Road Structure Refined CNN for Road Extraction in Aerial Image

Yanan Wei, Zulin Wang, and Mai Xu

***Abstract*— In this letter, we propose a road structure refined convolutional neural network (RSRCNN) approach for road extraction in aerial images. In order to obtain structured output of road extraction, both deconvolutional and fusion layers are designed in the architecture of RSRCNN. For training RSRCNN, a new loss function is proposed to incorporate the geometric information of road structure in cross-entropy loss, thus called road-structure-based loss function. Experimental results demonstrate that the trained RSRCNN model is able to advance the state-of-the-art road extraction for aerial images, in terms of precision, recall, F-score, and accuracy.**

***Index Terms*— Convolutional neural network (CNN), machine learning, road extraction.**

I. Introduction

ROAD extraction aims at detecting and segmenting roads in aerial or satellite images by exploring some promising image processing and computer vision algorithms. It can be widely used in city planning, traffic management, GPS navigation, and so on. The tedious manual road region annotation [1] takes around 8 h per km^2. Therefore, automatic road extraction is worth studying and sometimes highly demanded. However, on account of the diversity of road appearance and occlusion, automatic road extraction from aerial or satellite images is still in its infancy, despite experiencing two decades of research.

There are many previous works attempting to extract roads in aerial images. Those works can be divided into two categories: either heuristic or data-driven. Heuristic approaches include mathematical morphology [2] and texture progressive analysis [3]. The heuristic approaches normally leverage some specific knowledge about road regions, thus being ineffective in handling diverse appearance of roads. Compared with heuristic approaches, data-driven approaches make full use of the huge data to accomplish road extraction. In the early time, some data-driven approaches have been proposed for road extraction, including clustering [4], Markov random fields (MRFs) [1], and conditional random fields (CRFs) [5]. For example, Maurya *et al.* [4] adopted K-means clustering to group the input image into various clusters followed by morphological operations, such that roads can be extracted. Máttyus *et al.* [1] employed MRF using extracted features and location information of OpenStreetMap to enhance segmented road maps. Wegner *et al.* [6] used the robust p^N-Potts model with a linear truncated cost function to obtain efficient inference in high-order CRF, for road extraction.

Recently, the state-of-the-art convolutional neural network (CNN) has been applied for remote sensing image processing, e.g., remote sensing image classification [7], dense semantic labeling of aerial images [8], and object detection in remote sensing images [9]. Most recently, fully convolutional network (FCN) [10], as one kind of CNN, show promising results in dense semantic labeling of aerial images [11]. In the field of road extraction from aerial images, CNN and FCN have also been incorporated [12], [13] to automatically learn features of roads, and then to make decision on road regions. In [12], the vector output of CNN ignores the 2-D correlation of road structure, thus leading to inferior performance of road extraction. Although structured output is applied in [13] for exploring 2-D spatial correlation of extracted roads, [13] does not consider the geometric information of road structure, when designing architecture and loss function for CNN. This may result in inferior road extraction performance. Different from the conventional object segmentation, roads normally satisfy geometric constraint, which can be seen as an obvious cue in learning to extract roads in aerial images. To the best of our knowledge, none of the existing data-driven approaches, including CNN approaches, directly imposes road structure information in the loss function, when training the road extractor.

In this letter, we propose a road structure refined CNN (RSRCNN) approach[1] for automatic road extraction, which considers both spatial correlation and geometric information of road structure in the CNN framework. Specifically, our RSRCNN approach incorporates the deconvolutional and fusion layers to provide a structured output. As such, the 2-D correlation of road structure can be taken into consideration for exploring the local structure of roads. More importantly, a road-structure-based loss function is proposed in our approach. The proposed loss function employs each pixel's minimum Euclidean distance to the road region for yielding a weight map, in which not only the importance of each pixel, but also the road geometric structure is modeled for the global structure of roads. To the best of our knowledge, our approach is the first one to apply road-structure-based loss function, for the CNN solution to road extraction.

II. RSRCNN Architecture

In this section, we present the architecture of our RSRCNN approach for road extraction, which benefits from the most recent success of fully convolutional network (FCN) [10].

Manuscript received January 23, 2017; accepted February 17, 2017. Date of publication March 13, 2017; date of current version April 20, 2017. This work was supported by the National Nature Science Foundation of China projects under Grant 61573037 and Grant 61471022. *(Corresponding author: Mai Xu.)*

The authors are with Beihang University, Beijing 100191, China (e-mail: maixu@buaa.edu.cn).

Color versions of one or more of the figures in this letter are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LGRS.2017.2672734

[1]The code of our approach is available online: https://github.com/yananweinbaa/RSRCNN.

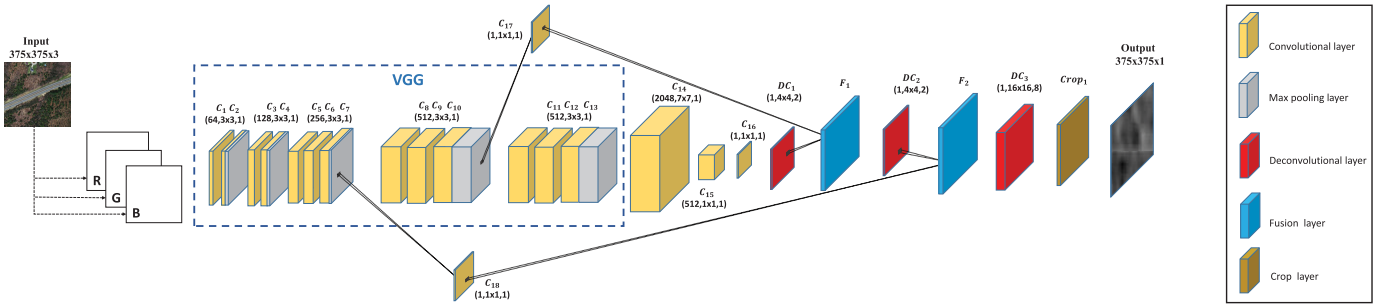

Fig. 1. RSRCNN architecture. $C_i(k, s \times s, c)$ means that the ith convolutional layer has k filters of size $s \times s$ and the stride is c pixels. The definition of $DC_i(k, s \times s, c)$ is similar to C_i. F_i denotes the ith fusion layer. Note that all convolutional and deconvolutional layers are followed by a rectified linear units. Besides, 2×2 max pooling with stride 2 is applied for the max pooling layers. Additionally, there is no padding for C_{14}–C_{18} and DC_1–DC_3.

Fig. 1 shows the proposed RSRCNN architecture. As seen from Fig. 1, the input to RSRCNN is Red-Green-Blue channels of the aerial image, whereas the output is the road map of the aerial image. Each pixel of the road map is represented by the probability of being road (=1) or background (=0). In this letter, all aerial images are cropped into 375×375 format as the input to our RSRCNN. Although our RSRCNN is fully convolutional, we crop all training images into smaller ones with the same size, for the purpose of simplification. However, it can be extended to size-variant images. Then, the first 13 convolutional layers of VGG [14] are used in our RSRCNN architecture for extracting the hierarchical features of aerial images, instead of the handcrafted features of previous road extraction approaches. VGG, as one of the typical CNN architectures, is applied here, since it has shown effectiveness in learning features for the CNN model with structured output (as verified in FCN [10]). Note that the learned parameters in VGG are used to initialize the parameters in our RSRCNN for fine-tuning. Next, three additional convolutional layers (C_{14}, C_{15}, and C_{16}) are applied for adapting to road structure. Afterward, the deconvolutional layers and fusion layers are designed and applied, since road segmentation map requires structure output. Finally, a crop layer is used to make the output of the road map the same size as input (i.e., 375×375).

Now, we describe the design of each kind of layers in our RSRCNN architecture with more details. Note that we do not discuss the convolutional layer and max pooling layer in detail in this letter, as they have been widely used in many areas. Instead, we present the new layers incorporated in this letter as follows.

1) *Deconvolutional Layer:* The deconvolutional layer takes an image or a feature map as the input, and then multiplies each pixel value in the image or feature map with parameters of the learned deconvolution kernel. Finally, the output of the deconvolutional layer can be obtained upon the multiplied pixelwise values with a fixed size stride. The spatial resolution of subsampled output of convolutional layers can be increased, through the learned upsampling operation of the deconvolutional layer. Assume that $\mathrm{DC}(k, s \times s, c)$ is the deconvolutional layer, in which k is the number of filters, $s \times s$ is the size of filters, and c is the stride of deconvolution. The mathematical operation of deconvolutional layer $\mathrm{DC}(k, s \times s, c)$ can be written by

$$O^j_{u,v} = (\mathbf{X} \overline{*} \mathbf{M}^j)_{u,v} = \sum_{w=1}^{W} \sum_{h=1}^{H} x_{w,h} m^j_{u-(w-1)c, v-(h-1)c}$$
$$j = 1, \ldots k, \ \mathbf{X} = \begin{bmatrix} x_{1,1} & \cdots & x_{1,H} \\ \vdots & \ddots & \vdots \\ x_{W,1} & \cdots & x_{W,H} \end{bmatrix},$$
$$\mathbf{M}^j = \begin{bmatrix} m^j_{1,1} & \cdots & m^j_{1,s} \\ \vdots & \ddots & \vdots \\ m^j_{s,1} & \cdots & m^j_{s,s} \end{bmatrix} \tag{1}$$

where $O^j_{u,v}$ denotes the (u, v)th element of output matrix generated by the jth deconvolutional filter. In (1), $\overline{*}$ is the deconvolution operation; $\mathbf{X}$ is the input matrix; $\mathbf{M}^j$ is the parameter matrix of the jth learned deconvolution filter. In addition, W and H denote the numbers of rows and columns for the input matrix. The principle for the deconvolutional layer is shown in Fig. 2.

2) *Fusion Layer:* The fusion layer contains two step operations. At the first step, since the input (one convolutional layer and one deconvolutional layer) may be with different sizes, the crop operation is leveraged to make them the same size in the fusion layer. Here, the output of convolutional layer is cut down to be the same size as that of the deconvolutional layer, by retaining the central part. At the second step, the cropped convolutional and deconvolutional layers are combined together with pixelwise summing. As a result, the fusion layer combines the high-level semantic information and low-level detail information together to refine the semantic precision of our RSRCNN. Summing, as a simple way of fusion, may have several drawbacks, e.g., a strong edge in the early layer of CNN may result in falsely detected roads. Our RSRCNN approach is able to relieve such drawbacks, by embedding the road structure in the loss function, to be presented in Section III.
3) *Crop Layer:* The crop layer is applied to produce the final output of RSRCNN with marginal cut, in order to ensure that the output of RSRCNN is with the same size as its input.

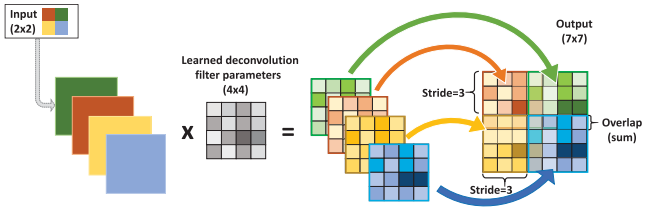

Fig. 2. Illustration for the principle of the deconvolutional layer. Assume that the input has the size of 2×2, and that the deconvolutional layer has one filter with the size of 4×4 and the stride of 3 (i.e., $k = 1$, $s = 4$, and $c = 3$). Each value of 4 pixels in the input patch is multiplied by the 4×4 matrix of the learned deconvolutional filter. Then, four 4×4 matrices are obtained, corresponding to each pixel of the input patch. Finally, the four obtained matrices are combined with stride being 3 to form the final output of the deconvolutional layer, according to the arrangement of the corresponding pixels in the input patch. Note that in the combination, the values are summarized for the overlapping region.

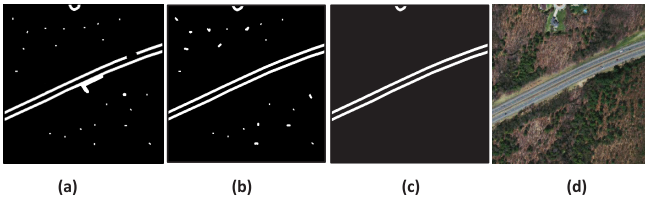

Fig. 3. (a) Extraction result I. (b) Extraction result II. (c) Ground truth. (d) Image. Illustration of an example revealing the ineffectiveness of the cross-entropy loss in modeling the loss function. The white regions in (a)–(c) denote the road regions. Assume that the falsely labeling pixels of (a) and (b) are with the same amount, given the ground truth of (c). Therefore, the two loss values of (a) and (b) are the same, when applying cross entropy to evaluate them. However, the structural integrity of roads in (a) and (b) is different. In fact, more loss should be imposed on (a), because the structural integrity of roads in (a) is missed, which can be hardly recovered. In contrast, the structural integrity of roads in (b) can be easily recovered. Thus, the existing cross entropy is ineffective in modeling the loss function for road extraction.

III. Road-Structure-Based Loss Function

In this section, the road-structure-based loss function is presented for training RSRCNN. Cross entropy is widely used as the loss function in deep learning networks to deal with binary classification problems, which calculates the probability of being one specific class or not. Thus, our loss function is also on the basis of the cross-entropy loss C defined by

$$C = \sum_{i=1}^{I} (y_i \log a_i + (1 - y_i) \log(1 - a_i)). \tag{2}$$

Assume that $\{y_i\}_{i=1}^{I}$ indicates the ground truth of the I-pixel road map, in which $y_i = 1$ means that the ith pixel belongs to road and $y_i = 0$ stands for background. In (2), $\{a_i\}_{i=1}^{I}$ is the output road map, which is modeled by the following sigmoid function:

$$a_i = \frac{1}{1 + e^{-z_i}} \tag{3}$$

where z_i denotes the input to the loss layer.[2]

By observing the definition of C in (2), we can find that the cross-entropy loss assigns equal weights to the loss of different pixels, failing to consider road structure. Fig. 3 shows that road structure is important in designing the loss function for road extraction. Accordingly, we develop the following road-structure-based loss function by encoding road structure in the cross-entropy loss of C:

$$L = \sum_{i=1}^{I} (y_i \log a_i + e^{-f(d_i)}(1 - y_i) \log(1 - a_i)). \tag{4}$$

In (4), $f(d_i)$ is a function defined as

$$f(d_i) = \begin{cases} 0, & d_i = 0 \\ \dfrac{d_i}{\max\limits_{i \in I}\{d_i\}}, & 0 < d_i \le T \\ \dfrac{T}{\max\limits_{i \in I}\{d_i\}}, & d_i > T \end{cases} \tag{5}$$

where d_i denotes the minimum Euclidean distance of the ith pixel to the road region. In addition, T is a threshold to decide whether 1 pixel is far enough from the road region. In this letter, we empirically set $T = 0.3 \max\limits_{i \in I}\{d_i\}$, to make road extraction results appropriate. Note that the loss function of (4) is only required for training our RSRCNN model, whereas we simply apply the trained RSRCNN model to predict whether a pixel belongs to road in the test images. As a result, the ground truth of road region is needed for calculating $f(d_i)$ in the training phase. In contrast, $f(d_i)$ is not calculated during the test phase, such that the ground truth of test images is not needed in the test phase.

We can see from (4) that our loss function imposes small penalty of loss on the pixel far from road regions, which has small influence on road structure. By contrast, large penalty is set to those pixels close to road regions, which have potential to influence road structure. This way, the road geometric structure is modeled in our road-structure-based loss function (4). Next, the back propagation (BP) algorithm can be directly applied to our road-structure-based loss function, and all parameters of our RSRCNN can be obtained for road extraction. Given RSRCNN with learned parameters, the probability of each pixel being road or background is calculated for generating road maps. Finally, road maps need to be binarized to classify roads and background, through the classical segmentation method [15].

IV. Experiments and Results

In this section, we present the experimental results to evaluate the road extraction performance of our RSRCNN approach. For evaluation, all 1171 aerial images[3] of [16] were used in our experiment. There exist a lot of data sets for analysis on aerial images, such as the Bavaria, Aerial KITTI, Vaihingen, and Potsdam data sets.[4] To the best of our knowledge, the data set of [16] is the largest one among all existing data sets established for road extraction, and it is thus used as the benchmark in our experiments. We followed [16]

[2]The loss layer is the last layer of CNN during the phase of training.

[3]These aerial images were stemming from publicly available Massachusetts Roads data set, which covers more than 2600 km^2 in total [13]. They were captured at the resolution of 1 m/pixel, all containing roads.

[4]http://www.dlr.de/eoc/en/desktopdefault.aspx/tabid-5431/9230_read-45479/;
http://www2.isprs.org/commissions/comm3/wg4/semantic-labeling.html

to divide those 1171 aerial images into training (1108 images), validation (14 images), and test sets (49 images). Note that the resolution of all these images is 1500×1500. Also, note that the ground truth of training aerial images is utilized to learn the RSRCNN model, and then, the roads of test aerial images are extracted using the trained RSRCNN model without any ground truth. Here, the ground truth of the test images is only used to evaluate the performance of road extraction. Next, we discuss on the setting of our experiment.

A. Setting

First, we segmented each image, in training, validation, and test sets, into 16 nonoverlapping 375×375 images, as the input to RSRCNN. Then, RSRCNN was implemented in deep learning platform "Caffe" [17]. Here, we utilized the pretrained parameters of the first 13 convolutional layers of VGG as the initial parameters, for fine-tuning on our RSRCNN. Then, the BP algorithm was applied to train our RSRCNN. For training, the learning rate was set to 10^{-12}, and it was gradually reduced by a factor of 0.1 every 20 000 iterations. Note that we apply the learning rate for unnormalized loss,[5] since the size of input image patches is the same. Besides, the batch size was set to be 8. There were in total 40 000 iterations for training our RSRCNN as to be discussed in the following. Here, all the above-mentioned hyperparameters were tuned on the validation set to minimize the road extraction error.

B. Performance Comparison

We compare our RSRCNN with three state-of-the-art approaches [5], [6], [13], among which [13] is the latest CNN approach. The metrics of precision, recall, F-score, and accuracy are measured for comparison. Table I reports the results of four approaches. It can be seen that our approach significantly outperforms all other three approaches. In particular, our RSRCNN approach reaches the highest accuracy rate (92.4%) among four approaches. Meanwhile, our RSRCNN approach significantly increases precision rate, with at least 13.5% improvement. Then, F-score is measured for quantifying the tradeoff between precision and recall. As we can see, the F-score of our approach is 66.2%, which is far better than 35.9% of [5], 55.6% of [6], and 53.2% of [13]. Besides, Fig. 4 shows the subjective results of roads extracted by four approaches, as well as ground-truth segmentation. One may see that the extracted roads by our approaches are much closer to the ground truth, validating the effectiveness of our approach in road extraction.

C. Validation on the Proposed Loss Function

Since the road-structure-based loss function is the core of our RSRCNN approach, we analyze its effectiveness from two aspects: convergence and performance. In Fig. 5, we plot the performance curves of the designed RSRCNN model trained with the proposed road-structure-based loss function and with the cross-entropy loss function, alongside the iterations in the training stage. Note that the results of Fig. 5 are averaged over all test images. As shown in Fig. 5, the CNN model trained with road-structure-based loss function can achieve best performance at 40 000 iterations, while the other one trained with cross-entropy loss function achieves best performance at 50 000 iterations. Thus, it can be concluded that the road-structure-based loss function makes the convergence speed of training road extractor faster. More importantly, we can observe from Fig. 5 that the performance of the model trained with road-structure-based loss function is much better than that trained with cross-entropy loss function. It is worth mentioning that after 60 000 iterations, the CNN models obtained by road-structure-based and by cross-entropy loss functions both incur reduction on precision, recall, and F-score rates. It is mainly because of overfitting on training data. Next, in Fig. 6, we show some road extraction results by two models trained with road-structure-based and cross-entropy loss functions. One may see that our road-structure-based loss function is able to well preserve the geometric structure of extracted roads. By contrast, the extracted roads obtained from the CNN model trained with cross-entropy loss are more likely to miss the geometric structure of roads, as highlighted in red boxes. In summary, our road-structure-based loss function is capable of achieving fast convergence and better performance for the task of road extraction.

D. Analysis on Data Set Balance

There exists the probability that the good performance of our RSRCNN approach is mainly due to balanced training samples, since the loss function of (4) imposes small

[5]Unnormalized loss means that the value of loss is not normalized by dividing the total number of pixels included in the loss function.

TABLE I

PERFORMANCE OF ROAD EXTRACTION (IN PERCENTAGE) BY OUR RSRCNN AND OTHER THREE APPROACHES

	[5]	[6]	[13]	Our
Precision (%)	40.5	47.1	43.5	**60.6**
Recall (%)	32.2	67.9	68.6	**72.9**
F-score (%)	35.9	55.6	53.2	**66.2**
Accuracy (%)	82.5	89.9	90.4	**92.4**

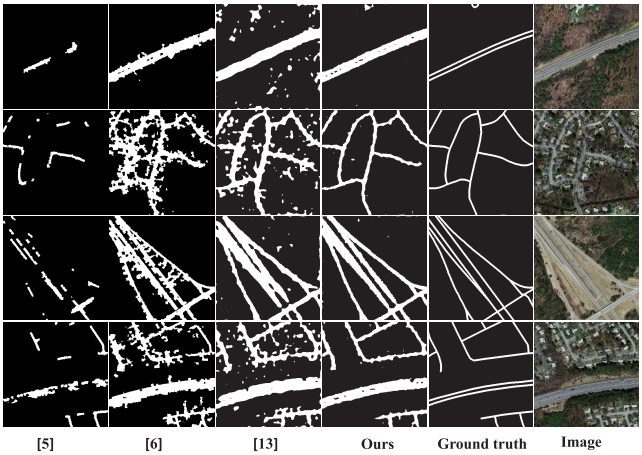

Fig. 4. Subjective results of road extraction by our RSRCNN and other three state-of-the-art approaches.

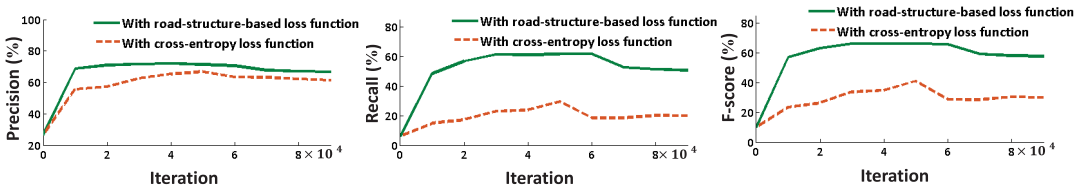

Fig. 5. Curves of precision, recall, and F-score for our RSRCNN model trained with cross entropy and our road-structure-based loss functions at different iterations.

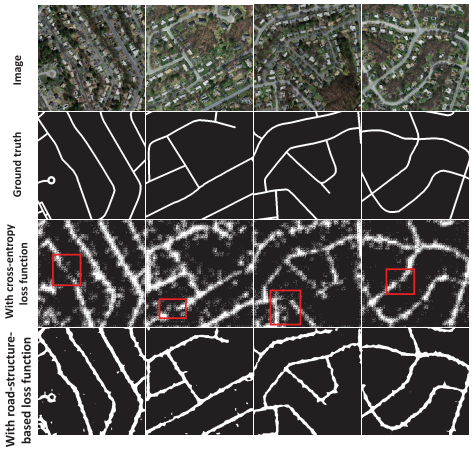

Fig. 6. Subjective results of road extraction by the designed RSRCNN model trained with cross-entropy and our road-structure-based loss functions.

TABLE II

PERFORMANCE OF ROAD EXTRACTION BY OUR RSRCNN MODEL TRAINED BY THE LOSS FUNCTION WITH CONSTANT AND ROAD-STRUCTURE-BASED WEIGHTS

	With constant weight	With road-structure-based weight
Precision (%)	12.5	60.6
Recall (%)	61.3	72.9
F-score (%)	20.8	66.2
Accuracy (%)	70.9	92.4

weight ($\leq$1) on the pixels of background. Next, we conduct an experiment to analyze whether our approach benefits from data set balance. Specifically, we apply constant penalty weight (instead of road-structure-based weight) in loss function (4), to train the proposed RSRCNN model. Here, the constant penalty weight is set to 0.12, which is proportion of pixels of road regions to all pixels, averaged in the training set. Table II reports the results of our RSRCNN model trained by the loss function with constant weight and with road-structure-based weight. It can be seen from Table II that the constant weight in (4) leads to inferior performance of road extraction. This reveals that the good performance of our RSRCNN approach takes the advantage of the proposed road-structure-based loss function, rather than data set balance.

V. CONCLUSION

This letter has proposed the RSRCNN approach in road extraction for aerial images, which incorporates road structure in learning the CNN model with structured output of road regions. First, benefitting from the recent VGG model of CNN, a new RSRCNN architecture was developed for learning to yield structured road regions of aerial images. To train our RSRCNN model, the road-structure-based loss function was then developed by embedding the geometric structure of roads. Finally, the experimental results showed that our approach outperforms other state-of-the-art road extraction approaches.

REFERENCES

[1] G. Máttyus, S. Wang, S. Fidler, and R. Urtasun, "Enhancing road maps by parsing aerial images around the world," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1689–1697.

[2] C. Zhu, W. Shi, M. Pesaresi, L. Liu, X. Chen, and B. King, "The recognition of road network from high-resolution satellite remotely sensed data using image morphological characteristics," *Int. J. Remote Sens.*, vol. 26, no. 24, pp. 5493–5508, 2005.

[3] J. B. Mena and J. A. Malpica, "An automatic method for road extraction in rural and semi-urban areas starting from high resolution satellite imagery," *Pattern Recognit. Lett.*, vol. 26, no. 9, pp. 1201–1220, 2005.

[4] R. Maurya, P. R. Gupta, and A. S. Shukla, "Road extraction using k-means clustering and morphological operations," in *Proc. IEEE Int. Conf. Image Inf. Process. (ICIIP)*, Nov. 2011, pp. 1–6.

[5] J. D. Wegner, J. A. Montoya-Zegarra, and K. Schindler, "A higher-order CRF model for road network extraction," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1698–1705.

[6] J. D. Wegner, J. A. Montoya-Zegarra, and K. Schindler, "Road networks as collections of minimum cost paths," *ISPRS J. Photogramm. Remote Sens.*, vol. 108, pp. 128–137, Oct. 2015.

[7] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Convolutional neural networks for large-scale remote-sensing image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 645–657, Feb. 2017.

[8] M. Volpi and D. Tuia, "Dense semantic labeling of subdecimeter resolution images with convolutional neural networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 2, pp. 881–893, Feb. 2017.

[9] I. Ševo and A. Avramović, "Convolutional neural network based automatic object detection on aerial images," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 5, pp. 740–744, May 2016.

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

[11] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Fully convolutional neural networks for remote sensing image classification," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2016, pp. 5071–5074.

[12] S. Saito, T. Yamashita, and Y. Aoki, "Multiple object extraction from aerial imagery with convolutional neural networks," *Electron. Imag.*, vol. 60, no. 1, pp. 1–9, 2016.

[13] 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)*, Jul. 2016, pp. 1591–1594.

[14] K. Simonyan and A. Zisserman. (Sep. 2014). "Very deep convolutional networks for large-scale image recognition." [Online]. Available: https://arxiv.org/abs/1409.1556

[15] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. 9, no. 1, pp. 62–66, Jan. 1979.

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

[17] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. 22nd ACM Int. Conf. Multimedia*, 2014, pp. 675–678.