RoadNet: Learning to Comprehensively Analyze Road Networks in Complex Urban Scenes from High-Resolution Remotely Sensed Images

Yahui Liu, Jian Yao, *Member, IEEE*, Xiaohu Lu, Menghan Xia, Xingbo Wang, and Yuan Liu

***Abstract*— It is a classical task to automatically extract road networks from very high-resolution (VHR) images in remote sensing. This paper presents a novel method for extracting road networks from VHR remotely sensed images in complex urban scenes. Inspired by image segmentation, edge detection, and object skeleton extraction, we develop a multitask convolutional neural network (CNN), called RoadNet, to simultaneously predict road surfaces, edges, and centerlines, which is the first work in such field. The RoadNet solves seven important issues in this vision problem: 1) automatically learning multiscale and multilevel features [gained by the deeply supervised nets (DSN) providing integrated direct supervision] to cope with the roads in various scenes and scales; 2) holistically training the mentioned tasks in a cascaded end-to-end CNN model; 3) correlating the predictions of road surfaces, edges, and centerlines in a network model to improve the multitask prediction; 4) designing elaborate architecture and loss function, by which the well-trained model produces approximately single-pixel width road edges/centerlines without nonmaximum suppression postprocessing; 5) cropping and bilinear blending to deal with the large VHR images with finite-computing resources; 6) introducing rough and simple user interaction to obtain desired predictions in the challenging regions; and 7) establishing a benchmark data set which consists of a series of VHR remote sensing images with pixelwise annotation. Different from the previous works, we pay more attention to the challenging situations, in which there are lots of shadows and occlusions along the road regions. Experimental results on two benchmark data sets show the superiority of our proposed approaches.**

***Index Terms*— Benchmark data set, bilinear blending, centerline extraction, convolutional neural networks (CNNs), edge detection, image segmentation, loss function, road network extraction, user interaction.**

Manuscript received November 5, 2017; revised March 20, 2018 and September 5, 2018; accepted September 6, 2018. This work was supported in part by the National Natural Science Foundation of China under Project 41571436, in part by the Hubei Province Science and Technology Support Program, China, under Project 2015BAA027, in part by the National Natural Science Foundation of China under Project 41271431, and in part by the Jiangsu Province Science and Technology Support Program, China, under Project BE2014866. *(Corresponding author: Yahui Liu.)*

Y. Liu, J. Yao, X. Lu, X. Wang, and Y. Liu are with the Computer Vision and Remote Sensing Laboratory, School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China (e-mail: liuyahui@whu.edu.cn; jian.yao@whu.edu.cn; fangzelu@gmail.com; wangxbzb@whu.edu.cn; liuyuan2011@whu.edu.cn).

M. Xia is with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong (e-mail: mhxia@cse.cuhk.edu.hk).

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

Digital Object Identifier 10.1109/TGRS.2018.2870871

I. Introduction

AUTOMATIC road extraction from remotely sensed images plays an important role in the urban design, georeferencing, vehicle navigation, geospatial data integration, and intelligent transportation system. However, it is extremely time consuming and tedious to manually label roads from the very high-resolution (VHR) images. Unsupervised learning-based methods, which often depend on several predefined features, have proved prone to failing in heterogeneous regions and achieved low accuracy. Recently, the supervised deep learning methods, such as convolutional neural networks (CNNs), have achieved the state-of-the-art performances in many high-level computer vision tasks, such as image recognition [1]–[3], object detection [4]–[6], semantic segmentation [7]–[11], edge/contour detection [12]–[14], and skeleton extraction [15]. With the development of CNNs, automatic road extraction from VHR images tends to be an economic and effective method.

In general, the road network extraction consists of three subtasks: *road surface segmentation*, *road edge detection*, and *road centerlines extraction*, as shown in Fig. 1, which involves several vision issues: semantic segmentation, edge detection, and object skeleton extraction. Therefore, it turns out to be a challenging task. *Road surface segmentation* is to extract the road pixels out [16]–[24]. We try to extract complete road surface segmentation even in some extreme situations (e.g., shadows and occlusions) that is of great difference from the previous studies. There are two main reasons that lead to the heterogeneous regions in road area: 1) buildings and avenue trees along the road can come into being shadows; 2) cars, buildings, and avenue trees can lead to occlusions. They make the road network extraction difficult and challenging. However, most of the current methods ignore or avoid the above-mentioned issues, in which the published benchmark data sets are elaborately selected in the urban areas. *Road edge detection* is to extract single-pixel width road boundaries [25], which is an important function for driver assistance systems. It is well known that fully CNNs [8], [9], [11] usually fail in the regions of heterogeneous objects, especially the boundary areas, and generate rough segmentation boundaries. We propose that the road surface segmentation results are gained from the road edges, in which some meaningful low-level features are learned to obtain refined prediction. *Road centerlines extraction* is a widely used way to represent road networks.

Fig. 1. Road network analysis in complex urban scenes from VHR images at a large scale. Our extracted road surface segmentation (blue), road edges (green), and road centerlines (red) are overlaid over the raw image. Different from previous studies, we pay more attention to the complex urban regions, where shadows and occlusions are very common. Our model predicts pretty good results in these scenes.

For most previous centerline extraction methods [26]–[31], two steps are included to obtain the final road network. First, various algorithms are applied to get the homogeneous road segmentation. Then, a centerline extraction algorithm is used to obtain the final road centerline network. On the whole, the information and memory consumption of road surface are much larger than the ones of road edge, which in turn is much bigger than the ones of road centerline.

In this paper, we propose a road network extraction system based on deep CNNs, which consists of three fully convolutional networks (FCNs) and predicts the above-mentioned three subtasks simultaneously. We explore the latest technologies to improve the performances of the proposed model. The main contributions of our approach are highlighted as follows.

1) We propose a multitask pixelwise end-to-end CNN, RoadNet, to simultaneously predict road surfaces, edges, and centerlines. RoadNet automatically learns multiscale and multilevel features and is holistically trained in a specially designed cascaded network, which can deal with the roads in various scenes and scales.
2) Above-mentioned subtasks are correlated during the training phase, in which the prediction of road surface segmentation is applied to both the road edge detection and road centerline extraction. On the one hand, the fine road surface segmentation facilitates road edge detection and road centerline extraction, which can be treated as an ideal initialization with a few complicated backgrounds. On the other hand, the accurate edges/centerlines of roads refine the segmentation boundary, especially the road edges.
3) Architecture and the loss function of the proposed network are elaborately designed. Hence, the well-trained model can produce approximately single-pixel width road edges/centerlines without nonmaximum suppression (NMS) postprocessing.
4) Simple user interaction approach is provided to solve the challenging regions with shadows and occlusions along the road, which is the first work in such field.
5) We develop a cropping and bilinear blending approach to cope with the large VHR images that are impossible to holistically train or test with finite-GPU resources.
6) A challenging benchmark data set for such multiple tasks is published, which contains images and their corresponding reference maps with 0.21-m spatial resolution per pixel covering 21 typical urban areas with complicated backgrounds.

The remainder of this paper is organized as follows. Section II reviews some related works of road network detection. The details of our proposed RoadNet is presented in Section III. Section IV introduces the proposed benchmark data set. Experiments, including evaluation metrics and performances, are provided in Section V. Conclusion and discussion are drawn in Section VI.

II. Related Works

In this section, we briefly discuss some prior works in road detection field, especially the recent deep learning-based works. In the previous works, road network detection is just confined to one or two of the mentioned subtasks (e.g., road surface segmentation [17], [21], [23], road surface segmentation and centerline extraction [32], and road centerline extraction [28], [31], [33]) with the VHR images (spatial resolution around 1 m). Especially, most previous works pay attention to extract road surface and road centerline. We first propose to comprehensively analyze road networks with higher resolution remote sensing images (0.21 m) in this field. Since visually salient road regions correspond to a variety of visual patterns, designing a universal approach to solving these tasks is difficult, especially in the cases with complicated backgrounds. It makes sense that extracting accurate road networks, including surface, edge, and centerline, from VHR images involves the visual perception of various "levels" [34]. Therefore, the traditional road detection methods [17], [18], [35] satisfy this requirement so that they suffered a series of problems in practice. Deep CNNs are powerful visual models that yield hierarchical features, which provide an ideal method to aggregate multiple "levels." Some attempts, e.g., [23], [32], [36]–[40], have applied deep CNNs to extract road networks and show promising performances.

Mnih and Hinton [16] proposed a patch-based restricted Boltzmann machines (RBMs) for road surface segmentation, in which the features obtained via principal component analysis (PCA) was set to the input. RBM was applied to learn from the PCA vectors and a postprocessing network was used to refine the incorporating structure, such as road connectivity to be the final road network. Mnih [36] applied a small CNN architecture and a postprocessing approach—conditional random fields (CRFs)—to achieve better predictions. References [37]–[39] explored deeper patch-based CNNs to improve the accuracy but ignored the consecutiveness over patches. Hence, the predictions of adjacent

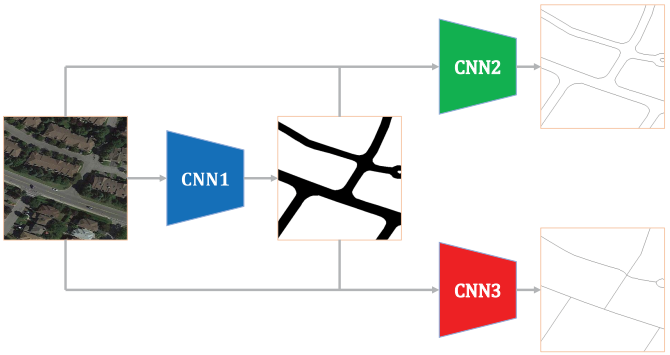


Fig. 2. Architecture of the proposed model. It consists of three CNNs: road surface segmentation network (blue, CNN1), road edge detection network (green, CNN2), and road centerline extraction network (red, CNN3).

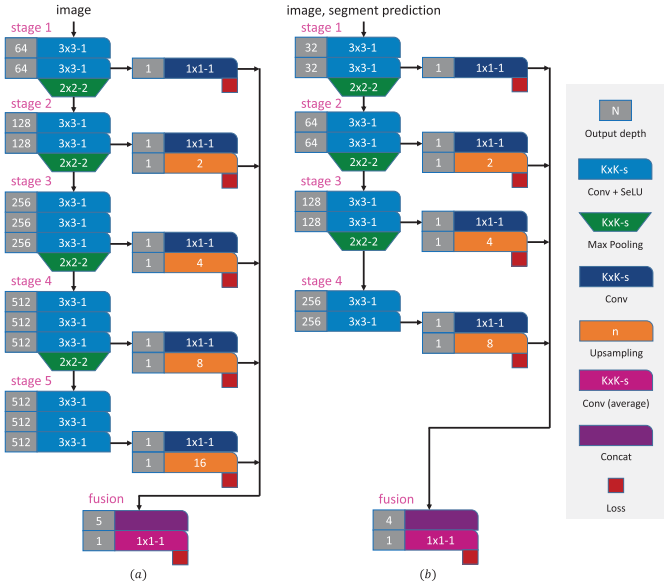


Fig. 3. Details on the proposed architectures of the three CNNs. It contains two architectures. (a) CNN1. (b) CNN2 and CNN3. "K × K" refers to receptive field size and "s" is denoted as stride. Both of them aggregate hierarchical features acquired from multiple convolutional layers.

patches usually appear poor continuity. Maggiori *et al.* [41] applied the architecture of FCNs [8], which uses upsampling/deconvolution operators to produce dense pixel-based classification. Panboonyuen *et al.* [23] developed the SegNet [11] architecture and applied the landscape metrics and CRFs methods to refine the predictions. Cheng *et al.* [32] first proposed a framework based on SegNet [32]—used the encoder–encoder architecture—to simultaneously predict road surface and centerline with a cascaded architecture, which is the most related work to ours. In Cheng *et al.*'s [31] method, an efficient NMS-based method is proposed to obtain smooth, complete, and single-pixel width road centerlines, which is not the aim of this paper but to explore one fast and efficient implementation for the multiple tasks.

Recent several fundamental works in computer vision, e.g., semantic segmentation [8], [9], [11], edge/contour detection [12]–[14], and skeleton extraction [15], have achieved promising performances. Especially, holistically-nested edge detection (HED) [12], which achieved the state-of-the-art performances on edge/contour detection, applied FCN [8] and deeply supervised net (DSN) [42] to learn meaningful features from multiple level layers in a single-trimmed VGG-16 net. Its integrated learning of hierarchical features was in distinction to previous multiscale approaches, which can be applied to the road detection task. Therefore, RoadNet is inspired by the these latest methods [12], [32], [42], [43] and achieves competitive performances.

III. Proposed Method

In this section, technologies applied in the proposed method, including architectures or RoadNet, loss function, supervision method, user interaction, bilinear blending, and training configuration, are discussed in detail.

A. RoadNet Architecture

We formulate road network extraction as a series of binary image labeling problems, where "1" and "0" refer to positive (road surface, edge, or centerline) pixel and negative one, respectively. Such applications are tasks that require both high-level features and low-level cues [13]. Our architecture, as showed in Fig. 2, is a cascaded deep CNN and contains three CNNs: road surface segmentation network (blue, CNN1), road edge detection network (green, CNN2), and road centerline extraction network (red, CNN3). In general, the CNN1, CNN2, and CNN3 can be set to any semantic segmentation CNNs [7]–[9], [11]–[13]. Here, we design a fast and efficient architecture, which aggregates hierarchical features acquired from multiple convolutional layers, inspired by VGG-16 [2] and HED-net [12]. The detailed architectures of the three networks are presented in Fig. 3.

As for the CNN1, we use the 13 convolutional layers that correspond to the first 13 convolutional layers of the VGG-16 which is designed for object classification. The fully connected layers and fifth pooling layer are discarded due to the following reasons: 1) we expect the meaningful side output with different scales, and a layer after the fifth pooling yields a too small output plane (the interpolated prediction feature map is too fuzzy to generate a refined result) and 2) the fully connected layers are computationally intensive, which is memory/time consuming [12]. The reserved part is modified to be a holistically nested network, which comprises a single-stream deep network with multiple side outputs. It contains the following three modules.

1) Stage $\{1, \ldots, 5\}$, a single-stream deep network derives from VGG-16, which is applied to learn multiscale features and different levels of visual perception.
2) Side outputs, a $1 \times 1 - 1$ *conv* layer follows each last *conv* layer of every stage. Then, an *upsampling* layer is applied to up-sample the feature map. Then, a *loss/sigmoid* layer is connected to the *upsampling* layer in each stage to get the corresponding *loss/output*.
3) Fusion, all the *upsampling* layers are concatenated by a *concat* layer. Then a $1 \times 1 - 1$ *conv* layer is applied to fuse the feature maps obtained from each side output. Finally, a *loss/ sigmoid* layer is followed to get the fusion *loss/output*.

In stages 1–5, each *conv* layer is comprised of convolution and scaled exponential linear units (SeLUs) [43]. Here, the convolution is a process with a filter bank to produce a set of feature maps. The SeLU is an activation function, which is close to zero mean and unit variance that are propagated through many network layers will converge toward zero mean and unit variance. It is proved that the SeLU allows to: 1) train deep networks with many layers; 2) employ strong regularization schemes; and 3) to make learning highly robust. The spatial pooling is carried out by a max pooling layer, which follows the last *conv* layer of each stage (not all the *conv* layers are followed by plane size reduction operation). The feature map size reduction operation is achieved by a stride 2 block: a max pooling with 2×2 pixel filter. It is used to achieve translation invariance over small spatial shifts in the image, which can also increase receptive field size for the deep *conv* layers, which gains to learn more abstract features. In the side output part, high-dimensional feature maps are reduced to 1 depth by a *conv* layer with 1×1 kernel and 1 output depth. Here, the *conv* layer is just a common convolutional layer without nonlinear units following. For simplicity, we fix all the *upsampling* layers to bilinear interpolation. Although [8] points out that one can learn arbitrary interpolation functions, [12] finds that learned deconvolutions provide no noticeable improvements for such tasks. In the fusion part, the *concat* layer is a utility layer that concatenates its multiple input blobs to one single-output blob. Then, the followed *conv* layer that its weight is a constant value Details on the loss part are shown in Section III-B, and the auxiliary supervision of the side output module is presented in Section III-C.

As for the CNN2 and CNN3, we modify the CNN1 architecture in the four aspects: 1) concatenating the feature maps of the fusion layer in the CNN1 and raw image as the input; 2) reducing the output depth by one half for each *conv* layer in stages 1–4; 3) discarding the third convolutional layers of stages 3 and 4; and 4) discarding the total stage 5 and its corresponding side output module. It is obvious that CNN2 and CNN3 are smaller than CNN1, which is designed for the following two reasons.

1) The complex network CNN1 is designed to learn features on road surface segmentation. Hence, there are less complicated backgrounds than the raw image in the prediction map generated by its fusion layer. Both the road edge and road centerline are correlated with the surface segmentation; thus, a relatively small network is adequate to solve the two subtasks.
2) Compared with the road surface segmentation issue, there are fewer positive pixels to train the two networks. Although we can apply reweight approaches in the *loss* layer, overfitting still occurs with a deep network without utilizing the pretrained model.

Therefore, we choose a simplified network, which makes full use of the image and segmentation prediction. On the one hand, the fine road surface segmentation is in favor of road edge detection and road centerline extraction, which can be treated as an ideal initialization with a few complicated backgrounds. On the other hand, the accurate edges/centerlines can refine the segmentation boundary, especially the road edges.

B. Loss Formulation

All of our tasks (road surface segmentation, road edge detection, and road centerline extraction) aim to distinguish two classes (i.e., road and background, edge and nonedge, and centerline and background), which fall into the category of semantic segmentation problem. Therefore, we take the CNN1 as an example to show the loss module. In this case, the other two networks—CNN2 and CNN3—are similar to CNN1. We define the training set as $\mathcal{S} = \{(\boldsymbol{\mathcal{X}}_n, \boldsymbol{\mathcal{Y}}_n), n = 1, \ldots, N\}$, where the image sample $\boldsymbol{\mathcal{X}}_n = \{X_j^{(n)}, j = 1, \ldots, |\boldsymbol{\mathcal{X}}_n|\}$ denotes the original input image and $\boldsymbol{\mathcal{Y}}_n = \{Y_j^{(n)}, j = 1, \ldots, |\boldsymbol{\mathcal{Y}}_n|\}$, $Y_j^{(n)} \in \{0, 1\}$ refers to the corresponding ground truth map of $\boldsymbol{\mathcal{X}}_n$. For simplicity, we subsequently omit the index n, since we consider each image holistically and independently. For each pixel, the goal of the training is to learn a model that minimizes the differences between the final prediction and the ground truth. We denote all the parameters of stages 1–5 as $\boldsymbol{\mathcal{W}}$. Each side output layer can be treated as a pixelwise classifier with the corresponding weights $\mathbf{w} = \{(\boldsymbol{w}^{(1)}, \ldots, \boldsymbol{w}^{(M)})\}$, where M is the number of side output layers.

1) Balanced Cross-Entropy Loss: Given that over 90% of the ground truth pixels are negative, the native cross-entropy loss suffers training difficulties due to trapping in the local optimal solution of most negative pixels correctly predicted but not the positive ones. Hence, we need to weight the loss differently which is termed *class balancing* [11], [12]. In our image-to-image training, with the prediction map $\boldsymbol{\mathcal{P}} = \{P_j, j = 1, \ldots, |\boldsymbol{\mathcal{X}}|\}, P_j \in \{0, 1\}$ of image $\boldsymbol{\mathcal{X}}$, the balanced cross-entropy loss of the fusion results is formulated as

$$\begin{aligned} \mathcal{L}_{bf} &= \ell(\boldsymbol{\mathcal{W}}, \mathbf{w}) \\ &= -\beta \sum_{j \in \boldsymbol{\mathcal{Y}}_+} \log \Pr(P_j = 1 | \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{W}}, \mathbf{w}) \\ &\quad - (1 - \beta) \sum_{j \in \boldsymbol{\mathcal{Y}}_-} \log \Pr(P_j = 0 | \boldsymbol{\mathcal{X}}, \boldsymbol{\mathcal{W}}, \mathbf{w}) \end{aligned} \tag{1}$$

where we denote $|\boldsymbol{\mathcal{Y}}|, |\boldsymbol{\mathcal{Y}}_-|, |\boldsymbol{\mathcal{Y}}_+|$ as the total number of all negative and positive (e.g., nonroad and road) pixels in an image $\boldsymbol{\mathcal{X}}$, respectively. Index j is over the image spatial dimensions of image $\boldsymbol{\mathcal{X}}$. $\beta = |\boldsymbol{\mathcal{Y}}_-|/|\boldsymbol{\mathcal{Y}}|$ and $1 - \beta = |\boldsymbol{\mathcal{Y}}_+|/|\boldsymbol{\mathcal{Y}}|$ are the class loss weights for corresponding negative pixels and positive ones, respectively. $\Pr(\cdot) \in [0, 1]$ refers to the probability of negative or positive for a pixel in the predicted map.

2) Construction Loss: Though the $\mathcal{L}_{bf}$ loss provides pretty good fitting between the data distribution and the trained discriminative distribution, there still exits overfitting problems, especially in extremely imbalanced class distribution case, as shown in Fig. 4. It is obvious that the soft predictions of edges and centerlines are not single-pixel width with the balanced cross-entropy loss, in which an NMS [12], [31] postprocessing is applied to gain a smooth and single-pixel width result. Therefore, we propose a construction loss

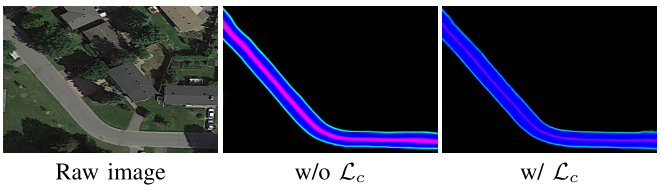


Fig. 4. Illustration of the proposed construction loss. The input raw image is predicted by two trained models: without $\mathcal{L}_c$ and with $\mathcal{L}_c$. There is a great difference between them that the prediction of the latter model generates approximately single-pixel width results.

to solve such issue during the training phase, which is formulated as

$$\mathcal{L}_c = \frac{1}{2|\boldsymbol{\mathcal{X}}|}\|\boldsymbol{\mathcal{P}} - \boldsymbol{\mathcal{Y}}\|_2^2 \tag{2}$$

where $\|\cdot\|_2^2$ refers to L2 norm, which is also known as least squares. $\mathcal{L}_c$ is basically minimizing the sum of square of the differences between the predicted map and ground truth.

3) Weight Decay Loss: Weight decay [44] is a common regularization method for optimization of mode parameters. It suppresses all irrelevant components of the weight vector by choosing the smallest vector that solves the learning task. If the size is properly chosen, it can suppress some of the effects of static noise on the prediction, which improves generalization a lot. It is defined as

$$\mathcal{L}_w = \frac{\lambda}{2}\|\boldsymbol{\mathcal{W}}\|_2^2 \tag{3}$$

where λ is a hyperparameter governing how strongly large weights are penalized and is set to a constant value $2e-4$.

4) Total Cost Function: We define the overall loss function as

$$\mathcal{L}_{\text{total}} = \alpha\mathcal{L}_{bf} + \gamma\,\mathcal{L}_c + \eta\mathcal{L}_w \tag{4}$$

where α, γ, and η are weights for the three different loss.

C. Supervision

To learn meaningful features for proposed tasks, we apply DSN [42] to supervise each side output layer. Studies [12], [45], [46] have proved that notion of auxiliary classifiers to improve the convergence of very deep networks. The original motivation is to push useful gradients to the lower layers to make them immediately useful. In addition, it improves the convergence during training by combating the vanishing gradient problem in very deep networks. Thus, more meaningful features ranging from low levels to high levels are learned by such supervision. Therefore, we apply the similar method proposed by HED [12] in our networks, in which the loss of the side outputs, $\mathcal{L}_{bs}$, is formulated as

$$\mathcal{L}_{bs} = \sum_{m=1}^{M} \omega_m \ell_{\text{side}}(\boldsymbol{\mathcal{W}}, \boldsymbol{w}^{(m)}), \tag{5}$$

where the $\ell_{\text{side}}(\cdot)$ is similar to the function $\ell(\cdot)$ of (1), and ω_m refers to the loss weight for the mth side output layer. In this case, the *loss* layer is connected to the *deconv* layer. Hence, our final overall loss function is

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_{\text{total}} + \delta\mathcal{L}_{bs} \\ &= \alpha\mathcal{L}_{bf} + \delta\mathcal{L}_{bs} + \gamma\,\mathcal{L}_c + \eta\mathcal{L}_w.\end{aligned} \tag{6}$$

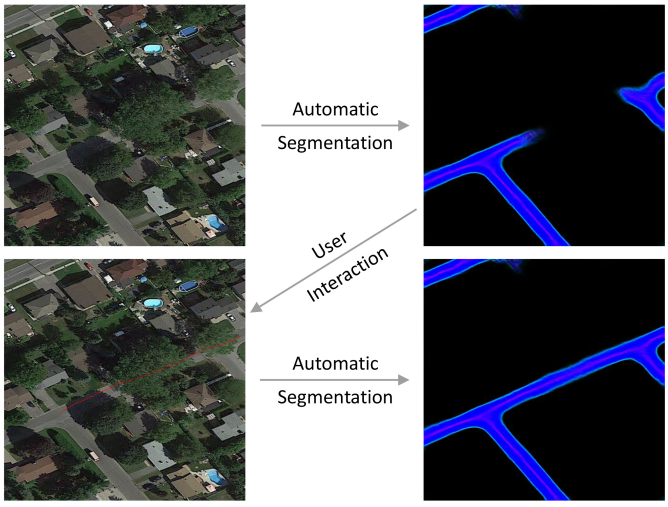


Fig. 5. We fine-tune our trained model with user editing. For the challenging heterogeneous regions, user can roughly mark with a single-pixel width brush (red). Then, the marked image is as input of the fine-tuned model, which generates better result.

We train this loss via standard back-propagation stochastic gradient descent. The details on the four hyperparameters, $\{\alpha, \delta, \gamma, \eta\}$, are discussed in Section V.

D. User Interaction

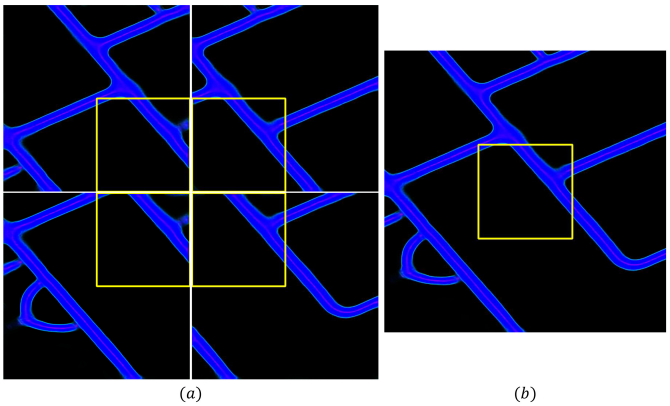
In general, priori knowledge (e.g., continuities, connectivities, and geometric features) is hard to learn by deep CNNs, which may lead to incomplete predictions in the complex and challenging road regions, as shown in Fig. 5. Some postprocessing could be proposed to solve these issues, but it may increase memory/time cost. Therefore, we introduce a simple and efficient user interaction approach which is sufficient to tackle the problems.

In our implementation, the user can roughly mark these challenging heterogeneous road regions (e.g., induced by shadows and occlusions) with a single-pixel width brush. The marked curve lines can be any shape, which are not required to be straight lines but should be located on the road regions. We use the marked images in place of the raw images in the training set and retrain the model. With such auxiliary information, our fine-tuned model can obtain a more desired result.

E. Bilinear Blending

In general, the size of a VHR image is far larger than that of a common image, which makes holistic image training and test impossible, considering the finite-memory and computational capabilities of the GPU hardware. We use a simple cropping method, in which a rectangle with fixed size slides on the VHR image with a constant stride (keeping some overlap, 75% for the training set and 50% for the testing set). Hence, the inputs of the network are cropped image patches in both the training and test phases.

During the test phase, such an approach may result in a fact that the predictions of adjacent patches appear inconsistent results, as shown in Fig. 6(a). Given that most such consecutiveness occurs in the particular places which are far from the image patch center, we assume that the prediction of a



(a) (b)

Fig. 6. (a) Example of the inconsistent predictions in different image patches. (b) Our bilinear blending result.

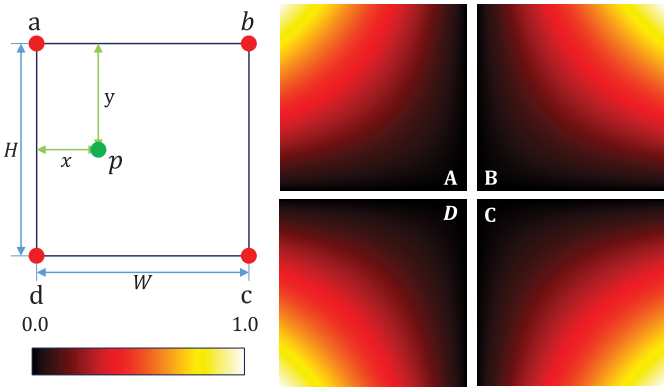


Fig. 7. Illustration of the proposed bilinear blending method. Red points $\{a, b, c, d\}$ are centers of four image patches, and the rectangle region is overlapping area of these patches. The right four heat maps $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$ are blending weight maps for each corresponding patch.

pixel is more reliable when the pixel is located more closely to the patch center. Therefore, a bilinear blending method is proposed to solve this problem.

As shown in Fig. 7, the overlapped regions—coming from adjacent patches—with centers $\{a, b, c, d\}$ are assigned weight masks $\{\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}\}$, respectively. If a point p locates in a $H \times W$ mask region with $Dist(p, \overline{ad}) = x$ and $Dist(p, \overline{ab}) = y$, we define the weights of the predictions provided by the four patches at this point as

$$\mathbf{p}_{bb} = \frac{1}{H \times W}[(W - x)(H - y), x(H - y), xy, (W - x)y]^T \tag{7}$$

where $Dist(\cdot)$ refers to the distance between a point and a line. An illustration of the bilinear blending result is presented in Fig. 6(b).

F. Training

We have trained both our network and compared methods with stochastic gradient optimization utilizing a neural network training interface tool, *tensorpack*,[1] which is based on the TensorFlow [47] distributed machine learning system using a NVIDIA TITAN X GPU with batch size 1 for 200 epochs. Our experiments used Adam [48] optimization with decay of 0.9 and $\epsilon = 1e - 3$. We applied a learning rate of $1e - 3$, dropped after every 40 epochs (40: $5e - 4$, 80: $1e - 4$, 120: $5e - 5$, and 160: $1e - 5$). Our proposed network is efficiently trained without utilizing any pretrained models for the following main two reasons.

1) Every task aims to distinguish only two classes (i.e., road and background, edge and nonedge, and centerline and background), which is easier than general semantic segmentation issues (e.g., 21 classes for PASCAL Visual Object Classes (VOC) [49], 40 categories for New York University (NYU) Depth v2 [50]). In addition, there are great differences about the semantic categories among the PASCAL VOC, NYU Depth v2, and road network in the VHR images, which lead to a result that initializing the proposed network with the pretrained model makes few effects.
2) SeLU activation function, side output supervision and elaborate loss function improve the convergence and accuracy of the proposed network.

For a 512×512 image patch, it takes about 0.09 s (11 frames/s) to simultaneously predict pixelwise road surfaces, road edges, and road centerlines, which is efficient enough.

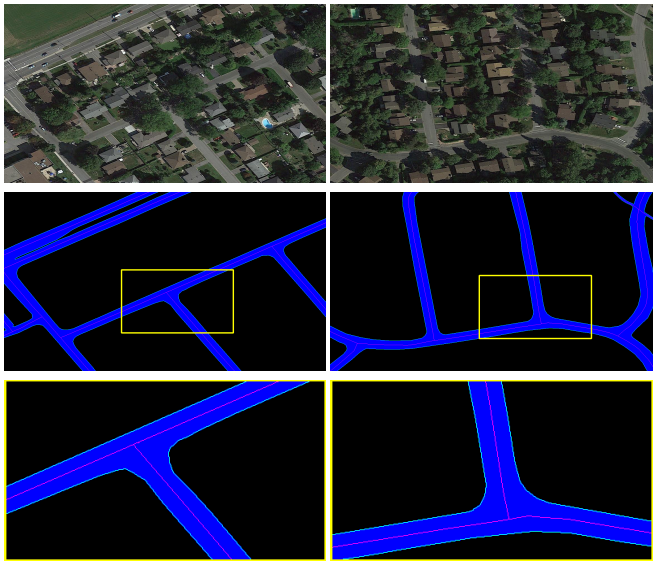


Fig. 8. Illustration of two image patches and their annotation maps. The first row shows the raw image patches, which are under complex backgrounds (e.g., shadows and occlusions). The second row presents their pixelwise annotation maps, in which the road surfaces (blue), road edges (green), and road centerlines (red) are manually labeled. The third row shows the corresponding close-ups of the yellow rectangle regions in the second row.

IV. Benchmark Data Set

This section provides details on our benchmark data set, which is used to train and evaluate the RoadNet. We collected several typical urban areas of Ottawa, Canada, from Google Earth.[2] The images with 0.21-m spatial resolution per pixel cover 21 regions about 8 km^2. We manually annotated the road surfaces, road edges, and road centerlines for each image, as shown in Fig. 8. The road width ranges from 10 to 80 pixels, and there are lots of shadows and occlusions which are caused

[1]Tensorpack: https://github.com/ppwwyyxx/tensorpack

[2]Google Earth: https://earth.google.com/web

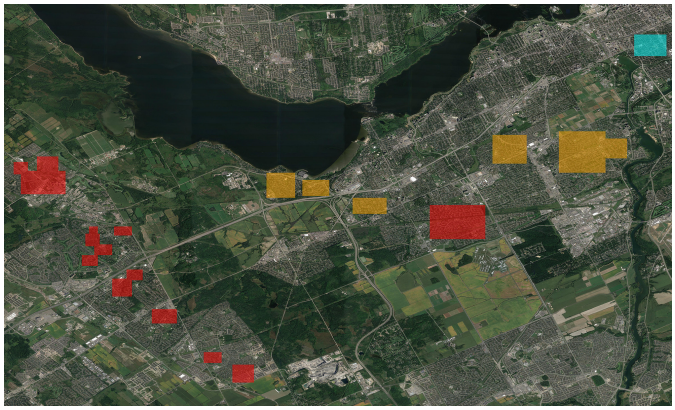


Fig. 9. Overview of the proposed benchmark data set in the urban areas of Ottawa, Canada. Red: train. Cyan: validation. Orange: test.

by cars and avenue trees along the road. Compared with the other data sets [16], [17], [32], [51], our data set is more comprehensive and challenging. Fig. 9 shows an overview of the proposed data set which is split into three subsets: a training set of 14 regions, a validation of one region, and a test set of six regions.

To build such a benchmark data set, we first manually annotated the road edges in Adobe Photoshop.[3] Then, the edge annotation could be converted to road surface segmentation map by region filling. Then, we applied a thinning method [52] to obtain the road centerline from the segmentation map. At last, the centerlines are manually refined to ensure the accuracy.

Extra manual annotation in the challenging regions with shadows and occlusions is provided for each image. Such annotation is obtained by roughly marking with a single-width brush, as shown in Fig. 5.

V. Experiments

In this section, details on the evaluation metrics and evaluation results in both qualitative and quantitative comparisons are presented.

A. Metrics

Considering the differences in road surface segmentation and road edge/centerline, we elaborately choose two specific evaluation systems.

1) Road Area Segmentation: To evaluate this paper, we introduce three metrics of common semantic segmentation evaluations [8], [11]. Let n_{ij} be the number of pixels of the class i predicted to be the class j, where there are n_{cls} different classes, and $t_i = \sum_j n_{ij}$ be the total number of pixels of the class i (both true positives and false positives are included). Then, we compute the following.

1) Global accuracy (G), which measures the percentage of the pixels correctly predicted: $\sum_i n_{ii} / \sum_i t_i$.
2) Class average accuracy (C), which means the predictive accuracy over all classes: $(1/n_{\text{cls}}) \sum_i n_{ii}/t_i$.
3) Mean intersection over union (I/U) over all classes: $(1/n_{\text{cls}}) \sum_i n_{ii}/(t_i + \sum_j n_{ji} - n_{ii})$.

[3]Adobe Photoshop: http://www.photoshop.com

TABLE I
Comparisons of the Mentioned Methods

Methods	**Layers**	**Params** (M)	**Infer Time** (ms)
FCN8s [8]	18/42	133.8/202.4	174
SegNet [11]	26/66	30.3/39.7	263
UNet [54]	21/51	40.6/45.6	169
CasNet [32]	20/48	4.0/5.8	100
RoadNet (Ours)	19/45	14.7/16.3	88

Note: "CNN1" part/overall convolutional layers ("**Layers**") and the corresponding parameters ("**Params**") shown in Fig. 14-17 are provided. Here, the **Infer Time** is overall inference time of the each model.

In addition, three common metrics in the road detection field [23], [39] are computed as follows.

1) Precision (P) $=$ (TP/TP $+$ FP).
2) Recall (R) $=$ (TP/TP $+$ FN).
3) F-score (F) $= (2PR/R + R)$.

Here, TP, FP, and FN denote the count of true positives, the count of false positives, and the count of false negatives, respectively.

2) Road Edge and Centerline: In previous studies, by comparing with the ground truth, those areas in the predicted edge/centerline map, which are within a given buffer width ρ to the ground truth, are considered as the matched areas. That is, a predicted centerline point is considered to be a true positive if it is within ρ-pixel distance from one reference centerline point. However, the buffer width ρ are usually set to different values. Hence, we propose to apply the evaluation approach [53], which is widely used in edge/contour detection problems. Edge/centerline detection accuracy is evaluated by three standard quantities: 1) the best F-measure on the data set for a fixed scale (ODS); 2) the aggregate F-measure on the data set for the best scale in each image (OIS); and 3) the average precision (AP) on the full recall range. Following the experiments in [12] and [13], the maximum tolerance allowed for correct matches of edge/centerline predictions to ground truth is set to 0.011 during evaluation.

B. Evaluation

Our RoadNet is compared with the other state-of-the-art methods, including FCN [8], SegNet [11], UNet [54], and CasNet [32]. We adjust the above-mentioned methods to cope with the multiple tasks. The adjusted models remain their original architectures as the "CNN1" module in Fig. 2 and the corresponding simplified architectures are set to "CNN2" and "CNN3." Architecture of each compared method is presented in Figs. 10–13. The performances, including the inference time (**Infer Time**) of each image patch, the best threshold (**bT**), the size of model parameters (**Params**), and the proposed metrics (**Metrics**), are evaluated. Table I shows that our proposed model is not only a lightweight model (compared with FCN8s, SegNet, and UNet) but also an efficient and fast architecture (compared with CasNet). Batch Normalization (BN) [55] operation, which is treated as a regularizer, can slightly reduce overfitting of the network and boost performances. Compared

TABLE II
ROAD SURFACE SEGMENTATION PERFORMANCES OF DIFFERENT METHODS ON OUR RNBD TESTING DATA SET

Methods	bT	Metrics					
		G	C	I/U	P	R	F
FCN8s	0.56	98.1	96.0	91.9	91.7	93.2	92.5
FCN8s$_{+}$	0.66	98.0	95.6	91.6	92.0	92.3	92.2
FCN8s$_{++}$	0.69	98.2	96.2	92.5	92.7	93.4	93.1
SegNet	0.87	96.3	92.1	85.3	84.8	86.4	85.6
SegNet$_{+}$	0.66	97.0	93.0	87.7	88.5	87.7	88.1
SegNet$_{++}$	0.37	97.7	95.1	90.5	90.4	91.8	91.1
UNet	0.50	97.3	93.5	88.7	89.9	88.5	89.2
UNet$_{+}$	0.65	97.9	95.1	91.1	91.6	91.6	91.6
UNet$_{++}$	0.72	98.3	96.2	92.7	92.8	93.6	93.2
CasNet	0.41	97.9	94.9	91.1	92.4	90.8	91.6
CasNet$_{+}$	0.54	98.2	95.8	92.1	92.6	92.7	92.7
CasNet$_{++}$	0.57	98.3	96.3	92.9	93.2	93.6	93.4
RoadNet$_{+}$	0.74	98.2	96.1	92.4	92.6	93.3	92.9
RoadNet$_{++}$	0.72	98.5	96.6	93.4	93.6	94.2	93.9

Note: "+" refers to train with our proposed loss function, and "++" is denoted as training with our proposed loss function aided by the extra manual annotation. The best performances of both "+" and "++" cases are highlighted by blue and red color, respectively.

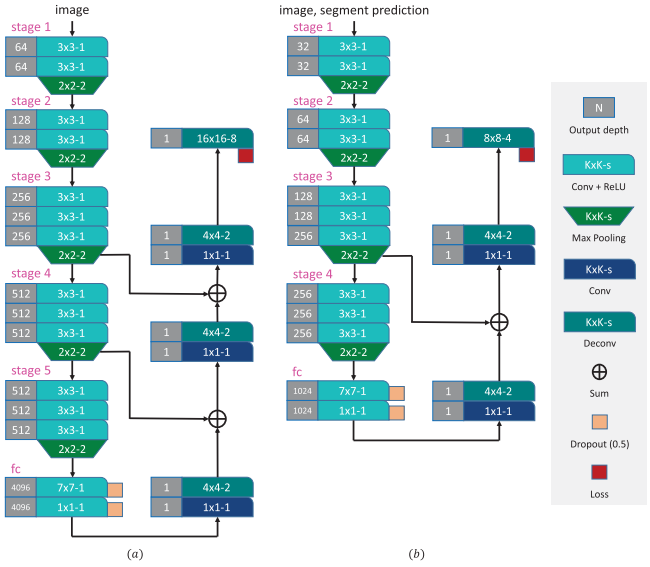


Fig. 10. Details on the architectures of FCN8s [8]. ReLU [58] and Dropout [59] (with drop rate 0.5) are applied.

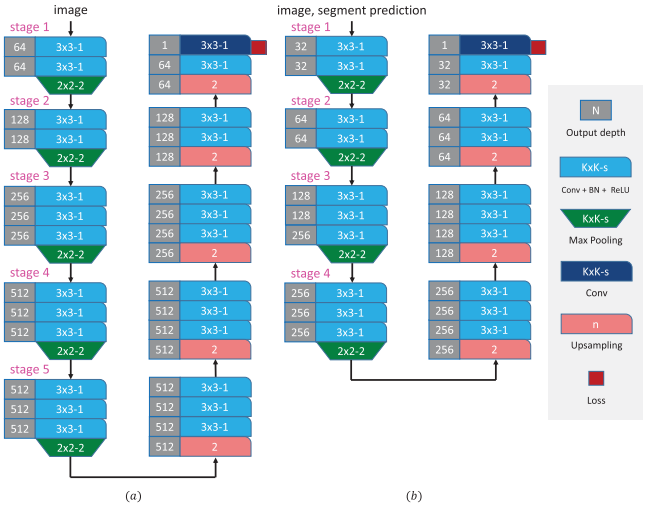


Fig. 11. Details on the architectures of SegNet [11].

FCN8s with SegNet, though the former has larger parameters than the latter, the inference time is the opposite owing to without using the BN operation during the convolutional operation. In our experiments, we find that the BN operation provides no noticeable improvements. Given such observation, we abandon applying BN but introduce SeLU [43] in order to reduce time consumption in our proposed model. In addition, parameters of each the above-mentioned methods are focused on the "CNN1" part. It is in accord with the assumption that a relatively small network is adequate to solve the two tasks (i.e., road edge detection and road centerline extraction) with the learned feature map of road surface segmentation, in which there are less complicated backgrounds than the raw image. The following qualitative and quantitative comparisons show that such an assumption works well in different methods.

We comprehensively evaluated our method on two road detection data sets: the proposed RoadNet benchmark data set (RNBD) and CasNet data set (CNDS) [32].

1) RNBD: The majority of our experiments were performed on the RNBD data set, which is described in Section IV. The performances of road surface segmentation, road edge detection, and road centerline extraction are presented as follows:

a) Architecture: As shown in Table II, RoadNet$_{+}$ and RoadNet$_{++}$ achieve the best performances of road surface segmentation on all metrics in the "+" and "++" situations, respectively. The performances verify that both the proposed architecture and loss function are effective, which provide

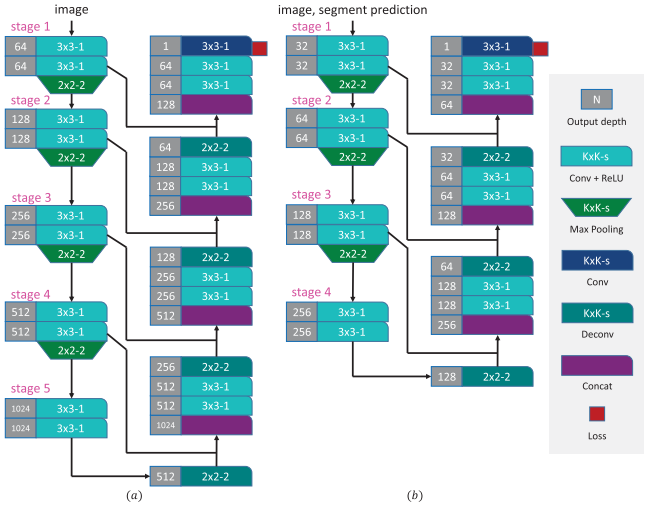


Fig. 12. Details on the architectures of UNet [54].

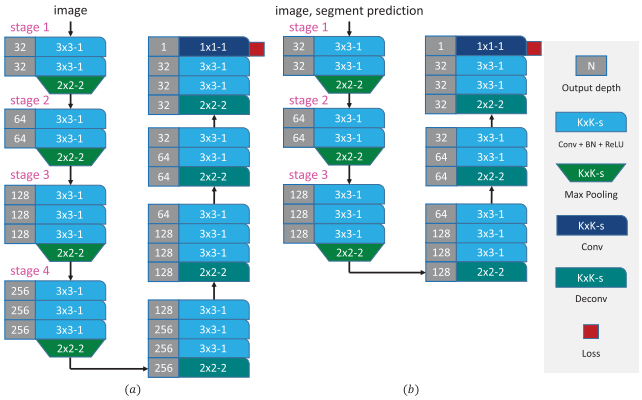


Fig. 13. Details on the architectures of CasNet [32].

TABLE III

PERFORMANCES OF ROAD EDGE DETECTION ON OUR RNBD TESTING DATA SET

Methods	**bT**	ODS	OIS	AP
FCN8s$_{++}$	0.30	92.8	93.4	91.8
FCN8s$_{++_{nms}}$	0.31	92.8	93.3	94.3
SegNet$_{++}$	0.27	92.5	92.9	90.1
SegNet$_{++_{nms}}$	0.27	92.5	92.6	92.8
UNet$_{++}$	0.28	92.0	92.5	90.3
UNet$_{++_{nms}}$	0.29	92.0	92.1	92.6
CasNet$_{++}$	0.34	92.2	92.7	52.8
CasNet$_{++_{nms}}$	0.33	92.2	92.7	52.8
RoadNet$_{++}$	0.32	93.5	94.0	93.3
RoadNet$_{++_{nms}}$	0.33	93.3	93.8	94.7

noticeable improvements on most of the compared methods. Comparisons shown in Tables III and IV are obtained from the trained models of "++" situation. Prior to evaluation, we apply a standard NMS method to the predicted road edge maps and road centerline maps to obtain thinned results, which is denoted as "++$_{\text{nms}}$." It shows that the NMS operation provides no noticeable improvements on the ODS and OIS scores but slight improvements on the AP scores. Specifically, RoadNet$_{++}$ achieves the best performances of road edge

TABLE IV

PERFORMANCES OF ROAD CENTERLINE DETECTION ON OUR RNBD TESTING DATA SET

Methods	**bT**	ODS	OIS	AP
FCN8s$_{++}$	0.22	89.6	91.1	91.4
FCN8s$_{++_{nms}}$	0.25	89.8	90.8	91.8
SegNet$_{++}$	0.22	90.8	91.1	90.7
SegNet$_{++_{nms}}$	0.21	90.8	90.4	91.6
UNet$_{++}$	0.20	89.7	91.2	89.1
UNet$_{++_{nms}}$	0.20	89.3	89.9	91.8
CasNet$_{++}$	0.25	90.3	91.0	73.2
CasNet$_{++_{nms}}$	0.25	90.4	90.9	73.3
RoadNet$_{++}$	0.23	90.5	91.8	91.0
RoadNet$_{++_{nms}}$	0.25	89.8	90.8	91.8

TABLE V

COMPARISONS OF DIFFERENT BLENDING METHODS

Methods	**Metrics**					
	G	C	I/U	P	R	F
Average	98.4	96.8	93.1	93.1	94.2	93.6
ERF	98.4	97.0	93.3	93.5	94.2	93.9
Bilinear	98.5	96.6	93.4	93.6	94.2	93.9

TABLE VI

COMPARISONS OF LOSS FUNCTION

Loss	**Metrics**					
	G	C	I/U	P	R	F
$\mathcal{L}_{bf}$	95.1	90.7	81.5	78.0	84.9	81.3
$\mathcal{L}_{c}$	96.7	92.2	86.5	87.2	86.4	86.8
$\mathcal{L}_{bf+c}$	98.5	96.6	93.4	93.6	94.2	93.9

detection on all metrics and achieves the best performances of OIS = 91.8 for road centerline extraction. Results on several image patches of the RNBD test samples are presented in Fig. 14. Two panoramas at a large scale are presented in Fig. 15 in the evaluation are obtained from the road surface segmentation results and ground truth maps. RoadNet shows competitive performances on visual effects in these experiments.

We verified the viewpoint that the road surface segmentation results are gained from the road edges/centerlines in extra experiments, in which a road surface segmentation network was trained alone with the same architecture of "CNN1" in Fig. 3 and aided by the extra manual annotation. With such training strategy, its best performances (I/U = 92.7 and F = 93.2) are lower than the ones of RoadNet$_{++}$ (I/U = 93.4 and F = 93.9). Hence, our proposed model, which correlating the multiple tasks during the training and test phases, is in favor of achieving noticeable improvements on the whole benchmark data set. It also points out a direction that simultaneously learning multitask may enhance some meaningful features to achieve improvements in both performances and computation.

b) Hyperparameters: Given two important observations that: 1) the low-level layers learn more local features with

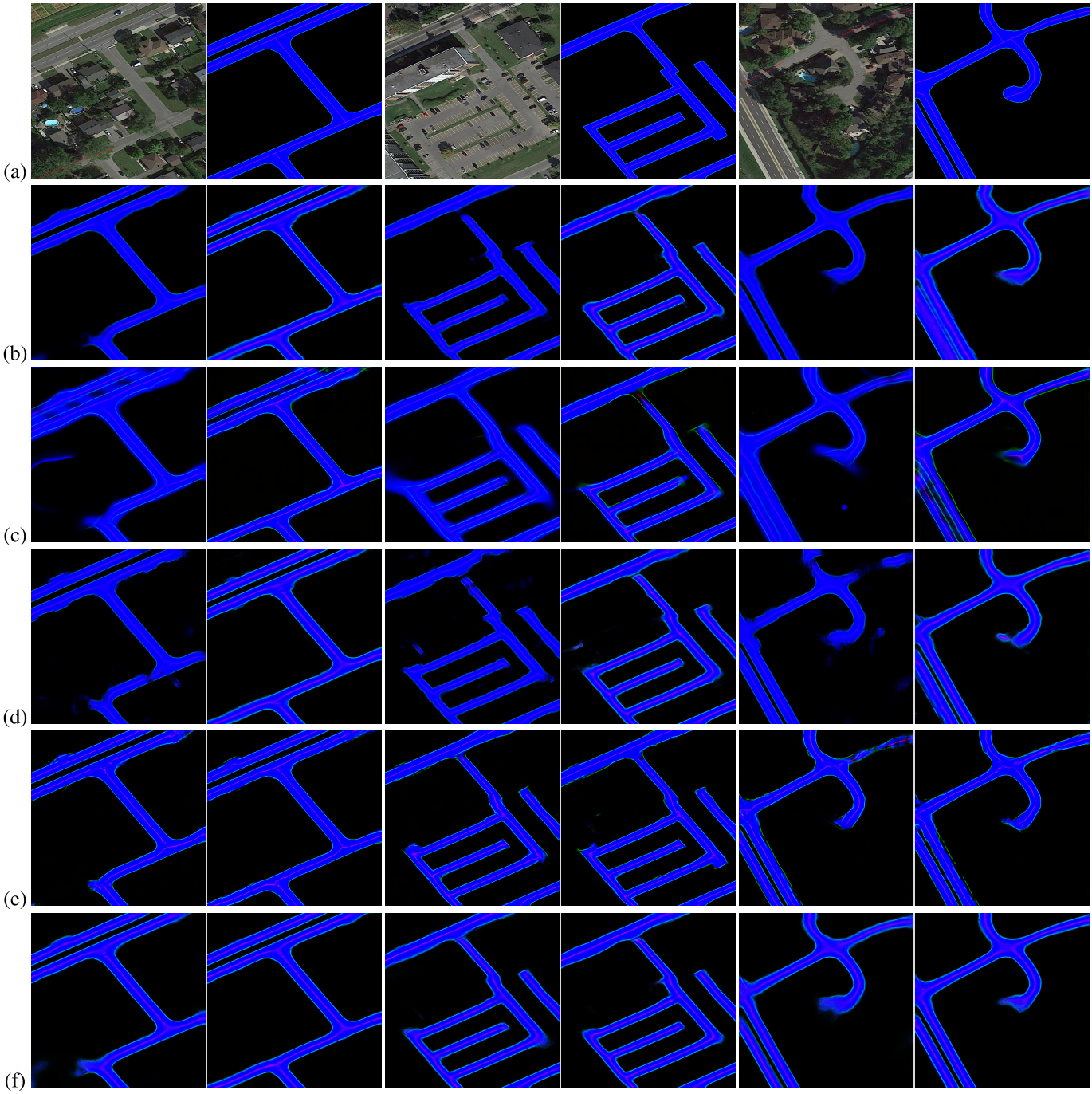


Fig. 14. Results on several image patches of the test samples. We present (a) raw image patches with extra annotation (red rough lines) and ground truth, (b) FCN8s [8], (c) SegNet [11], (d) UNet [54], (e) CasNet [32], and (f) proposed RoadNet methods. (b)–(e) Both (Left) corresponding methods trained via their original loss function with the raw images and (Right) corresponding methods trained via our proposed loss function aided with the extra annotation. (f) Results of RoadNet$_+$ and RoadNet$_{++}$.

smaller receptive fields, while more abstract features are learned from bigger receptive fields in the deep layers and 2) the low-level features provide more accurate segmentation boundary but are more susceptible to noise, such as spots, stains, and other objects similar to road regions, while higher level features show more antinoise capabilities but provide rougher segmentation boundary. Hence, the hyperparameters in (5), $\{\omega_1, \omega_2, \ldots, \omega_5\}$, are set to $\{0.5, 0.75, 1.0, 0.75, 0.5\}$, which is a trade off between segmentation accuracy and antinoise capabilities. Experiments shown in Fig. 16(a) verify these observations. The overall loss function in (6), $\mathcal{L}$, contains four hyperparameters, $\{\alpha, \delta, \gamma, \eta\}$, whose default setting is $\{1.0, 1.0, 2.0, 1.0\}$. We explore the effects of γ on the final performances, as shown in Fig. 16(b) (Here, the side output loss weights $\{\omega_i\}_{i=1}^{M}$ are set to $\{0.5, 0.75, 1.0, 0.75, 0.5\}$). It shows that there are slight improvements with $\gamma = 2.0$.

c) Bilinear Blending: We apply a quadratic weighting function to compensate the boundary effects, in which the prediction results tend to be susceptible. There are other two typical methods to deal with the inconsistent issue: 1) similar

Fig. 15. Visualization of our results on two larger image regions (4k × 4k pixels, 0.74 km^2). Green: true positive. Blue: false positive. Red: false negative.

TABLE VII
ROAD SURFACE SEGMENTATION PERFORMANCES OF DIFFERENT METHODS ON THE CNDS TESTING DATA SET

Methods	**bT**	**Metrics**					
		G	C	I/U	P	R	F
CasNet	0.02	96.8	92.2	89.2	94.5	87.6	90.9
RoadNet$_+$	0.13	97.0	94.7	90.2	90.9	91.6	91.3

Note: The original training, validation and test set are not provided in the dataset [32]. Hence, we set the last 30 images to test set to evaluate the performances and other images to train models.

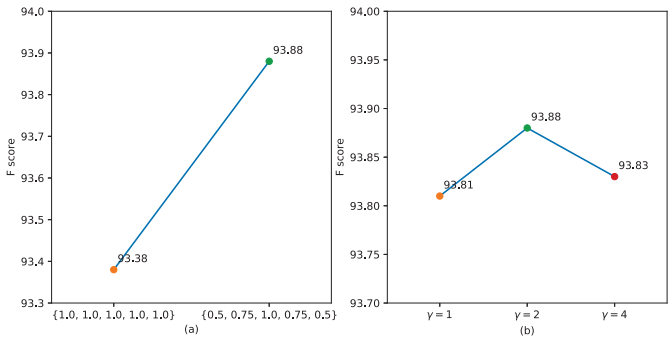


Fig. 16. Comparisons of different hyperparameters setting.

to our method, but directly averaging the overlapped regions, termed *average* and 2) discarding the boundary regions and only retaining the effective receptive field regions, termed *ERF*. We use all the three methods to process the road surface segmentation results of model RoadNet$_{++}$, and Table V shows that our proposed method achieves slight improvements compared with the above-mentioned two methods.

d) Loss function: We apply the loss term in 1 and 2 and their combination to train models, respectively. The performances on the road surface segmentation task are presented in Table VI. It is obvious that our mixed overall loss function achieves the significant improvement.

TABLE VIII
PERFORMANCES OF ROAD CENTERLINE DETECTION ON OUR CNDS TESTING DATA SET

Methods	**bT**	ODS	OIS	AP
CasNet	0.05	96.6	96.9	94.7
RoadNet$_+$	0.31	97.2	97.7	98.4

e) Edge network versus edge detection: Considering that the road edge detection can be conducted over the road surface segmentation mask, we use the Canny [56] algorithm to detect edges from the best road surface segmentation of model RoadNet$_{++}$. It achieves the performances (ODS = 86.0, OIS = 78.4, and AP = 86.0) that are lower than performances of our proposed edge network (ODS = 93.5, OIS = 94.0, and AP = 93.3).

f) Centerline network versus thinning operator: Given that the road centerline extraction can be obtained by thinning the road surface segmentation mask. We use the Guo–Hall thinning [52] operator to get road centerlines from the best road surface segmentation model of RoadNet$_{++}$. The performances of our proposed centerline network (ODS = 90.5, OIS = 91.8, and AP = 91.0) outperform the performances of the thinning operator (ODS = 89.9, OIS = 88.3, and AP = 90.0).

2) CNDS: CNDS [32] is a new benchmark with high-quality annotations for road detection and centerline

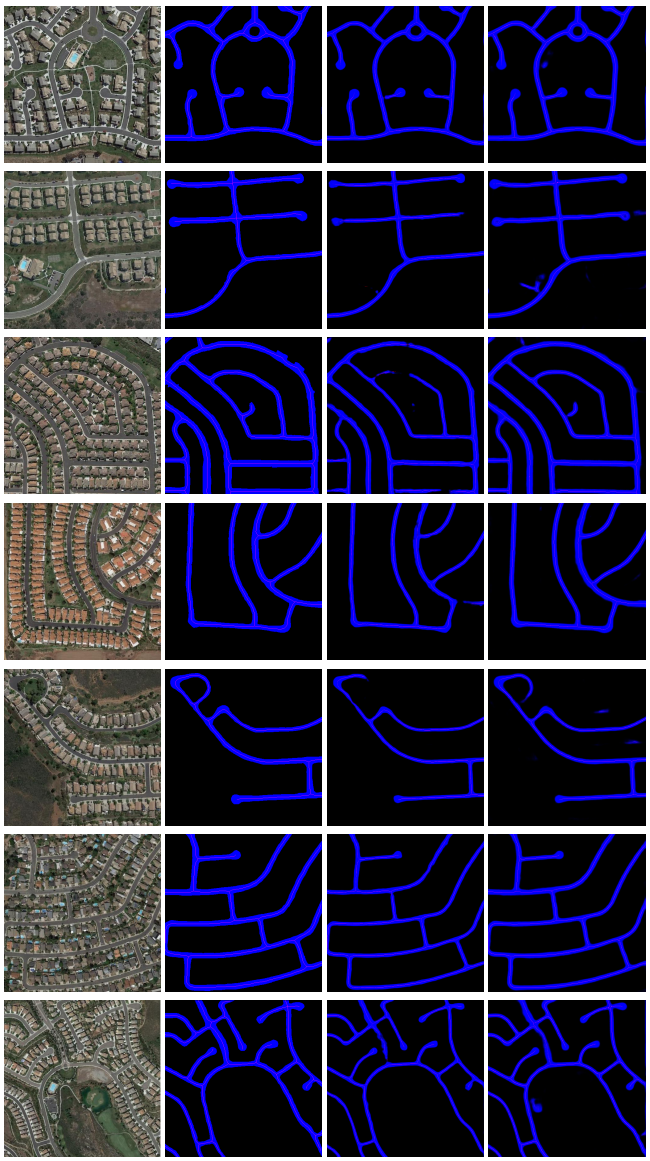


Fig. 17. Results on several image patches of the CNDS test samples. We present (a) raw image patches and (b) ground truth, (c) CasNet [32], and (d) our proposed method. In the maps from columns (b) to (d), road surface segmentation (blue) and road centerline (red) are presented.

extraction. The CNDS release includes 224 aerial images with a spatial resolution of 1.2-m per pixel. The data set is split into a training set of 180 images, a validation set of 14 images, and a test set of 30 ones. Considering the spatial resolution of CNDS is lower than the RNBD, we adjust the architecture in Fig. 2 by abandoning the last convolutional stage. Tables VII and VIII show the performances of CasNet [32] and our proposed method, in which our method is superior to the CasNet. Results on several image patches of the CNDS test samples are presented in Fig. 17.

VI. Conclusion

In this paper, a novel multitask cascaded end-to-end CNN, RoadNet is applied to simultaneously perform the tasks: road surface segmentation, road edge detection, and road centerline extraction. RoadNet automatically learns multiscale and multilevel features and is holistically trained in a specially designed cascaded network, which is applied to cope with the roads in various scenes and scales. Especially, we have been exploring the methods to deal with shadows and occlusions issues. The above-mentioned tasks are correlated during the training phase, in which the learned feature map of road surface segmentation is applied to both the road edges and road centerlines extraction. On the one hand, the fine road surface segmentation can be in favor of road edge detection and road centerline extraction, which can be treated as an ideal initialization without complicated backgrounds. On the other hand, the accurate edges and centerlines can refine the segmentation boundary, especially the road edges. Architecture and loss function of the proposed network are elaborately designed. Hence, the well-trained model can produce approximately single-pixel width road edges/centerlines without applying any NMS postprocessing. We develop a cropping and bilinear blending approach to deal with the large VHR images, which are impossible to holistically training or test with finite-GPU resources.

To evaluate the proposed method, we build a challenging benchmark data set for such multiple tasks, which contains images and their corresponding reference maps with 0.21-m spatial resolution per pixel covering 21 typical urban areas with complex background. To the best of our knowledge, it is the first comprehensive benchmark for road detection task. Experiments show that our proposed technologies are easy to apply to the previous works, in which noticeable improvements are obtained. The proposed user interaction operation solves the shadows and occlusions along the road regions well. To the best of our knowledge, this is the first work in such a field.

In the future, we plan to exploit the road detection task from two aspects: 1) refining methods, e.g., guided filtering [57] and CRFs [9], can be applied to achieve better performances; 2) designing a strategy to extract road topology information from the predicted maps, e.g., solving the problems: predicted boundary might not agree with the predicted segmentation mask, many predicted centerlines are broken within segmentation mask; and 3) exploring both the loss function and evaluation metrics to measure the topology and geometric similarity. Especially, it is worth to deal with several key issues in real-world map automation applications, e.g., unclosed boundary and wrong topologies.

Acknowledgment

The authors would like to thank NVIDIA for the generous donation of the GPUs.

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2012, pp. 1097–1105.

[2] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[4] 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)*, Jun. 2014, pp. 580–587.

[5] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1440–1448.

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2015, pp. 91–99.

[7] 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. Int. Conf. Learn. Represent. (ICLR)*, 2015.

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

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

[10] D. Eigen and R. Fergus, "Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2650–2658.

[11] V. Badrinarayanan, A. Handa, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for robust semantic pixel-wise labelling," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, 2015.

[12] S. Xie and Z. Tu, "Holistically-nested edge detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1395–1403.

[13] J. Yang, B. Price, S. Cohen, H. Lee, and M.-H. Yang, "Object contour detection with a fully convolutional encoder-decoder network," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 193–202.

[14] Y. Liu, X. H. Ming-Ming Cheng, K. Wang, and X. Bai, "Richer convolutional features for edge detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5872–5881.

[15] W. Shen, K. Zhao, Y. Jiang, Y. Wang, Z. Zhang, and X. Bai, "Object skeleton extraction in natural images by fusing scale-associated deep side outputs," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 222–230.

[16] V. Mnih and G. E. Hinton, "Learning to detect roads in high-resolution aerial images," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2010, pp. 210–223.

[17] S. Das, T. T. Mirnalinee, and K. Varghese, "Use of salient features for the design of a multistage framework to extract roads from high-resolution multispectral satellite images," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 10, pp. 3906–3931, Oct. 2011.

[18] J. Yuan, D. Wang, B. Wu, L. Yan, and R. Li, "LEGION-based automatic road extraction from satellite imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 49, no. 11, pp. 4528–4538, Nov. 2011.

[19] J. M. Alvarez, T. Gevers, Y. Lecun, and A. M. Lopez, "Road scene segmentation from a single image," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2012, pp. 376–389.

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

[21] G. Cheng, Y. Wang, F. Zhu, and C. Pan, "Road extraction via adaptive graph cuts with multiple features," in *Proc. Int. Conf. Image Process.*, 2015, pp. 3962–3966.

[22] G. Máttyus, S. Wang, S. Fidler, and R. Urtasun, "HD maps: Fine-grained road segmentation by parsing ground and aerial images," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 3611–3619.

[23] 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, pp. 2072–4292, 2017.

[24] S. Wang *et al.*, "TorontoCity: Seeing the world with a million eyes," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3028–3036.

[25] X. Li, S. Zhang, X. Pan, P. Dale, and R. Cropp, "Straight road edge detection from high-resolution remote sensing images based on the ridgelet transform with the revised parallel-beam radon transform," *Int. J. Remote Sens.*, vol. 31, pp. 5041–5059, Sep. 2010.

[26] X. Huang and L. Zhang, "Road centreline extraction from high resolution imagery based on multiscale structural features and support vector machines," *Int. J. Remote Sens.*, vol. 30, no. 8, pp. 1977–1987, 2009.

[27] Z. Miao, W. Shi, H. Zhang, and X. Wang, "Road centerline extraction from high-resolution imagery based on shape features and multivariate adaptive regression splines," *IEEE Geosci. Remote Sens. Lett.*, vol. 10, no. 13, pp. 583–587, May 2013.

[28] X. Hu, Y. Li, J. Shan, J. Zhang, and Y. Zhang, "Road centerline extraction in complex urban scenes from LiDAR data based on multiple features," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 11, pp. 7448–7456, Nov. 2014.

[29] W. Shi, Z. Miao, and J. Debayle, "An integrated method for urban main-road centerline extraction from optical remotely sensed imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 6, pp. 3359–3372, Jun. 2014.

[30] G. Cheng, F. Zhu, S. Xiang, Y. Wang, and C. Pan, "Accurate urban road centerline extraction from vhr imagery via multiscale segmentation and tensor voting," *Neurocomputing*, vol. 205, pp. 407–420, Sep. 2016.

[31] G. Cheng, F. Zhu, S. Xiang, and C. Pan, "Road centerline extraction via semisupervised segmentation and multidirection nonmaximum suppression," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 4, pp. 545–549, Apr. 2016.

[32] G. Cheng, Y. Wang, S. Xu, H. Wang, S. Xiang, and C. Pan, "Automatic road detection and centerline extraction via cascaded end-to-end convolutional neural network," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 6, pp. 3322–3337, Jun. 2017.

[33] 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.

[34] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *J. Physiol.*, vol. 160, no. 1, pp. 106–154, 1962.

[35] J. A. Montoya-Zegarra, J. D. Wegner, V. L. Ladický, and K. Schindler, "Mind the gap: Modeling local and global context in (road) networks," in *Proc. German Conf. Pattern Recognit. (GCPR)*, 2014, pp. 212–223.

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

[37] Y. Shu, "Deep convolutional neural networks for object extraction from high spatial resolution remotely sensed imagery," Ph.D. dissertation, Dept. Geogr. Environ. Manage., Univ. Waterloo, Waterloo, ON, Canada, 2014.

[38] S. Saito and Y. Aoki, "Building and road detection from large aerial imagery," *Proc. SPIE*, vol. 9405, no. 12, pp. 1814–1821, 2015.

[39] 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.

[40] D. Marmanis, K. Schindler, J. D. Wegner, S. Galliani, M. Datcu, and U. Stilla, "Classification with an edge: Improving semantic image segmentation with boundary detection," *ISPRS J. Photogramm. Remote Sens.*, vol. 135, pp. 158–172, Jan. 2018.

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

[42] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu, "Deeply-supervised nets," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2014, pp. 562–570.

[43] G. Klambauer, T. Unterthiner, A. Mayr, and S. Hochreiter. (2017). "Self-normalizing neural networks." [Online]. Available: https://arxiv.org/abs/1706.02515

[44] A. Krogh and J. A. Hertz, "A simple weight decay can improve generalization," in *Proc. Adv. Neural Inf. Process. Syst.*, 1992, pp. 950–957.

[45] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, 2015, pp. 1–9.

[46] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 2818–2826.

[47] M. Abadi *et al.* (2016). "TensorFlow: Large-scale machine learning on heterogeneous distributed systems." [Online]. Available: https://arxiv.org/abs/1603.04467

[48] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2015.

[49] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Sep. 2009.

[50] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, "Indoor segmentation and support inference from RGBD images," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2012, pp. 746–760.

[51] C. Poullis, "Tensor-cuts: A simultaneous multi-type feature extractor and classifier and its application to road extraction from satellite images," *ISPRS J. Photogramm. Remote Sens.*, vol. 95, pp. 93–108, Sep. 2014.

[52] Z. Guo and R. W. Hall, "Parallel thinning with two-subiteration algorithms," *Commun. ACM*, vol. 32, no. 3, pp. 359–373, 1989.

[53] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. 8th IEEE Int. Conf. Comput. Vis. (ICCV)*, vol. 2, Jul. 2001, pp. 416–423.
[54] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention*, vol. 9351. 2015, pp. 234–241.
[55] S. Ioffe and C. Szegedy. (2015). "Batch normalization: Accelerating deep network training by reducing internal covariate shift." [Online]. Available: https://arxiv.org/abs/1502.03167
[56] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–698, Nov. 1986.
[57] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1397–1409, Jun. 2013.
[58] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 807–814.
[59] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, 2014.

Yahui Liu received the M.E. degree from the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, China, in 2018.

He has authored several international conference papers. His research interests include machine learning, computer vision, natural language processing, and deep learning.

Jian Yao (M'08) received the B.Sc. degree in automation from Xiamen University, Xiamen, China, in 1997, the M.Sc. degree in computer science from Wuhan University, Wuhan, China, and the Ph.D. degree in electronic engineering from The Chinese University of Hong Kong, Hong Kong, in 2006.

From 2001 to 2002, he was a Research Assistant with the City University of Hong Kong, Shenzhen Research Institute, Shenzhen, China. From 2006 to 2008, he was a Post-Doctoral Fellow with the Computer Vision Group, Idiap Research Institute, Martigny, Switzerland. From 2009 to 2011, he was a Research Grantholder with the Institute for the Protection and Security of the Citizen, European Commission Joint Research Center, Ispra, Italy. From 2011 to 2012, he was a Professor with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, Shenzhen. Since 2012, he has been a Hubei Chutian Scholar Distinguished Professor with the School of Remote Sensing and Information Engineering, Wuhan University, where he is also the Director of the Computer Vision and Remote Sensing Laboratory. He has published over 90 papers in international journals and proceedings of major conferences and invented over 20 patents. His research interests include computer vision, image processing, machine learning, LiDAR data processing, and robotics.

Xiaohu Lu received the M.E. degree from the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, China, in 2017. He has published several international conference papers and journal ones, and invented several patents. His research interests include image processing, LiDAR data processing, vanish point detection, and edge detection.

Menghan Xia received the M.E. degree from Wuhan University, China, in 2017. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Hong Kong.

He has authored several international conference papers and invented several patents. His research interests include color consistency correction, image stitching, and deep learning.

Xingbo Wang received the B.E. degree from the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, China, in 2018.

He research interests include machine learning, computer vision, and data analytics.

Yuan Liu received the M.E. degree from the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, China, in 2018.

He has authored several international conference papers and invented several patents. His research interests include image processing, scene text detection, and deep learning.