 max pooling with stride 2), our design can reduce half the parameters and information loss from raw input images. Second, we enhance the feature generalization of diverse roads by maintaining the spatial sizes of the feature map. To obtain a fine result for road extraction requires that the spatial sizes of the last feature map should be close to the input image. We use atrous convolution [17] in the residual blocks to keep the size. The atrous convolution not only can keep the constant size of the feature map, but also increase the size of receptive fields and enrich semantic information. In our design, only the first conv-layer in C3 uses a convolution layer with stride 2, and two residual blocks, C4 and C5, use atrous convolution with hole rate (2,4). Therefore, the size of a remote sensing image after stem block with four residual blocks is 1/4,1/4,1/8,1/8, and 1/8, respectively, corresponding to the input images. The detailed structure of the backbone network is presented in Table 1.

3) FEATURE PYRAMID

In rural areas, the results of road extraction are often discontinuous. The main reason is that the features for narrow roads tend to disappear with a deeper network, resulting in fragmented forecast results, especially for country roads. This is also a common problem for algorithms extracting smaller targets. Here, we solve the multiscale problem by applying a feature pyramid. As shown in Fig. 2, we use a top-down pathway and lateral connections to construct a multiscale, rich feature pyramid. It combines low-level road location information with high-level abundant semantic features to cope with the gradual disappearance of useful information. In brief, the lateral connections are a 1 × 1 convolution layer applied to C2, C3, C4, and C5, acquiring four new enhanced feature maps, L2, L3, L4, and P5, which can efficiently strengthen the bottom-up features. As this indicates, the operation of the top-down pathway is to merge the adjacent level information by element-wise addition in L2, L3, L4, and P5, followed by a 3 × 3 convolution layer to get the final feature map. Owing to C5, C4, and C3 having the same spatial size in our network design, we directly get P4 by element-wise addition to P5 and L4, then acquire P3 by adding P4 and L3. For the fusion of L2 and P3, we get P2 according to the above operation after a coarse spatial upsampling of P3. The final feature maps we get are P2, P3, P4, and P5, corresponding to C2,C3,C4, and C5, respectively, with the same spatial sizes. Each level of the pyramid can be used for extraction of roads at a different scale. In Section 5, we will show that the benefit of this feature pyramid is significant for extraction performance.

4) TAILORED PYRAMID POOLING MODULE

Road extraction from remote sensing images can be viewed as a task of pixel-level classification of sparse objects. Meanwhile, roads in remote sensing images encounter many disturbances, such as illumination and obstruction by trees. Therefore, the useful information contained in each pixel is extremely important for pixel-level classification. There is no doubt that contextual information is of great importance to get a fine result. So, in [22], a pyramid pooling module was applied to semantic segmentation tasks, which achieved amazing results. In the original design, the pyramid pooling module consisted of four average pooling layers of different size. One of them generates a single bin output through global pooling, and the others use different kernel size to get pooling results in different subregions with bin sizes of 2×2, 3×3, and 6×6, respectively, producing different contextual information. However, for sparse road targets, this is still not enough. According to our statistics, in a remote sensing road image of 320 × 320 pixels, the widest road is only 50 pixels wide. Most of the common roads vary from 20 pixels to 40 pixels. Some of the smaller roads have smaller width, less than 15 pixels. The part of narrow roads is one of the most tough challenge, so we draw lessons from PSPNet and then design the tailored pyramid pooling module to focus on it. As shown in Fig.3, in the pyramid pooling module behind P2, P3, P4, and P5, two average pooling layers were added with smaller rectangular kernel size in different directions, adapting to the strip-like shape of roads. For example, in last feature map of 80 × 80 pixels, the narrow roads are mapped to <3 pixels after several sampling layers. We consider that the shape of the road is narrow and not horizontal or vertical on the image, and we add two extra average pooling layers with kernel size 5 × 80 and 40 × 5 in the pyramid pooling module. This strategy offers a good trade-off between accuracy and speed.

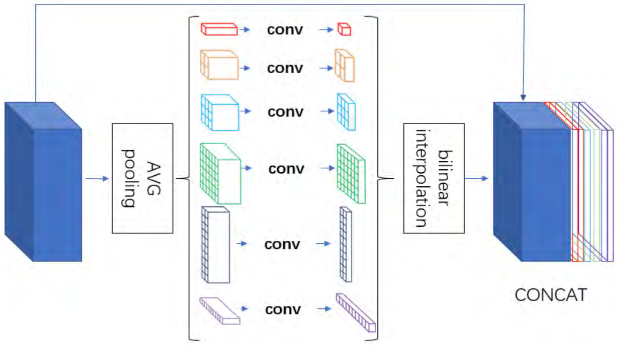


FIGURE 3. Tailored pyramid pooling module.

It is worth mentioning that the pyramid pooling module is beneficial to the extraction of sparse scenes. After a deep convolution network, only few subregion on the feature map contain less valid information on the sparse objects. Thus, if we use the feature map directly to make predictions, it is easy to get fragmented prediction results. In contrast, the pyramid pooling module divides the feature map into several different subregions by average pooling layer, using different kernel sizes. A subregion is likely to contain the sparse object information we need. Then it will be upsampled to be the same size feature as the original feature map via bilinear interpolation. In this process, not only the image context information is included, but representation information on the sparse object is further increased.

B. WEIGHTED BALANCE LOSS

1) LOSS FUNCTION

Regarding either object detection, like the series of RCNN [31]–[33], [34], [46], or semantic segmentation, like FCN and DeepLab, class imbalance is an important problem during training. Due to many easy samples that contribute no useful information, training will be easily inefficient. Furthermore, vast numbers of background examples may lead to local optimization. Different from hard negative mining [35]–[38] and focal loss [39], we design a weighted balance loss to solve the extreme class imbalance caused by road sparsity, which can suppress the bad influence of background pixels by dynamically adjusting weights.

We design the weighted balance loss function based on the following principles:

(a) Focusing on the rare class (foreground). In sparse scenes, the limited information inherently contained in pictures is very important for our optimization. To enhance the feature representation of the rare class, we shall be obliged to pay more attention to it in our loss function.

(b) Suppressing the bad influence of background to total loss. The frequent class (background) can dominate total loss, causing local optimization spaces, further resulting in a lot of waste of computing resources. Here, we use a self-adjusting weight to reduce the loss produced by background.

(c) Punishing the misclassified pixels. Ideally, we want to predict every pixel correctly, but this is just an expectation. Therefore, our loss function should try hard to rectify this error, and a simple way is to put it on the same status as foreground.

In brief, our weighted balance loss on the basis of the cross-entropy loss CE, which is defined by

$$\mathrm{CE} = \sum_{i,j} (y_{i,j} \log p_{i,j} + (1 - y_{i,j}) \log (1 - p_{i,j})) \qquad (1)$$

Assume that $\{y_{i,j} | 1 \le i \le h, 1 \le j \le w\}$ indicates the ground truth of input images, with a size of h × w × c, where h and w are spatial dimensions and c is the channel dimension. Note that $y_{i,j} = 1$ means that the pixel at location (i, j) in the image belongs to road and $y_{i,j} = 0$ stands for background. Meanwhile, $p_{i,j}$ represents the probability that the (i, j) *th* pixel of input image is predicted to be road, which can be calculated by the following softmax function:

$$p_{i,j} = \frac{e^{z_{i,j,k=1}}}{e^{z_{i,j,k=0}} + e^{z_{i,j,k=1}}} \qquad (2)$$

where $z_{i,j,}$ is the output vector at location (i, j) in the last features map with two output channels, road and background.

By observing the definition of CE in Equation (1), we can find that the CE loss pays equal attention to different pixels, failing to consider class imbalance. Following the three principles mentioned above, our weighted balance loss can be written as:

$$\begin{aligned} \mathrm{L}\,(\mathrm{W}) &= \sum_{n=1}^{N} l\,(W_n, CE_n) \\ &= \sum_{n=1}^{N} W_n * CE_n \\ &= \sum_{n=1}^{N} \sum_{i,j} (w_1^n y_{i,j} \log p_{i,j} + w_2^n (1 - y_{i,j}) \log (1 - p_{i,j})) \end{aligned} \tag{3}$$

where N is the number of training batch sizes. $W_n * CE_n$ ($n = 1, 2 \ldots, N$) indicates the loss of a single image, calculated by assigning weight to different pixels, where w_1^n denotes the weight of misclassified and sparse real pixels and w_2^n controls the contribution of background to total loss. In our loss function, w_1^n and w_2^n are expressed as follows:

$$w_1^n = 1, \forall\, pixel \in \{misclassified,\ ground\ truth \tag{4}$$

$$w_2^n = \max\left(T, p_{i,j}\right), \quad \forall\, pixel \in \{others\} \tag{5}$$

where T is the manual threshold to decide how much each background pixel contributes to total loss. Note that $p_{i,j}$ is always less than 0.5 in $\{others\}$, since it represents the probability that pixels are predicted road. So if we set T to be greater than 0.5, w_2^n is a constant, otherwise the value of w_2^n will change with $p_{i,j}$.

Now, each pixel has its own weight to distinguish its impact on total loss. Naturally, foreground pixels receive more attention than background pixels by applying self-adjusting weight. The setting of threshold T not only restrains the background loss, but, more importantly, can stabilize the training process in early training. We will explore this in detail below.

2) CLASS IMBALANCE AND w_2^n INITIALIZATION

During the training, we find a problem caused by instability in initializing the weight of the loss function. Each step of the training will directly affect the prediction of the next batch of images. Due to the existence of w_2^n, we focus more on optimizing misclassified pixels and real road categories in the current step. In the next step, most of the road pixels are located at previously unoptimized positions, which leads to instability in early training. To counter this, we limit the initial value of w_2^n by applying the threshold T to avoid training instability, caused by the large disparity in contributions to the total loss between foreground and background. In other words, $p_{i,j}$ is very small due to easily predicting background pixels and w_2^n is linearly related to $p_{i,j}$ if we do not set the threshold T. The small w_2^n will lead to unstable training because of the uneven contribution of foreground and background to total loss.

C. TRAINING SETTING

We implement our model based on the TensorFlow framework and all experiments are executed on a computer with a Tesla P100 GPU. As common practice, the MFPN model is initialized with pretrained ResNet-101 on the ImageNet dataset. The training is carried out by optimizing the weighted balance loss function using MomentumOptimizer with momentum of 0.9. Inspired by Chen *et al.* [17] and Liu *et al.* [42], we use a "poly" learning rate policy where the initial learning rate of 0.0005 was multiplied by $(1 - \frac{iter}{\max\_irer})^{power}$ with power $=$ 0.9. We adopte the L2-loss with weight decay of 0.0001 and batch normalization technique [43] with a mini-batch size of 16. Other experimental details will be given in the next section.

IV. EXPERIMENTS DATASETS AND EVALUATION

To verify the competitive performance of the end-to-end MFPN model, several experiments on road extraction from remote sensing images are carried out on two different datasets, the Massachusetts and RSI datasets. All controlled experiments are evaluated based on precision, recall, F-measure and, mIoU.

A. MASSACHUSETTS DATASET

This dataset was collected by Mnih [25], and includes 1108 training, 14 validation, and 49 testing images. Each image is 1500 × 1500 pixels in size with a spatial resolution of 1 meter per pixel, composed of tri-band information (red, green, and blue channels). In addition, the dataset is aerial imagery and covers a total area of more than 2634 square kilometers, including urban, suburban, and countryside regions. Samples of this dataset are shown in Fig. 4.

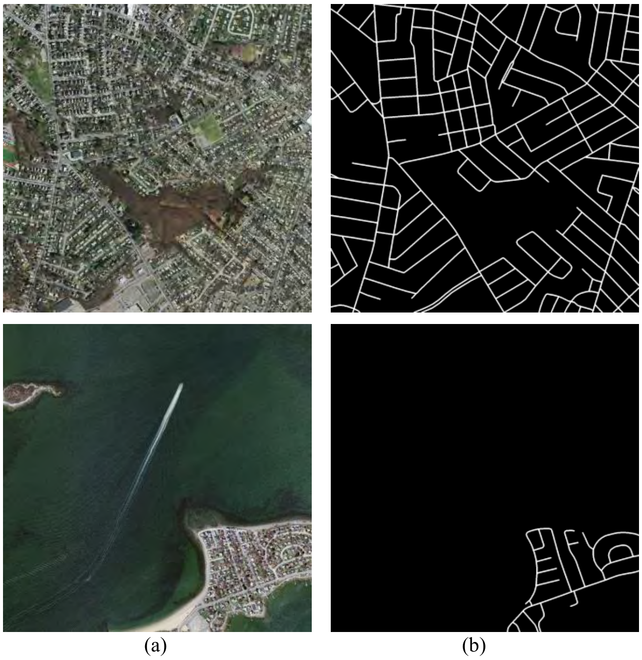
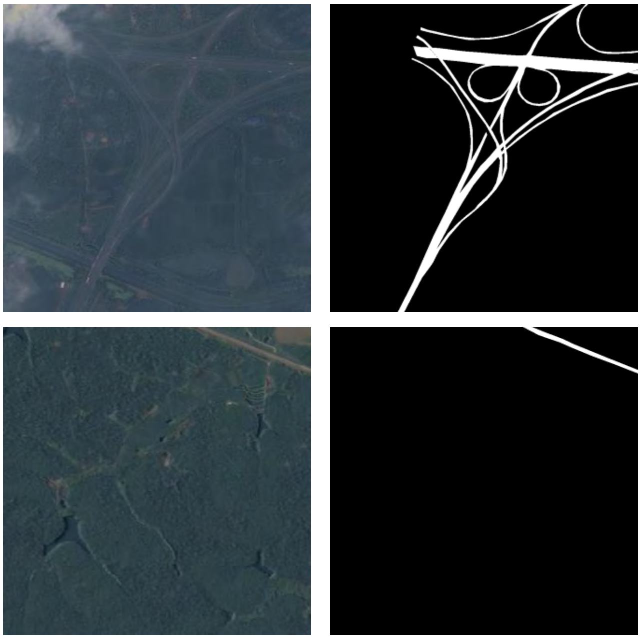


FIGURE 4. Sample aerial images from the Massachusetts dataset: (a) aerial images; (b) binary maps, ground-truth images denoting the locations of roads.

B. RSI DATASET

We obtain 1500 1000 × 1000-pixel satellite remote sensing images from QuickBird, of which 20% were made into test dataset. Each image has red, green, and blue channel information after geometric correction. As shown in Fig. 5, the dataset covers a wide variety of terrain, including plains, basins, and hills, leading to more challenges for road extraction in this work.



(a) (b)

FIGURE 5. Sample satellite images from the RSI dataset: (a) satellite images; (b) binary maps, ground-truth images denoting the locations of roads.

There are two major differences between the Massachusetts and RSI datasets: type and annotation methods. The Massachusetts images were obtained by aerial photography, and the others by satellite imagery. As for annotation methods, the Massachusetts dataset used equal width lines to mark the centers of roads and the ground truth of the RSI dataset is completely coincident with the road, retaining the geometric information.

C. EVALUATION METRICS

To quantitatively evaluate the performance of different frameworks on road extraction, we use four common metrics, as shown in Equations (6)–(9): precision, recall, F-measure (F1), and mIoU. All of them are based on four basic components in information retrieval: true positive (TP), true negative (TN), false positive (FP), and false negative (FN). TP indicates the number of correctly classified road pixels, TN indicates the number of correctly classified background pixels, FP indicates the number of mistakenly classified road pixels, and FN indicates the number of mistakenly classified background pixels.

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{8}$$

$$\text{mIoU} = \frac{1}{2} \times (\frac{TP}{TP + FP + FN} + \frac{TN}{TN + FP + FN}) \tag{9}$$

Precision is the ratio of correctly classified road pixels among all predicted road pixels, while recall measures the percentage of correctly classified road pixels among all actual road pixels. F1 and mIoU are common combinations based on these four components. The higher the value, the better the performance.

V. EXPERIMENTAL RESULTS

We design several groups of comparative experiments to explore the effectiveness of each component for our MFPN model. In this section, we investigate the impact of each improvement in our network, followed by a detailed analysis of the performance on two different datasets.

A. ABLATION STUDY ON RSI DATASET

We now demonstrate the validity of the key design component elaborated earlier. Several controlled experiments with our MFPN network are conducted on the RSI dataset for this ablation study. Except for the components we are validating, all experiments keep the parameters consistent. The effectiveness of the feature pyramid and the pyramid pooling module is shown in Table 2, and the curve in Fig. 6 illustrates the contribution of our weighted balance loss function.

TABLE 2. Effectiveness of various designs on the RSI test set.

Component	MFPN				
Pyramid Pooling Module?		✓		✓	
Tailored Pyramid Pooling Module?			✓		✓
Feature Pyramid?				✓	✓
mIoU (%)	**82.4**	**87.6**	**88.5**	**89.9**	**90.4**

1) EVALUATION OF FEATURE PYRAMID

One idea is that low-level feature maps contain scarce semantic information, but with obvious location information. Conversely, high-level coarse semantic features are abundant, but location information is crude. The structure of the feature pyramid is proposed to take full advantage of both. Therefore, the choice of the combined feature map has become an important factor that can affect the experimental performance. The performance of different feature map combination strategies is shown in Table 3. As we could see, using only the monolayer feature map, neither the middle layer P3

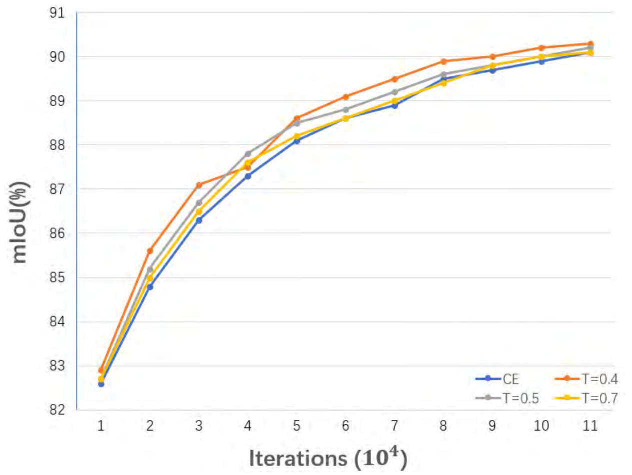


FIGURE 6. Curves of mIoU for MFPN model trained with cross entropy and weighted balance loss functions with different *T* at different iterations.

TABLE 3. Performance of different feature map combination strategies.

Combination Strategies	Precision (%)	Recall (%)	F1 (%)	mIoU (%)
P3	85.0	82.8	83.9	84.5
P5	89.6	83.9	86.7	87.6
P2+P3	85.7	83.4	84.5	85.1
P3+P4	88.0	85.1	86.5	87.3
P4+P5	90.4	86.4	88.4	88.7
P3+P4+P5	91.0	87.2	89.0	89.4
P2+P3+P4+P5	91.3	87.6	89.4	89.9

nor the last layer P5 have achieved prominent results. The fusion of P4 and P5 has better performance than the fusion of P2 and P3 or P3 and P4, which is based on the fact that the combination of P4 and P5 reached an ingenious balance between semantic features and location information. Note that the more layers we combine, the higher mIoU we get. As seen by the results presented in Table 3, when using all feature maps, the performance of mIoU is improved from 87.6% to 89.9% with other metrics reaching the optimum: 91.3% for precision, 87.6% for recall, and 89.4% for F1.

In summary, the results of the comparative experiments verify the validity of the feature pyramid. The multilevel fusion structure is better than single-level architecture in the road extraction network, which is due to the ability to fuse multilayer features for better extraction of narrow roads.

2) EVALUATION OF THE TAILORED PYRAMID POOLING MODULE

The task of road extraction can be considered as a binary classification, where nonroad pixels are negatives and road pixels are positives. The information contained in each pixel is related to the degree of difficulty in classification. Increasing the contextual information of pixels is of very useful for predicting pixels correctly. Table 4 shows the improvement of TPPM. As seen in Table 2, the model with PPM increased mIoU by 5.2% compared with the model without PPM. This is due to the fact that PPM has the advantage of increasing the contextual information of images and expanding the representation information of sparse objects. Compared with PPM, TPPM adds several averaged pooling layers that are adapted to the strip-like shape of the road. Table 4 describes the relationship between mIoU and modification strategies. The required computational resources grow with the number of pooling layers we added. Considering the computational time cost, we add two pooling layers, 5 × 80 *and*40 × 5 in subsequent experiments, offering a good trade-off between accuracy and speed. As shown in Table 6 (rows 3 and 4), mIoU increases from 87.6% to 88.5%, while F1 improves from 86.7% to 87.8%, which validates that our design is more suitable for sparse road extraction.

TABLE 4. Performance of different designs in TPPM.

Module	Extra pooling layer	mIoU (%)
PPM	None	87.6
TPPM	$\begin{cases} 5 \times 40 \\ 40 \times 5 \end{cases}$	88.3
	$\begin{cases} 5 \times 80 \\ 80 \times 5 \end{cases}$	88.4
	$\begin{cases} 5 \times 80 \\ 40 \times 5 \end{cases}$	88.5
	$\begin{cases} 5 \times 40 \\ 40 \times 5 \\ 5 \times 80 \\ 80 \times 5 \end{cases}$	88.7

Note that 5 × 40 indicates the kernel size of average pooling layer.

3) EVALUATION OF THE WEIGHTED BALANCE LOSS

The original intention of designing the weighted balance loss is to improve the performance of MFPN by focusing on the optimization of sparse objects. However, in the actual experiment, we find, amazingly, that this loss function promotes convergence of the parameters and save a lot of computing resources. In contrast, the evaluation indicator do not show surprising performance. Fig. 6 plots the curve of the relationship between mIoU and training steps. The blue curve indicates the performance of the model with standard CE loss function. The other three curves represent the performance of models with different thresholds T in the weighted balance loss function. As shown in Fig. 6, using weighted balance loss in the model can save about 104 training steps in a given mIoU. Similarly, with a specific training step, the network with the weighted loss function has higher mIoU. Comparing different thresholds, we find that the smaller T is set, the faster mIoU grows with training steps. When $T = 0.4$, we get a most beautiful curve.

TABLE 5. Performance of different methods on massachusetts dataset.

Method	Tailored Pyramid Pooling Module	Feature Pyramid	Precision (%)	Recall (%)	F1 (%)	mIoU (%)
Deeplab-v2	None	None	78.6	55..7	65.2	73.8
Deeplab-v3	None	None	78.5	56.2	65.5	74.0
PSPNet	None	None	78.1	57.1	65.9	74.3
FC-DenseNet	None	None	88.0	53.6	66.6	74.9
RSRCNN [30][a]	None	None	60.6	72.9	66.2	*[b]
MFPN	✓	✗	79.8	61.8	69.7	76.8
	✗	✓	83.9	71.7	77.3	80.6
	✓	✓	**85.1**	**74.2**	**79.3**	**82.1**

[a] Note that the experimental data of RSRCNN are from [30].

[b] The value of * represents that no corresponding evaluation indicator was used in the paper.

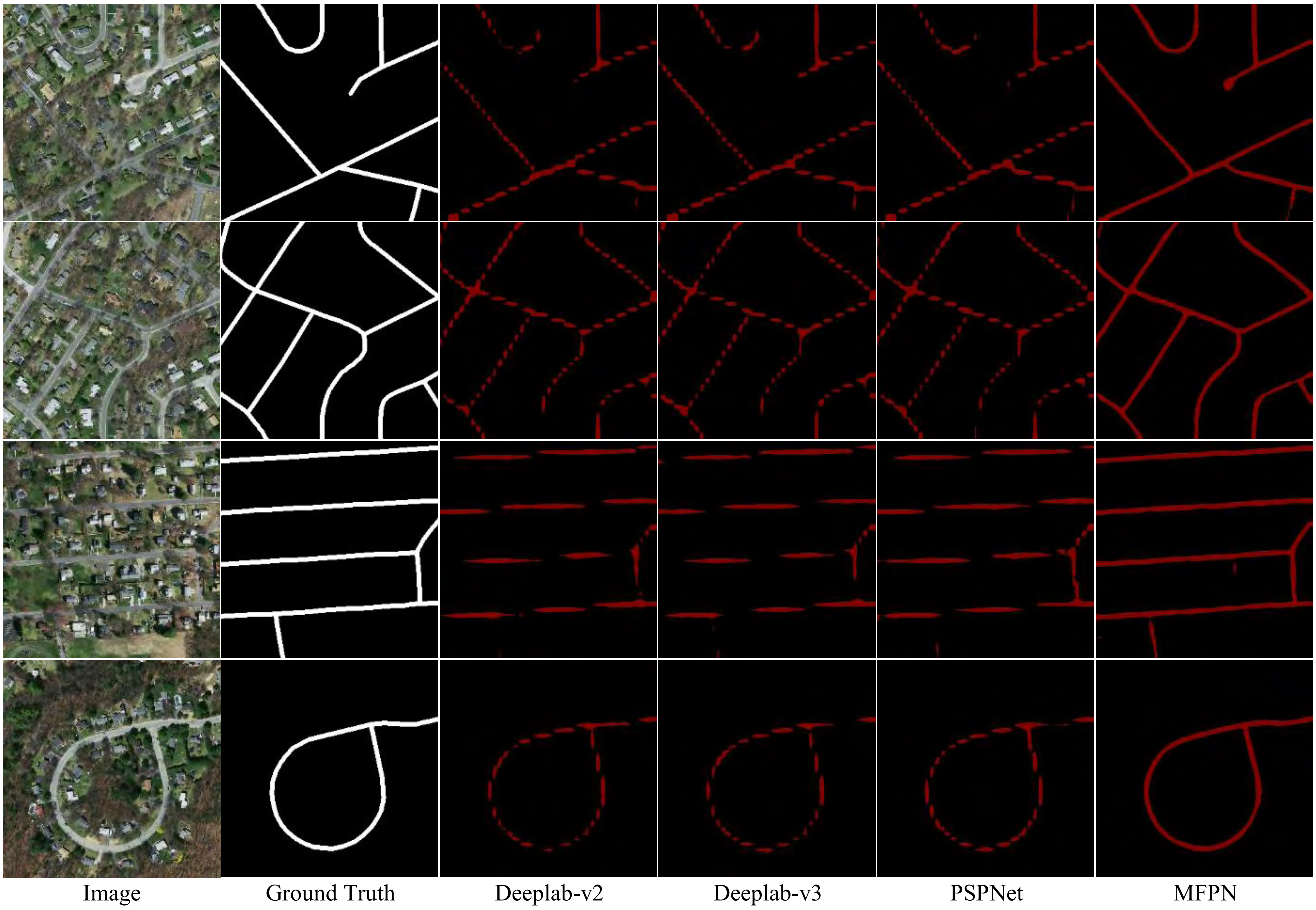


FIGURE 7. The Extraction result on Massachusetts dataset compared with other traditional methods.

What if we set a smaller T manually, such as T = 0.1? It is interesting to see the performance of MFPN with a smaller T. However, the actual experimental results show that a too-small T will cause the model to diverge. As previously mentioned, we only set the threshold T in order to solve the problem that $\mathbf{w_2^n}$ may cause instability in early training. Therefore, an excessively small T is of no practical significance and will cause the model to diverge equally. Here, we empirically set $T = 0.4$ to make road extraction results appropriate.

B. RESULTS COMPARISON ANALYSIS

In this subsection, we compare MFPN with other traditional extraction methods that are state-of-the-art in computer vision, DeepLab-v2, DeepLab-v3, FC-DenseNet [47] and PSPNet. A consistent setting was imposed on all

TABLE 6. Performance of different methods on RSI dataset.

Method	Tailored Pyramid Pooling Module	Feature Pyramid	Precision (%)	Recall (%)	F1 (%)	mIoU (%)
Deeplab-v2	None	None	89.0	81.3	85.0	86.2
Deeplab-v3	None	None	89.2	84.7	86.9	87.9
FC-DenseNet	None	None	92.0	82.4	86.9	87.8
PSPNet	None	None	89.6	83.9	86.7	87.6
MFPN	✓	✗	90.1	85.7	87.8	88.5
	✗	✓	91.3	87.6	89.4	89.9
	✓	✓	**91.8**	**88.3**	**90.0**	**90.4**

the experiments, unless the T of MFPN was 0.4. The details of results are as follows.

1) RESULTS WITH MASSACHUSETTS DATASET

All models are trained based on the union of training and valid datasets. Limited by the capacity of GPU memory, we used a batch size of 16. The training is completed when it reached 80,000 iterations.

Samples of the extraction results are shown in Fig. 7. As we can see, the images of MFPN look very close to the ground truth, which includes more detail for both dense roads in urban areas and sparse roads in the countryside. From Table 5, the results show that MFPN with all of our components was the winner compared with other classic extraction methods on any metrics (precision, recall, F1, and mIoU). As for mIoU, it outperforms DeepLab-v2, DeepLab-v3, FC-DenseNet and PSPNet by 8.3%, 8.1%, 7.2% and 7.8%, respectively; this yielded higher F1 at 14.1%, 13.8%, 12.7% and 13.4%. Meanwhile, RSRCNN performs extremely well on recall, because it uses a mechanism that fuses feature layers when extracting features. This is similar to the feature pyramid and can handle multiscale problems. Compared with the PSPNet, the experimental results of the MFPN using only TPPM are not significantly improved, while only recall increases by 4.7%. This is because our TPPM is designed based on the geometry of the road. When MFPN uses feature pyramid, the experimental results are significantly improved on precision and recall. Therefore, the value of F1 and mIOU are improved by approximately 11% and 5% respectively compared to other methods. Of course, the F1 of MFPN with all components is higher, at 13.1%, than that of RSRCNN. All comparative results confirm that MFPN is more effective and very suitable for road extraction.

2) RESULTS WITH RSI DATASET

We evaluate our MFPN on the RSI dataset in this subsection. The batch size is also set as 16 and there are 110,000 total training iterations.

Table 6 shows the quantitative comparison results of six methods measured by mIoU. The mIoU of DeepLab v3 (87.9%) is superior to that of DeepLab v2 (86.2%). The main reason is that DeepLab v3 adds global information by image pooling in the prediction phase, which benefits from the pyramid polling module of PSPNet. By fusing the contextual information of four subregions with different scales, PSPNet achieves an mIoU of 87.6%. Compared with other classical algorithms, FC-DenseNet outperforms on precision, but it has the lower performance on recall. However, all these methods still have certain limitations on the issue of multiscale. The feature pyramid combines low-level location information and high-level semantic features to solve this problem and the result in Table 6 (rows 5) show the competitive performance. Adding another improvement module, the MFPN with feature pyramid can acquire the best performance: 91.8% for precision, 88.3% for recall, 90.0% for F1, and 90.4% for mIoU. Fig. 8 shows examples of our extraction results.

3) RESULTS ON NARROW RURAL ROADS

After the above analysis, we can see that the MFPN model achieves superior accuracy. Further, we continue to explore what kind of roads MFPN is more suitable for, and whether it meets our original design intention, adapting to sparse and multiscale scenarios.

To verify our conjecture, we carefully selecte a subset of narrow roads, called the S-subset, from the RSI test dataset, where the width of the road is less than 30 pixels on the 1000 × 1000-pixel remote sensing images. We named the subset of remaining images in the RSI test dataset the N-subset. We divide the images into 320 × 320 sub-images with an overlap of 0.1. Table 7 and Fig. 9 show details of the two subsets.

TABLE 7. Numbers of two subsets of the RSI dataset.

Module	Size	Numbers
S-Subset	320 x 320	698
N-Subset	320 x 320	2080

Tables 8 and 9 represent the comparative performance of MFPN and other traditional extraction methods on the two test subsets. Compared with PSPNet (rows 3 and 4), mIoU of MFPN is increased by 10.5% on the S-subset and 2.9% on the N-subset. The increase of F1 on the S-subset is 5 times that on the N-subset. Note that recall amazingly increases by 25.6% on the S-subset, yielding the higher 5% on the N-subset,

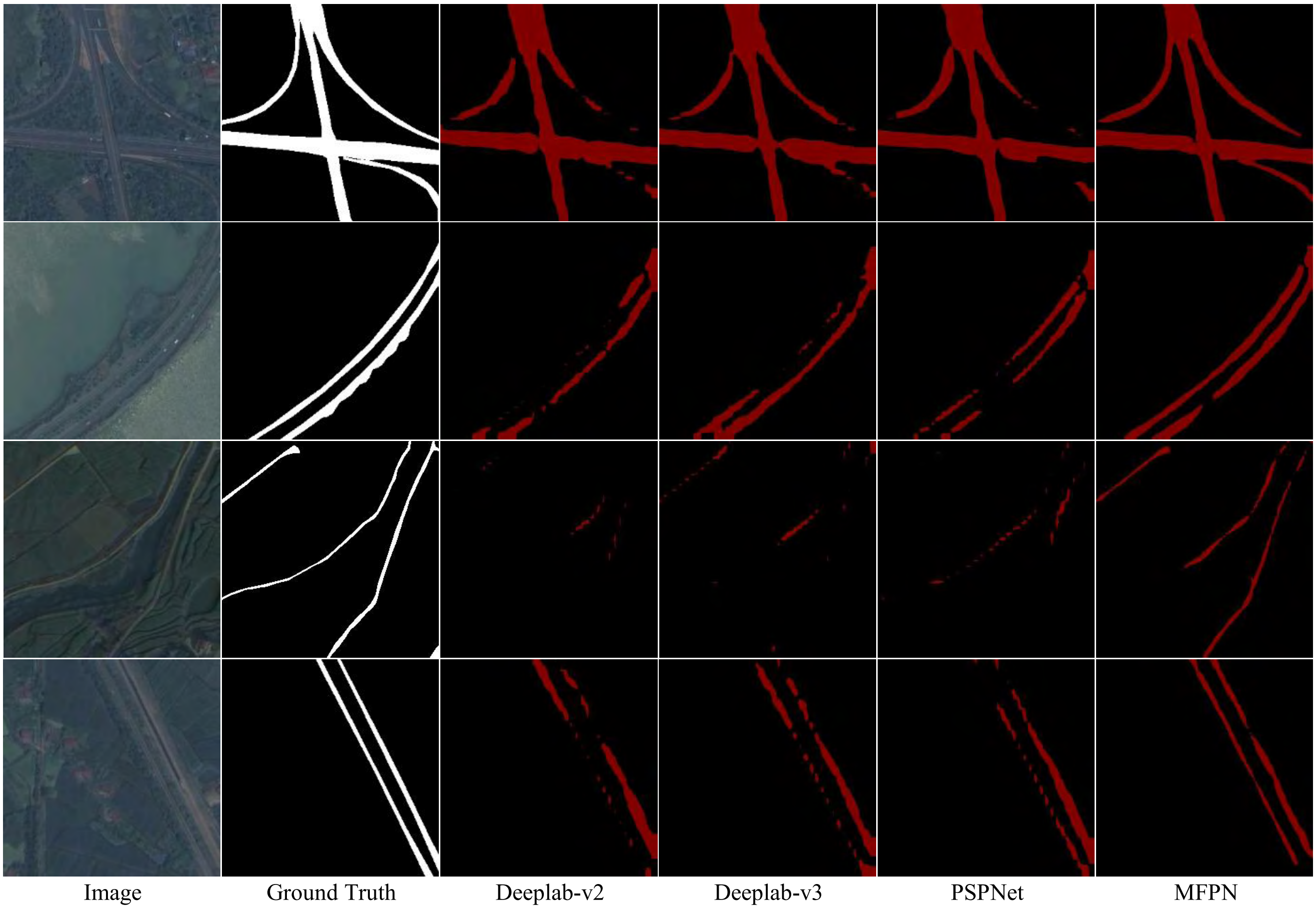


FIGURE 8. Visual improvements on the RSI dataset. MFPN produces more accurate and detailed results.

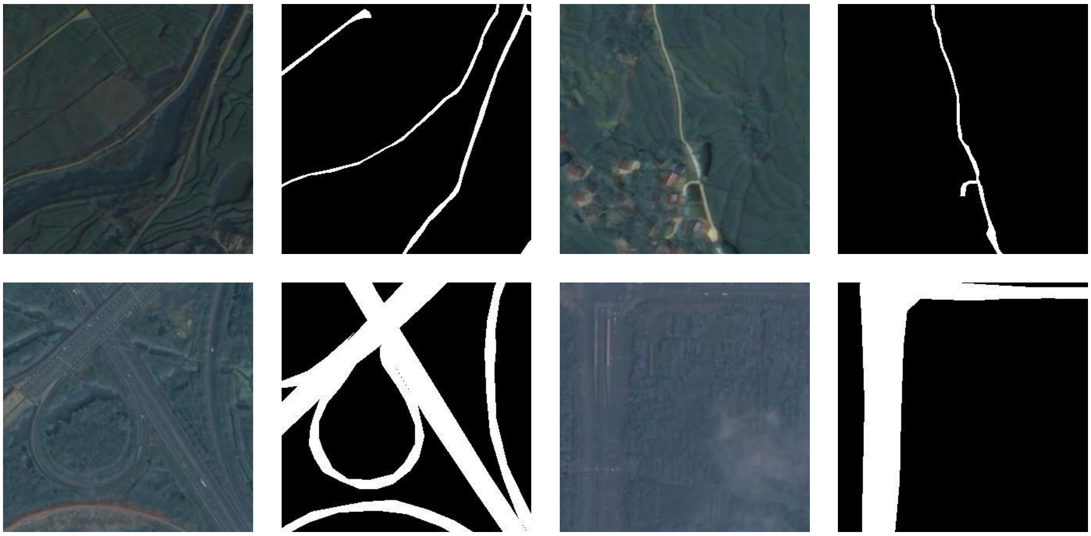


FIGURE 9. Samples of two subsets in the RSI dataset; the first row indicates the S-subset and the second row the N-subset.

which obviously indicates that MFPN can address the issue of narrow roads, because recall denotes the percentage of correctly classified road pixels among all actual road pixels.

The results on two subsets are shown in Fig. 8, where the first and second rows are from samples of the N-subset and the remainder are from the S-subset. All in all, the MFPN model

TABLE 8. Performance of different methods on the S-subset dataset.

Method	Precision (%)	Recall (%)	F1 (%)	mIoU(%)
Deeplab-v2	78.9	47.7	59.5	70.4
Deeplab-v3	79.6	49.7	61.2	71.3
PSPNet	82.3	46.3	59.3	70.3
FC-DenseNet	53.6	60.5	56.8	68.7
MFPN	**83.1**	**71.9**	**77.1**	**80.8**

TABLE 9. Performance of different methods on the N-subset dataset.

Method	Precision (%)	Recall (%)	F1 (%)	mIoU(%)
Deeplab-v2	89.5	83.7	86.5	87.3
Deeplab-v3	90.3	85.0	87.6	88.2
PSPNet	90.7	84.5	87.5	88.2
FC-DenseNet	92.5	83.9	87.8	88.7
MFPN	**92.1**	**89.5**	**90.8**	**91.1**

has superior ability to extract narrow roads, benefiting from the effective feature pyramid and tailored pooling pyramid model.

VI. CONLUSION

In this paper, we build an end-to-end MFPN model for road extraction from high-resolution remote sensing images. First, we extract image features by refined ResNets with atrous convolution, which can increase the size of receptive fields while keeping the spatial size of feature map. Then we use an effective top-down feature pyramid to fuse multilevel information, aiming to get a feature map that includes rich semantic information and sparse target location information. Subsequent tailored pyramid pooling modules further increase the combination of sparse target information and context information at different scales. Finally, training is carried out by optimizing the weighted balance loss, which accelerates the convergence of the model under the premise of ensuring mIoU. Experiments were conducted on two remote sensing datasets of different types and compared to other prevailing methods. The results show that MFPN outperforms other methods on all performance measures. Especially it gains obvious improvements for narrow rural roads.

ACKNOWLEDGMENT

The authors would like to thank all the colleagues in the lab, who generously provided their image dataset with the ground truth. The authors would also like to thank the anonymous reviewers for their very competent comments and helpful suggestions.

REFERENCES

[1] Z. Lv, Y. Jia, Q. Zhang, and Y. Chen, "An adaptive multifeature sparsity-based model for semiautomatic road extraction from high-resolution satellite images in urban areas," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 8, pp. 1238–1242, Aug. 2017.

[2] F. Tupin, H. Maitre, J.-F. Mangin, J.-M. Nicolas, and E. Pechersky. "Detection of linear features in SAR images: Application to road network extraction," *IEEE Trans. Geosci. Remote Sens.*, vol. 36, no. 2, pp. 434–453, Mar. 1998.

[3] M. Wang and C. J. Luo, "Extracting roads based on Gauss Markov random field texture model and support vector machine from high-resolution RS image," *IEEE Trans. Geosci. Remote Sens.*, vol. 9, pp. 271–276, 2005. [Online]. Available: http://refhub.elsevier.com/S2095-7564(16)30107-6/sref40

[4] N. Yager and A. Sowmya, "Support vector machines for road extraction from remotely sensed images," in *Proc. Int. Conf. Comput. Anal. Images Patterns*, N. Petkov and M. A. Westenberg, Eds. Berlin, Germany: Springer, 2003, pp. 285–292.

[5] C. Simler, "An improved road and building detector on VHR images," in *Proc. Int. Geosci. Remote Sens. Symp.*, Vancouver, BC, Canada, Jul. 2011, pp. 507–510.

[6] J. Zhou, W. F. Bischof, and T. Caelli, "Road tracking in aerial images based on human–computer interaction and Bayesian filtering," *ISPRS J. Photogramm. Remote Sens.*, vol. 61, no. 2, pp. 108–124, 2006.

[7] Z. Miao, B. Wang, W. Shi, and H. Zhang, "A semi-automatic method for road centerline extraction from VHR images," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 11, pp. 1856–1860, Nov. 2014.

[8] J. Wang and H. Q. Wang, "An interactive image segmentation method based on graph theory," *J. Electron. Inf. Technol.*, vol. 8, no. 30, pp. 1973–1976, 2008.

[9] C. Unsalan and B. Sirmacek, "Road network detection using probabilistic and graph theoretical methods," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 11, pp. 4441–4453, Nov. 2012.

[10] J. Shen, X. Lin, Y. Shi, and C. Wong, "Knowledge-based road extraction from high resolution remotely sensed imagery," in *Proc. Congr. Image Signal Process.*, Sanya, China, May 2008, pp. 608–612.

[11] R. Ma, W. Wang, and S. Liu, "Extracting roads based on Retinex and improved Canny operator with shape criteria in vague and unevenly illuminated aerial images," *J. Appl. Remote Sens.*, vol. 6, no. 1, p. 063610, 2012.

[12] P. N. Anil and S. Natarajan, "A novel approach using active contour model for semi-automatic road extraction from high resolution satellite imagery," in *Proc. 2nd Int. Conf. Mach. Learn. Comput.*, Shijiazhuang, China, Feb. 2010, pp. 263–266.

[13] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NIPS*, Lake Tahoe, NV, USA, 2012, pp. 1097–1105.

[14] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Boston, MA, USA, Jun. 2015, pp. 3431–3440.

[15] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Washington, DC, USA, Dec. 2015, pp. 1520–1528.

[16] V. Badrinarayanan, A. Kendall, and R. Cipolla. (2015). "SegNet: A deep convolutional encoder-decoder architecture for image segmentation." [Online]. Available: https://arxiv.org/abs/1511.00561

[17] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs," in *Proc. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1–14. [Online]. Available: https://arxiv.org/abs/1412.7062

[18] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018, doi: 10.1109/TPAMI.2017.2699184.

[19] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking Atrous Convolution for Semantic Image Segmentation," in *Proc. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 1–14. [Online]. Available: https://arxiv.org/abs/1706.05587

[20] S. Zheng *et al.*, "Conditional random fields as recurrent neural networks," in *Proc. IEEE Conf. Comput. Vis.*, Dec. 2015, pp. 1529–1537.

[21] G. Lin, C. Shen, A. van dan Hengel, and I. Reid, "Efficient piecewise training of deep structured models for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun./Jul. 2016, pp. 3194–3203.

[22] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1–10. [Online]. Available: https://arxiv.org/abs/1612.01105

[23] S. Saito, T. Yamashita, and Y. Aoki, "Multiple object extraction from aerial imagery with convolutional neural networks," *Electron. Imag.*, vol. 10, pp. 1–9, Feb. 2016.

[24] Z. Zhong, J. Li, W. Cui, and H. Jiang, "Fully convolutional networks for building and road extraction: Preliminary results," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Beijing, China, Jul. 2016, pp. 1591–1594.

[25] V. Mnih, "Machine learning for aerial image labeling," Ph.D. dissertation, Dept. Comput. Sci. Univ. Toronto, Toronto, ON, Canada, 2013.

[26] T. Panboonyuen, K. Jitkajornwanich, S. Lawawirojwong, P. Srestasathiern, and P. Vateekul, "Road segmentation of remotely-sensed images using deep convolutional neural networks with landscape metrics and conditional random fields," *Remote Sens.*, vol. 9, no. 7, p. 680, 2017, doi: 10.3390/rs9070680.

[27] R. Alshehhi, P. R. Marpu, W. L. Woon, and M. D. Mura, "Simultaneous extraction of roads and buildings in remote sensing imagery with convolutional neural networks," *ISPRS J. Photogramm. Remote Sens.*, vol. 130, pp. 139–149, Aug. 2017.

[28] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012.

[29] A. Tremeau and P. Colantoni, "Regions adjacency graph applied to color image segmentation," *IEEE Trans. Image Process.*, vol. 9, no. 4, pp. 735–744, Apr. 2000.

[30] Y. Wei, Z. Wang, and M. Xu, "Road structure refined CNN for road extraction in aerial image," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 5, pp. 709–713, May 2017.

[31] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Columbus, OH, USA, Jun. 2014, pp. 580–587.

[32] R. Girshick, "Fast R-CNN," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Boston, MA, USA, Jun. 2015, pp. 1440–1448.

[33] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.

[34] S. Wang, X. Gao, H. Sun, X. Zheng, and X. Sun, "An aircraft detection method based on convolutional neural networks in high-resolution SAR images," *J. Radars*, vol. 6, no. 2, pp. 195–203, 2017.

[35] K. K. Sung and T. A. Poggio, "Learning and example selection for object and pattern detection," Ph.D. dissertation, MIT, Cambridge, MA, USA, 1994.

[36] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. CVPR*, Kauai, HI, USA, Dec. 2001, pp. I-511–I-518.

[37] P. F. Felzenszwalb, R. B. Girshick, and D. McAllester, "Cascade object detection with deformable part models," in *Proc. CVPR*, San Francisco, CA, USA, Jun. 2010, pp. 2241–2248.

[38] A. Shrivastava, A. Gupta, and R. Girshick, "Training regionbased object detectors with online hard example mining," in *Proc. CVPR*, Las Vegas, NV, USA, Jun./Jul. 2016, pp. 761–769.

[39] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proc. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 2980–2988.

[40] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 770–778.

[41] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2117–2125.

[42] W. Liu, A. Rabinovich, and A. C. Berg, "Parsenet: Looking wider to see better," in *Proc. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 1–11. [Online]. Available: https://arxiv.org/abs/1506.04579

[43] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn*, Lille, France, 2015, pp. 448–456.

[44] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2016, pp. 4278–4284.

[45] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. CVPR*, Las Vegas, NV, USA, Jun./Jul. 2016, pp. 2818–2826.

[46] X. Yang *et al.*, "Automatic ship detection in remote sensing images from Google earth of complex scenes based on multiscale rotation dense feature pyramid networks," *Remote Sens.*, vol. 10, no. 1, p. 132, 2018.

[47] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio, "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit. Workshops (CVPRW)*, Honolulu, HI, USA, Jul. 2017, pp. 1175–1183.

XUN GAO (S'18) received the B.Sc. degree from Jilin University, Changchun, China, in 2016. He is currently pursuing the M.S. degree with the Institute of Electronics, Chinese Academy of Sciences, Beijing, China.

His research interests include computer vision, pattern recognition, and remote sensing image processing, especially on semantic segmentation.

XIAN SUN received the B.Sc. degree from the Beijing University of Aeronautics and Astronautics, Beijing, China, in 2004, and the M.Sc. and Ph.D. degrees from the Institute of Electronics, Chinese Academy of Sciences, Beijing, in 2009.

He is currently an Associate Professor with the Institute of Electronics, Chinese Academy of Sciences. His research interests include computer vision and remote sensing image understanding.

YI ZHANG received the B.Sc. degree from Xidian University, Xi'an, China, in 2010, and the M.Sc. and Ph.D. degrees from the University of Science and Technology of China, Hefei, China, in 2015.

He is currently a Research Assistant with the Institute of Electronics, Chinese Academy of Sciences, Beijing, China. His research interests include computer vision, array signal processing, and remote sensing image understanding.

MENGLONG YAN received the B.Sc. degree from Wuhan University, Wuhan, China, in 2007, and the M.Sc. and Ph.D. degrees from Peking University, Beijing, China, in 2012.

He is currently an Associate Professor with the Institute of Electronics, Chinese Academy of Sciences, Beijing. His research interests include LiDAR data processing and high-resolution remote sensing image processing.

GUANGLUAN XU received the B.Sc. degree from Peking University, Beijing, China, in 2000, and the M.Sc. and Ph.D. degrees from the Institute of Electronics, Chinese Academy of Sciences, Beijing, in 2005.

He is currently an Associate Professor with the Institute of Electronics, Chinese Academy of Sciences. His research interests include computer vision and geospatial information application technology.

HAO SUN received the B.Sc. degree from the Beijing Institute of Technology, Beijing, China, in 2007, and the M.Sc. and Ph.D. degrees from the Institute of Electronics, Chinese Academy of Sciences, Beijing, in 2012.

He is currently an Associate Professor with the Institute of Electronics, Chinese Academy of Sciences. His research interests include computer vision, pattern recognition, and remote sensing image processing.

JIAO JIAO (S'18) received the B.Sc. degree from Jilin University, Changchun, China, in 2016. She is currently pursuing the M.S. degree with the Institute of Electronics, Chinese Academy of Sciences, Beijing, China.

Her research interests include computer vision, pattern recognition, and SAR image processing, especially on object detection.

KUN FU received the B.Sc., M.Sc., and Ph.D. degrees from the National University of Defense Technology, Changsha, China, in 1995, 1999, and 2002, respectively.

He is currently a Professor with the Institute of Electronics, Chinese Academy of Sciences, Beijing, China. His research interests include computer vision, remote sensing image understanding, geospatial data mining, and visualization.