Automatic Road Detection and Centerline Extraction via Cascaded End-to-End Convolutional Neural Network

Guangliang Cheng, Ying Wang, Shibiao Xu, Hongzhen Wang, Shiming Xiang, *Member, IEEE*, and Chunhong Pan

***Abstract*— Accurate road detection and centerline extraction from very high resolution (VHR) remote sensing imagery are of central importance in a wide range of applications. Due to the complex backgrounds and occlusions of trees and cars, most road detection methods bring in the heterogeneous segments; besides for the centerline extraction task, most current approaches fail to extract a wonderful centerline network that appears smooth, complete, as well as single-pixel width. To address the above-mentioned complex issues, we propose a novel deep model, i.e., a cascaded end-to-end convolutional neural network (CasNet), to simultaneously cope with the road detection and centerline extraction tasks. Specifically, CasNet consists of two networks. One aims at the road detection task, whose strong representation ability is well able to tackle the complex backgrounds and occlusions of trees and cars. The other is cascaded to the former one, making full use of the feature maps produced formerly, to obtain the good centerline extraction. Finally, a thinning algorithm is proposed to obtain smooth, complete, and single-pixel width road centerline network. Extensive experiments demonstrate that CasNet outperforms the state-of-the-art methods greatly in learning quality and learning speed. That is, CasNet exceeds the comparing methods by a large margin in quantitative performance, and it is nearly 25 times faster than the comparing methods. Moreover, as another contribution, a large and challenging road centerline data set for the VHR remote sensing image will be publicly available for further studies.**

***Index Terms*— Cascaded convolutional neural network (CasNet), end-to-end, road centerline extraction, road detection.**

I. Introduction

AUTOMATIC road extraction from remote sensing images has been an active and open research problem in remote sensing. It is an essential preprocessing step for various applications, such as vehicle navigation [1], urban planning, image registration, geographic information system update [2], and so on. However, it is extremely time-consuming and tedious to manually label the road area. With the aid of machine learning, artificial intelligence, and image processing, automatic road extraction from remotely sensed images is an economic and effective way to acquire the road information. Although various recent studies on road extraction [3]–[6] have been proposed to address this challenging task, they are far from the optimal.

Generally speaking, the road extraction task contains two subtasks: *road detection* and *road centerline extraction*. The former is to extract all the road pixels out, while the latter only aims at labeling the centerline pixels of road, which can provide directions for the vehicle navigation and other applications. Most road detection approaches [5], [7]–[10] are based on the pixel-level labeling. Due to the noise and occlusions under cars and trees in the very high resolution (VHR) remote sensing images [11], [12], the existing methods often obtain heterogeneous results. Though some object-level-based approaches [13]–[15] can alleviate this shortcoming, they are ineffective to gain consistent results around the road boundaries.

For the road centerline extraction task, morphological thinning algorithm [13], [16], [17] is widely used in extracting single-pixel width centerline network. Although the thinning-based algorithm is fast and easy to implement, it often produces small spurs around the centerline, which will largely affect the construction of road network. Recently, the regression-based methods [10], [18] and nonmaximum suppression (NMS)-based methods [14], [15], [19] have been proposed to alleviate this shortcoming. However, the regression-based approaches are ineffective to extract the centerline around the road interactions, and the NMS-based approaches may obtain road centerline network with more than single-pixel width.

Krizhevsky *et al.* [20] won the first place in the ILSVRC 2012 competition with their eight-layer AlexNet, which boomed the deep convolutional neural network (DCNN). Then, Long *et al.* [21] first proposed a framework of fully convolutional network (FCN) to the semantic segmentation task, which is efficient and has achieved the state-of-the-art performance. Under the framework of FCN, Chen *et al.* [22] introduced kernel dilation into their networks, and then dense conditional random fields (CRFs) was applied as a postprocessing operation to further improve the segmentation performance. Different from FCN, another widely used DCNN for semantic segmentation is deconvolution network [23], [24], which consists of encoder layers and decoder layers. In the deconvolution network, unpooling layer was utilized

Manuscript received June 13, 2016; revised November 25, 2016; accepted February 7, 2017. Date of publication March 7, 2017; date of current version May 19, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 91646207, Grant 91338202, Grant 61620106003, and Grant 61305049, and in part by the Beijing Natural Science Foundation under Grant 4162064.

The authors are with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China (e-mail: guangliang.cheng@nlpr.ia.ac.cn; ywang@nlpr.ia.ac.cn; shibiao.xu@nlpr.ia.ac.cn; hongzhen.wang@nlpr.ia.ac.cn; smxiang@nlpr.ia.ac.cn; chpang@nlpr.ia.ac.cn).

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.2017.2669341

0196-2892 © 2017 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

to obtain dense pixelwise class probability and precise object segmentation boundaries.

Though DCNN has achieved important breakthroughs in semantic segmentation and object recognition from natural images, few studies [25]–[31] have attempted to remote sensing applications on DCNN, which is due to the following reasons: 1) compared with natural images, remote sensing images have more complex and cluttered backgrounds and 2) there are few big public data set that is adequate to train the DCNN well.

The most related research to our framework is [24], while they are different in two perspectives. First, to better maintain spatial information, we employ the first ten convolutional layers for the encoder network, randomly initialize the parameters, and train the network from scratch. Another difference is that two convolutional networks are cascaded into one framework, through which the road detection task and road centerline extraction task can be achieved simultaneously.

To overcome the above-mentioned shortcomings in the existing road extraction methods, we propose a cascaded end-to-end convolutional neural network (CasNet) to simultaneously extract consistent road area and smooth road centerline from VHR remote sensing images. Specifically, two convolutional networks are concatenated into one framework. The first network is utilized to extract the road area. Due to the encoder and decoder layers, we can obtain more consistent road detection result than other comparing methods under complex backgrounds and occlusions of cars and trees. Through the second network and the thinning algorithm, the smooth, complete, and single-pixel width road centerline network can be achieved.

In summary, the main contributions of the proposed approach are highlighted as follows.

1) A novel cascaded end-to-end convolutional neural network is proposed. To the best of our knowledge, it is the first attempt to use one cascaded network to bridge two subtasks together in the remote sensing applications.
2) Due to the encoder and decoder layers, as well as their strong representation ability, the proposed method can achieve more consistent road detection result than other comparing methods under complex backgrounds and occlusions of cars and trees.
3) The road centerline extraction task is formulated as two-class classification problem (i.e., centerline and noncenterline) via a convolutional network. Then, a thinning algorithm is utilized. In the end, smooth, complete, and single-pixel width road centerline network can be achieved.
4) A challenging road centerline extraction data set of large size will be publicly shared to facilitate further studies. It contains 224 VHR remote sensing images together with their corresponding segmentation reference maps and centerline reference maps. To the best of our knowledge, this is the biggest road centerline data set so far.

The remainder of this paper is arranged as follows. The related road extraction work is systematically reviewed in Section II. In Section III, we briefly review some basic modules used in CasNet. Section IV presents the details of the proposed road detection and centerline extraction method. Section V provides the detailed descriptions of our new data set. Experimental evaluations as well as detailed comparisons between our method and state-of-the-art methods are provided in Section VI. Finally, the conclusion and discussion will be outlined in Section VII.

II. Previous Works

In general, the existing road extraction methods can be roughly classified into two categories: *road detection* methods [5], [7]–[9], [32]–[34] and *road centerline extraction* methods [10], [13], [14], [18], [19], [35]–[39].

The road detection methods mainly depend on pixel-level or superpixel-level classification. In this field, Song and Civco [7] proposed a two-step-based method consisting of support vector machine (SVM) and shape index to detect road network. Zhang and Couloigner [8] proposed an integrated approach to extract road area. In their method, *k*-means, fuzzy logic classifier and shape descriptors of angular texture signature were employed. To some extent, it can distinguish the parking lots from the road area. An automatic road extraction method from remote sensing images was introduced by Yuan *et al.* [9]. In their method, locally excitatory globally inhibitory oscillator network was utilized. Das *et al.* [5] introduced a multistage framework to extract road from the high-resolution multispectral satellite image, in which probabilistic SVM and salient features were used.

Recently, Álvarez *et al.* [32] proposed a convolutional neural network-based algorithm to learn features from noisy labels for road extraction. In this method, a new algorithm of generating training labels and a novel texture descriptor were introduced. A patch-based deep neural network method was proposed by Mnih and Hinton [33] to extract the urban road network from high-resolution images, in which unsupervised pretraining and supervised postprocessing were introduced to refine the performance of the road detector substantially. Wegner *et al.* [34] proposed a higher order CRF model to detect road network. In their method, the road prior was represented by higher order cliques that connect sets of superpixels along straight line segments.

It is a tricky problem to extract the road centerline directly from the remote sensing images. To get around this tough task, most popular and successful road centerline extraction methods are based on two processing steps: road detection and centerline extraction. To extract road centerline, Zhu *et al.* [35] utilized the binary-grayscale mathematical morphology to extract the road area, and then a line segment match algorithm was introduced to extract centerline network. Gamba *et al.* [36] proposed an integrated road centerline extraction algorithm for the urban areas. In this method, predominant directions of road were captured via an adaptive filtering procedure; then, a perceptual grouping algorithm was introduced to discard redundant segments and avoid gaps. Finally, the road network topology was considered by checking for the road intersections and regularizing the overall patterns. A novel road centerline extraction method was proposed in [13] by integrating mutiscale information and SVM. In their method,

hybrid spectral-structural features were analyzed by using SVM classifier. Then, multiscale results were fused to obtain classification result. Finally, morphological thinning algorithm was employed to detect the centerline.

A novel system [37] was introduced to extract road centerline form high-resolution images. It had three modules: probabilistic road center detection, road shape extraction, and graph-theory-based road network formation. An automatic road centerline extraction method was presented by Miao *et al.* [10]. To achieve this, shape and spectral features were utilized to obtain potential road segments. Then, multivariate adaptive regression splines (MARSs) were employed to extract smooth road centerline. Shi *et al.* [18] proposed an integrated urban main-road centerline detection method. In their method, by fusing spectral-spatial classification and local Geary's C method, consistent road area extraction result was obtained. Then, local linear kernel smoothing regression algorithm was introduced to extract smooth road centerline. The regression-based centerline extraction methods can extract smooth centerline, while they are ineffective to extract centerline in the intersections. Cheng *et al.* [14] presented a three-step road centerline extraction approach. To achieve this, road detection result was obtained via multiscale joint collaborative representation and graph cut. Then, tensor voting and NMS were utilized to extract smooth centerline. Finally, a fitting-based connection algorithm was introduced to effectively connect centerline around the intersections. Another three-step road centerline extraction approach was proposed by Hu *et al.* [38]. It integrated adaptive mean shift, stick tensor voting, and weighted Hough transform into a framework.

Sironi *et al.* [19] provided a regression algorithm to directly detect road centerline from the remote sensing images. To achieve it, they trained regressors to return the distances to the closest centerline in scale space. Then, NMS was utilized to extract road centerline in the scale space. Though it can deliver satisfactory performance, discontinuities and topological errors were presented in the centerline network. To alleviate these problems, Sironi *et al.* [39] introduced a projection-based method. It projects the patches of the score map to their nearest neighbors in a set of ground-truth training patches. This method can achieve spatially and geometrically consistent centerline result.

Although many methods have been proposed to tackle the road detection and road centerline extraction problem, the existing methods have some deficiencies. In terms of the road detection task, most existing methods cannot extract the road area well in the heterogeneous areas or under the occlusions of trees and cars; in terms of the road centerline extraction task, the existing centerline extraction methods produce small spurs around the centerline, or cannot extract centerline well around the road intersections. To alleviate the above-mentioned shortcomings, a cascaded deep neural network is proposed to tackle the road detection and centerline extraction problem simultaneously. The proposed method can achieve homogenous road result even in the heterogeneous areas or under the occlusions of trees and cars. In addition, our method can overcome the deficiencies of the existing centerline extraction methods. Thus, it can achieve accurate, smooth, and complete road centerline network.

III. Preliminaries

Convolutional neural network consists of a series of convolutional operations, pooling operations, and nonlinear operations. It has following characteristics.

1) *Local Connectivity:* Like the perception mechanism of human brain, convolutional network exploits spatially local connectivity pattern between neurons of the adjacent layers, that is, each neuron is connected to only a small region of the input layer.
2) *Shared Weights:* In convolutional network, each filter is replicated across the entire visual field. These replicated units share the same parameters (i.e., weight and bias), which greatly reduces the number of parameters.
3) *Hierarchical Feature:* The multilayer structure enables the convolutional network to acquire low-level feature, middle-level feature, and high-level semantic feature.
4) *Task-Driven Feature:* In convolutional network, the feature extraction module and classifier module are integrated in one framework; thus, the learned feature is more suited to the specific task than those hand-crafted features, such as HOG [40] and SIFT [41].

Compared with the conventional multilayer perceptron (MLP), which only consists of fully connected layers, the convolutional network has less parameters due to its local connectivity characteristic. For example, for a 300×300 image, we assume that there are ten hidden neurons. There are $300 \times 300 \times 10 = 900\,000$ weight parameters for MLP. In convolutional network, if we use 10×10 local connectivity pattern, the number of weight parameters is $10 \times 10 \times 10 = 1000$. Thus, the number of weight parameters in convolutional network is only $1/900$ the number of weight parameters in MLP.

In the following, we will introduce some basic modules mentioned in the following network.

Convolutional Layer: The convolutional (Conv) layer computes the output of neurons that are connected to local area in the previous layer. Each output is computed via a dot product between their weights and the area they are connected to in the previous layer. Like the VGG16 [42] network, we use the 3×3 Conv layer in our experiments.

Batch Normalization Layer: The batch normalization (BN) was introduced by Ioffe and Szegedy [43] to avoid gradient vanishing and reduce internal covariate shift (the change in the input distribution to a learning system). Like the whitening technique, BN normalizes the distribution of each input feature in each layer across each minibatch [43] to Gaussian distribution with zero-mean and unit variance. It can greatly accelerate the learning process of deep neural network. In addition, it allows us to use much higher learning rates and be less careful about initialization [43].

Relu Layer: The rectified linear units (Relu) apply an elementwise activation as $\max(0, x)$, which thresholds the nonpositive value as zero and makes the positive value remain unchanged. Krizhevsky *et al.* [20] demonstrated that training the DCNNs with Relu is several times faster than their equivalents with $tanh$ or $sigmoid$ units.

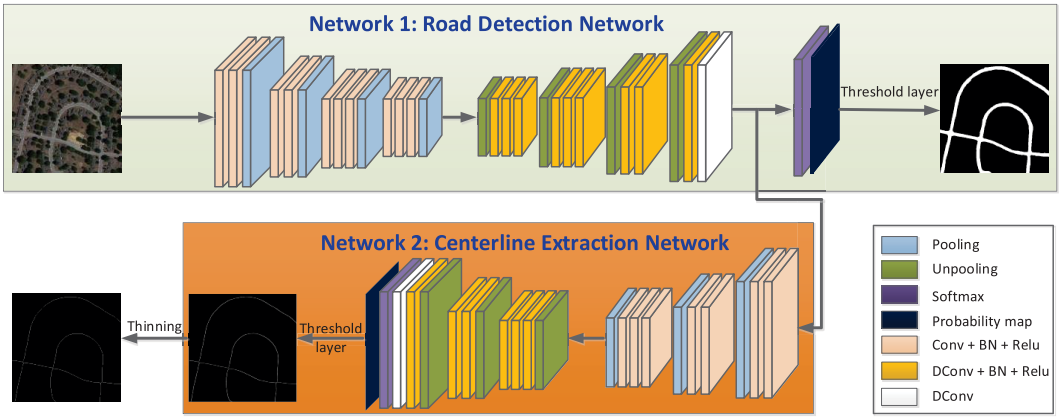


Fig. 1. Flowchart of CasNet. It contains two convolutional neural networks: road detection network and centerline extraction network. The detailed architecture of the two networks is listed in Tables I and II, respectively.

Pooling Layer: The pooling layer performs a subsampling operation along the spatial dimensions, resulting in reduced-sized feature maps. In our experiments, we use the size of 2×2 max-pooling layer.

Unpooling Layer: The unpooling layer performs the reverse operation of pooling layer. In the pooling operation, the locations that are the maximum activations are recorded in switch variables. Then, in the unpooling operation, these switch variables are employed to place each activation back to its original pooled location [23].

Deconvolutional Layer: After unpooling operation, enlarged yet sparse activation maps are obtained. The deconvolutional (DConv) layer aims to densify these sparse activations through convolutionlike operations with multiple learned filters [23], [44]. Contrary to convolutional layer, DConv layer connects a single input activation with multiple outputs.

Threshold Layer: The threshold layer applies an element-wise threshold operation to the probability maps. The values above the given threshold will be assigned as one and the others as zero.

IV. Cascaded Convolutional Network

In this section, we first describe the proposed CasNet architecture, which consists of road detection network and centerline extraction network. Then, the learning algorithm is introduced to train the CasNet via an end-to-end training strategy. Finally, the inference stage is presented to simultaneously achieve the road detection result and centerline extraction result.

A. CasNet Architecture

Fig. 1 shows the detailed configuration of the proposed CasNet. CasNet is composed of two convolutional networks—road detection network and centerline extraction network. As Fig. 1 shows, the convolutional network consists of an encoder network, the corresponding decoder network, and a softmax layer. The encoder network corresponds to feature extractor that transforms the input image to multidimensional shrinking feature maps. These feature maps contain semantic information of the input image. Then, with the help of the unpooling and deconvolutional operations, the decoder network upsamples these feature maps extracted from the encoder network back to the same size of input image. Finally, a softmax classifier is employed to obtain the final output of the image, which is two probability maps indicating the likelihood of each pixel that belongs to the road class and nonroad class, respectively.

1) Road Detection Network: Table I summarizes the detailed configuration of the proposed road detection network. Actually, our network architecture is substantially smaller than the ones commonly used in the VGG16-based semantic segmentation network [23], [24]. We believe that such a small network is more appropriate for the road detection task due to the following reasons. First, road detection task aims to distinguish only two categories, i.e., road and background, which is an easier task than general semantic segmentation task (e.g., 21 categories for PASCAL VOC 2012 [45]). Second, to achieve better performance, the parameters in VGG16-based semantic segmentation network [23], [24] are typically pretrained on the large ImageNet object classification data set [46]. However, road data set is much smaller than the ImageNet data set. Thus, VGG16-based network may not be fully trained with this small road data set. Furthermore, the semantic categories in the ImageNet and the road in the remote sensing images are from thoroughly different domains; thus, the road detection network should be trained from scratch without utilizing the pretrained model on ImageNet data set. Finally, a smaller network can reduce memory consumption and shorten inference time.

As Table I shows, our encoder network has ten convolutional layers altogether, BN layer and Relu layer are attached to each convolutional layer, and one pooling operation is sometimes performed between convolutions. The decoder network is a mirrored version of the encoder network, which consists of multiple series of unpooling, deconvolutional, Relu, and BN layers. A softmax layer is attached to the encoder–decoder network to transform the output to the probability maps. In this section, to train the road detection network, cross entropy loss is utilized, which is defined as

$$\mathrm{L}_{\mathrm{seg}}(y, f(x), \boldsymbol{\theta}_1) = \sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{k=1}^{K} -1(y_i^j = k)\ \log\ p_k\left(x_i^j\right) \tag{1}$$

TABLE I

DETAILED CONFIGURATION OF ROAD DETECTION NETWORK. "CONV" AND "DCONV" DENOTE THE CONVOLUTIONAL LAYER AND DECONVOLUTIONAL LAYER IN ENCODER NETWORK AND DECODER NETWORK, RESPECTIVELY. THE BN LAYERS AND RELU LAYERS ARE OMITTED FROM THE TABLE FOR BREVITY

name	kernel size	stride	pad	output size
Input	–	–	–	$300 \times 300 \times 3$
Conv1-1	3×3	1	1	$300 \times 300 \times 32$
Conv1-2	3×3	1	1	$300 \times 300 \times 32$
Pool1	2×2	2	0	$150 \times 150 \times 32$
Conv2-1	3×3	1	1	$150 \times 150 \times 64$
Conv2-2	3×3	1	1	$150 \times 150 \times 64$
Pool2	2×2	2	0	$75 \times 75 \times 64$
Conv3-1	3×3	1	1	$75 \times 75 \times 128$
Conv3-2	3×3	1	1	$75 \times 75 \times 128$
Conv3-3	3×3	1	1	$75 \times 75 \times 128$
Pool3	2×2	2	0	$38 \times 38 \times 128$
Conv4-1	3×3	1	1	$38 \times 38 \times 256$
Conv4-2	3×3	1	1	$38 \times 38 \times 256$
Conv4-3	3×3	1	1	$38 \times 38 \times 256$
Pool4	2×2	2	0	$19 \times 19 \times 256$
Unpool4	2×2	2	0	$38 \times 38 \times 256$
DConv4-1	3×3	1	1	$38 \times 38 \times 256$
DConv4-2	3×3	1	1	$38 \times 38 \times 256$
DConv4-3	3×3	1	1	$38 \times 38 \times 128$
Unpool3	2×2	2	0	$75 \times 75 \times 128$
DConv3-1	3×3	1	1	$75 \times 75 \times 128$
DConv3-2	3×3	1	1	$75 \times 75 \times 128$
DConv3-3	3×3	1	1	$75 \times 75 \times 64$
Unpool2	2×2	2	0	$150 \times 150 \times 64$
DConv2-1	3×3	1	1	$150 \times 150 \times 64$
DConv2-2	3×3	1	1	$150 \times 150 \times 32$
Unpool1	2×2	2	0	$300 \times 300 \times 32$
DConv1-1	3×3	1	1	$300 \times 300 \times 32$
DConv1-2	3×3	1	1	$300 \times 300 \times 32$
Output	1×1	1	1	$300 \times 300 \times 2$
Softmax	–	–	–	$300 \times 300 \times 2$

where $\boldsymbol{\theta}_1$ represents the parameters of road detection network; M is the minibatch size; N is the number of pixels in each patch; K is the number of classes (here, $K = 2$); $1(y = k)$ is an indicator function, it takes 1 when $y = k$, and 0 otherwise; x_i^j is the jth pixel in the ith patch and y_i^j is the ground-truth label of x_i^j; $f(x_i^j)$ is the output of the last deconvolutional layer at pixel x_i^j (see Fig. 1); and $p_k(x_i^j)$ is the probability of pixel x_i^j being the kth class, which is defined as

$$p_k(x_i^j) = \frac{\exp\left(f_k\left(x_i^j\right)\right)}{\sum_{l=1}^{K} \exp\left(f_l\left(x_i^j\right)\right)}. \tag{2}$$

2) Centerline Extraction Network: Table II summarizes the detailed architecture of the centerline extraction network. Similar to the road detection network, it consists of encoder network, decoder network, and a softmax layer, while it is much smaller than road detection network. We choose a relatively small centerline extraction network for two reasons. On the one hand, the feature maps produced by the last deconvolutional layer in the road detection network have less complicated backgrounds than the original image; thus, a relatively small network is adequate to tackle the centerline extraction task. On the other hand,

TABLE II

DETAILED CONFIGURATION OF CENTERLINE EXTRACTION NETWORK. "CONV" AND "DCONV" DENOTE THE CONVOLUTIONAL LAYER AND DECONVOLUTIONAL LAYER IN ENCODER NETWORK AND DECODER NETWORK, RESPECTIVELY. THE BN LAYERS AND RELU LAYERS ARE OMITTED FROM THE TABLE FOR BREVITY

name	kernel size	stride	pad	output size
Input	–	–	–	$300 \times 300 \times 2$
Conv1-1	3×3	1	1	$300 \times 300 \times 32$
Conv1-2	3×3	1	1	$300 \times 300 \times 32$
Pool1	2×2	2	0	$150 \times 150 \times 32$
Conv2-1	3×3	1	1	$150 \times 150 \times 64$
Conv2-2	3×3	1	1	$150 \times 150 \times 64$
Pool2	2×2	2	0	$75 \times 75 \times 64$
Conv3-1	3×3	1	1	$75 \times 75 \times 128$
Conv3-2	3×3	1	1	$75 \times 75 \times 128$
Conv3-3	3×3	1	1	$75 \times 75 \times 128$
Pool3	2×2	2	0	$38 \times 38 \times 128$
Unpool3	2×2	2	0	$75 \times 75 \times 128$
DConv3-1	3×3	1	1	$75 \times 75 \times 128$
DConv3-2	3×3	1	1	$75 \times 75 \times 128$
DConv3-3	3×3	1	1	$75 \times 75 \times 64$
Unpool2	2×2	2	0	$150 \times 150 \times 64$
DConv2-1	3×3	1	1	$150 \times 150 \times 64$
DConv2-2	3×3	1	1	$150 \times 150 \times 32$
Unpool1	2×2	2	0	$300 \times 300 \times 32$
DConv1-1	3×3	1	1	$300 \times 300 \times 32$
DConv1-2	3×3	1	1	$300 \times 300 \times 32$
Output	1×1	1	1	$300 \times 300 \times 2$
Softmax	–	–	–	$300 \times 300 \times 2$

compared with the road detection task, there are fewer positive pixels (i.e., centerline pixels) to train the centerline extraction network. In this case, overfitting may occur with a relatively deep model. Therefore, a small network is preferred.

As is shown in Fig. 1 and Table II, the centerline network takes feature maps produced by the last deconvolutional layer in road detection network as input. It has seven convolutional layers in the encoder network, and another seven layers in the corresponding decoder network. The detailed visual configuration of all the layers is shown in Fig. 1. To train the centerline extraction network, its cross entropy loss is defined as follows:

$$\mathrm{L}_{\mathrm{cen}}(z, h(x), \boldsymbol{\theta}_2) = \sum_{i=1}^{M} \sum_{j=1}^{N} \sum_{k=1}^{K} -1\left(z_i^j = k\right) \log\ p_k\left(f\left(x_i^j\right)\right) \tag{3}$$

where $\boldsymbol{\theta}_2$ represents the parameters of centerline extraction network; M, N, and K are defined in the same way as the aforementioned section; $f(x_i^j)$ is output of the last deconvolutional layer in the road detection network at pixel x_i^j (see Fig. 1); $h(f(x_i^j))$ is output of the last deconvolutional layer in the centerline extraction network; z_i^j is the centerline ground-truth label of x_i^j; and $p_k(f(x_i^j))$ is the road centerline probability of pixel x_i^j being the kth class, which is defined as

$$p_k(f(x_i^j)) = \frac{\exp\left(h_k\left(f\left(x_i^j\right)\right)\right)}{\sum_{l=1}^{K} \exp\left(h_l\left(f\left(x_i^j\right)\right)\right)}. \tag{4}$$

B. Learning Algorithm

The goal of our learning algorithm is to train the CasNet to simultaneously achieve road detection task and centerline extraction task. In our experiments, we use a new and challenging road data set, which consists of original images, the corresponding segmentation reference images, and centerline reference images. The detailed descriptions of the data set will be presented in Section IV-C.

The overall loss function in CasNet is the summation of losses in (1) and (3), which is defined as follows:

$$\text{Loss}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2) = \text{L}_{\text{seg}}(y, f(x), \boldsymbol{\theta}_1) + \text{L}_{\text{cen}}(z, h(x), \boldsymbol{\theta}_2). \quad (5)$$

Like the weight strategies used by Dai *et al.* [47] in their network, the balance weight of 1 is also implicitly used among the above-mentioned two terms.

To train the CasNet with end-to-end manners, $\text{Loss}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$ is minimized with respect to the CasNet parameters $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$. We should calculate the derivative of the loss in (5) to the different component layers with chain rule, and then update the parameters layer-by-layer with the backpropagation strategies. For clarity, we only illustrate the derivative of loss to the output of the final deconvolutional layer in the road detection network and centerline extraction network (see Fig. 1). The derivative of $\text{Loss}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$ to the output [i.e., $h_k(f(x_i^j))$] of the last deconvlutional layer in the centerline extraction network is defined as

$$\begin{aligned}\frac{\partial\ \text{Loss}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)}{\partial\ h_k\left(f\left(x_i^j\right)\right)} &= \frac{\partial\ \text{L}_{\text{seg}}(y, f(x), \boldsymbol{\theta}_1)}{\partial\ h_k\left(f\left(x_i^j\right)\right)} + \frac{\partial\ \text{L}_{\text{cen}}(z, h(x), \boldsymbol{\theta}_2)}{\partial\ h_k\left(f\left(x_i^j\right)\right)} \\ &= \sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{k=1}^{K} -1\left(z_i^j = k\right)\left(1 - p_k\left(f\left(x_i^j\right)\right)\right).\end{aligned} \quad (6)$$

The specific derivation can be referred in Appendix A of Supplementary Material. It should be noted that $\partial\ \text{L}_{\text{seg}}(y, f(x), \boldsymbol{\theta}_1)/\partial\ h_k(f(x_i^j)) = 0$; thus, there is only one backpropagation flow along the negative direction of the centerline extraction network.

The derivative of $\text{Loss}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)$ to the output [i.e., $f_k(x_i^j)$] of the last deconvlutional layer in the road detection network is defined as

$$\begin{aligned}\frac{\partial\ \text{Loss}(\boldsymbol{\theta}_1, \boldsymbol{\theta}_2)}{\partial\ f_k\left(x_i^j\right)} &= \frac{\partial\ \text{L}_{\text{seg}}(y, f(x), \boldsymbol{\theta}_1)}{\partial\ f_k\left(x_i^j\right)} + \frac{\partial\ \text{L}_{\text{cen}}(z, h(x), \boldsymbol{\theta}_2)}{\partial\ f_k\left(x_i^j\right)} \\ &= \sum_{i=1}^{M}\sum_{j=1}^{N}\sum_{k=1}^{K} -1\left(y_i^j = k\right)\left(1 - p_k\left(x_i^j\right)\right) \\ &\quad + \nabla_{f_k\left(x_i^j\right)}^{\text{L}_{\text{cen}}(z, h(x), \boldsymbol{\theta}_2)}.\end{aligned} \quad (7)$$

As (7) shows, there are two backpropagation flows along the negative direction of the road detection network. One is from $\partial\ \text{L}_{\text{seg}}(y, f(x), \boldsymbol{\theta}_1)/\partial\ f_k(x_i^j)$, and its detailed derivation can be referred in Appendix B of Supplementary Material. The other is from the $\nabla_{f_k(x_i^j)}^{\text{L}_{\text{cen}}(z, h(x), \boldsymbol{\theta}_2)}$, which can be obtained with the chain rule as follows:

$$\nabla_{f_k\left(x_i^j\right)}^{\text{L}_{\text{cen}}(z, h(x), \boldsymbol{\theta}_2)} = \frac{\partial\ \text{L}_{\text{cen}}(z, h(x), \boldsymbol{\theta}_2)}{\partial\ h_k\left(f\left(x_i^j\right)\right)} \frac{\partial\ h_k\left(f\left(x_i^j\right)\right)}{\partial\ f_k\left(x_i^j\right)}. \quad (8)$$

We employ an end-to-end strategy to train the road detection network and the centerline extraction network simultaneously. Specifically, one separate training data set is utilized for each subtask. For the road detection task, original images and the corresponding segmentation reference images are used. For the centerline extraction task, original images and its corresponding centerline reference images are utilized.

In the experiments, we implement the CasNet based on the Caffe [48] framework. The 300×300 patches are utilized for training the CasNet. All the network parameters in CasNet are initialized using the techniques introduced by He *et al.* [49]. Due to the limit of GPU memory, we set the minibatch size as 4. To train the CasNet, we use stochastic gradient descent with the fixed learning rate of 0.01 and drop the learning rate by a factor of 0.1 every 10K iterations. The dropout ratio is 0.5 and the momentum is 0.95. We stop training until the training loss converges.

C. Inference

In the reference stage, the road detection and centerline extraction can be simultaneously performed through the CasNet. The inference procedures of road detection and centerline extraction are shown in Fig. 1. In the reference of road detection, given an input image, it is transformed to probability maps through the road detection network. Then, a threshold layer is attached to convert these probability maps into two-class maps. We use 0.5 as the threshold in our experiments. In the inference stage of centerline extraction network, given the feature maps (the output of the last deconvolutional layer) produced by the road detection network, the centerline network transforms these feature maps into centerline-based probability maps. Then, to obtain smooth, complete, and single-pixel width road centerline network, morphological thinning algorithm is performed.

V. Data Set Descriptions

This section introduces the detailed information of the data set used to train the CasNet. It should be noted that few VHR urban road data set is publicly available. Thus, we collected 224 VHR images from *Google Earth* [50], and we manually labeled their road segmentation reference maps and corresponding centerline reference maps. To the best of our knowledge, this is the largest road data set with accurate segmentation maps and centerline maps. This data set will be publicly available for further research. Actually, our data set contains two subdata sets: road detection data set and road centerline data set.

A. Road Detection Data Set

As Fig. 2 shows, the original images in our data set are with a spatial resolution of 1.2 m per pixel. The second row in Fig. 2 shows the corresponding segmentation maps. In this data set, there are at least 600×600 pixels in each image, and the road width is about 12–15 pixels. Most original images in our data set are under complex backgrounds and occlusions of cars and trees, which make the road detection task very challenging.

Fig. 2. Illustration of two representative images, their segmentation reference maps, and centerline reference maps. The first row shows the original images, which are under complex backgrounds and occlusions of cars and trees. The second row shows the segmentation reference maps. For better visual effects and comparison, the third row illustrates the superposition between road segmentation reference map and road centerline reference map. The fourth row illustrates the corresponding close-ups of the blue rectangles in the third row.

B. Road Centerline Data Set

The third row in Fig. 2 shows the corresponding road centreline map of original image in the first row. For better visual effects, we show the superposition between the road segmentation reference map and road centerline reference map. It shows that the road centerline in our data set is with single-pixel width; meanwhile, our centerline network is indeed at the center positions of the road. The close-ups in the fourth row can be more obvious to reflect these properties.

VI. Experiments and Evaluation

In this section, experiment setting, comparing methods, and extensive experiments in both visual and quantitative comparisons are presented.

A. Experiment Setting

We use the road detection data set and road centerline data set to train the road detection network and centerline extraction network, respectively. Our road data set consists of 224 aerial images in total. We randomly split the data into a training set of 180 images, a validation set of 14 images, and a test set of 30 images. This data set is relatively small to train the CasNet. To reduce overfitting and train an appropriate model, some data augmentation strategies and regularization approaches are introduced.

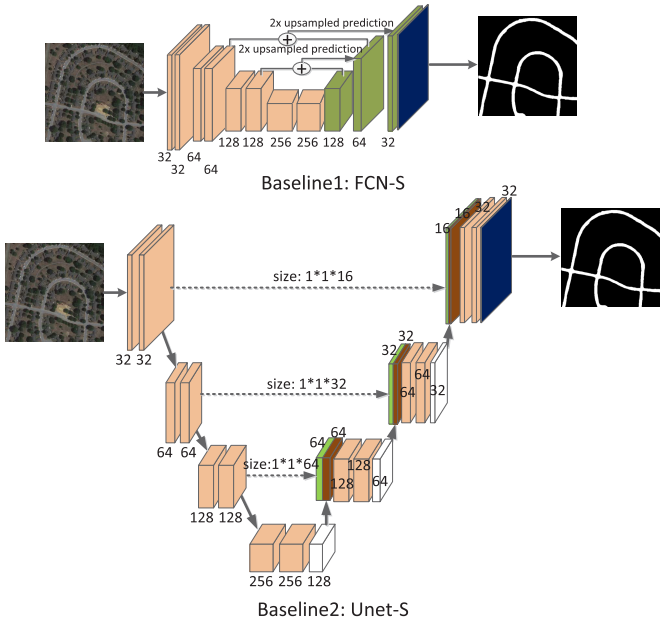


Fig. 3. Architecture of two baseline networks. In FCN-S, feature maps summation and 2× upsampled prediction are utilized. In Unet-S, concat layer is employed to concatenate two groups of feature maps. The dotted line represents convolution operation with 1×1 kernel.

Data Augmentation: In our experiments, 300×300 patches are employed to train the CasNet. For the training set and validation set, we use a two-stage method to augment the data. In the first stage, given an image, we extract 5 fixed-position patches (i.e., the 4 corner patches and the center patch) and another 20 random free-position patches. In the second stage, for each patch, we rotate it at the step of 90°, and then flip these rotated patches in horizontal and vertical reflections. For each patch, seven another different patches are produced by rotations and reflections in the second stage. The above-mentioned method increases the size of our training set or validation set by a factor of 200 [25(sampled patches) × 2(horizontal and vertical flips) × 4(rotated patches)]. Thus, the number of patches in our training set and validation set is 36 000 and 2800, respectively.

To avoid overfitting, in our experiments, dropout strategies [51] and dilated strategies [22] (also called "hole" algorithm) are added in the last four decoder layers in the road detection network and centerline extraction network. These strategies provide a computationally inexpensive yet powerful regularization to the network.

B. Comparing Algorithms

To verify the performance, the proposed CasNet is compared with other state-of-the-art methods in two aspects: road detection comparison and road centerline comparison.

1) Segmentation Comparing Algorithms: For the road detection comparison, we compare the proposed algorithm with three state-of-the-art road segmentation algorithms and three DCNN baselines. Here we use the suffix "-S" to denote segmentation. The main information of these methods (including our method) are summarized as follows.

1) *Ours-S:* As Fig. 1 shows, the road detection network and threshold layer are employed to obtain the final road detection results.

2) *Huang-S:* One multiscale classification-based algorithm was introduced to road extraction task by Huang and Zhang [13]. In their method, multiscale structural features and SVM were employed.

3) *Shi-S:* To obtain the road detection result, Shi *et al.* [18] fused the spectral-spatial features and homogeneous properties via SVM.

4) *Cheng-S:* Cheng *et al.* [14] introduced road detection method via multiscale segmentation. In this method, fused multiscale collaborative representation and graph cuts algorithm were used.

5) *FCN-S:* Long *et al.* [21] proposed FCN for the semantic segmentation, which had achieved state-of-the-art performance. In our experiments, as Fig. 3 shows, we modified the architecture of FCN with less feature maps and less convolution layers. We trained this network using the same data set as our CasNet. We denote this modified version as "FCN-S."

6) *Unet-S:* Based on the FCN, Ronneberger *et al.* [52] proposed a new network architecture, which is called "U-net." U-net had achieved state-of-the-art performance in biomedical image segmentation. In our experiments, as Fig. 3 shows, we modified the U-net with less features maps and less layers. In addition, we added a 1×1 convolutional layer to reduce the feature maps rather than concatenate the feature maps directly. We denote this modified version as "Unet-S."

7) *Baseline-S:* We only trained the road detection network (network 1 in Fig. 1) with the original images and their segmentation reference images. We denote this method as Baseline-S.

2) Centerline Comparing Algorithms: For the road centerline comparison, four state-of-the-art approaches and one baseline method are introduced to compare with the proposed method. The suffix "-C" is used to denote centerline. The details of all these methods are listed as follows.

1) *Ours-C:* As Fig. 1 shows, we use CasNet and morphological thinning algorithm to obtain the final road centerline results.

2) *Huang-C:* Huang and Zhang [13] used the above-mentioned "Huang-S" algorithm to obtain road detection results. After that, to obtain centerline, morphological thinning algorithm was used.

3) *Miao-C:* A spectral and shape features-based road extraction method was proposed by Miao *et al.* [10]. It can extract homogenous regions well. Then, to overcome the shortcomings of morphological thinning algorithm, they introduced MARSs to obtain smooth road centerline network.

4) *Shi-C:* Shi-S was utilized to obtain road detection results. After that, a local linear kernel smoothing regression algorithm [18] was introduced to extract centerline network.

5) *Cheng-C:* We utilized the overall architecture of Cheng *et al.* [14] to obtain the final road centerline. In their method, Cheng-S was first introduced to generate road detection results. Then, tensor voting, NMS, and fitting-based connection algorithm were proposed to obtain road centerline network.

6) *Baseline-C:* We only trained the road centerline extraction network (network 2 in Fig. 1) with the original images and their centerline reference images. We denote this method as Baseline-C.

It should be noted that there is only one key parameter in the CasNet that is the threshold value in threshold layer. In our experiments, we find that satisfactory results can be obtained when we set it as 0.5. Thus, we keep this parameter fixed in the following experiments. For the comparing algorithms, we adjust the parameters according to their original papers to gain the best performance for fair comparison.

C. Evaluation Metrics

To assess the quantitative performance in both road detection and road centerline extraction, three benchmark metrics [53] are introduced, i.e., *Completeness* (COM), *Correctness* (COR), and *Quality* (Q). COM measures the proportion of matched areas in the reference map. COR represents the percentage of matched road areas in the segmentation map. Q is an overall metric, which combines COM and COR. They are defined as

$$\mathrm{COM} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}}, \quad \mathrm{COR} = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}, \quad Q = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}+\mathrm{FP}} \tag{9}$$

where TP, FP, and FN represent the true positive, false positive, and false negative, respectively.

To evaluate the performance of different comparing algorithms on road detection, we compare the extracted road map with its segmentation reference map in the corresponding locations.

Due to the deviation between the manually labeled centerline and the real centerline, it is not an appropriate means to evaluate pixel comparisons between the extracted centerline and the centerline reference map. To assess the performance of different comparing algorithms on road centerline, a "buffer method" [54] is introduced. In this method, by comparing with the reference centerline map, those areas in the predicted centerline map, which are within a given buffer width ρ to the reference centerline, are considered as the matched areas. That is, a predicted centerline point is considered to be a TP if it is within ρ-pixel distance from one reference centerline point.

D. Comparison of Road Detection

To evaluate the effectiveness of the proposed CasNet on road detection, qualitative comparisons and quantitative comparisons with the state-of-the-art methods are exhibited in Fig. 4 and Table III, respectively.

Fig. 4. Visual comparisons of road area extraction results with different comparing algorithms. There are four rows and eight columns of subfigures. The first three rows illustrate the results of three images, and the fourth row shows the close-ups of the corresponding regions in the third row. (a) Original image. (b) Result of Huang-S [13]. (c) Result of Shi-S [18]. (d) Result of Cheng-S [14]. (e) Result of FCN-S [21]. (f) Result of Unet-S [52]. (g) Result of Baseline-S. (h) Result of Ours-S. The fourth row shows the close-ups of the black rectangles in the third row. (Green) TP. (Red) FP. (Blue) FN.

TABLE III

QUANTITATIVE COMPARISONS AMONG DIFFERENT METHODS ON ROAD DETECTION, WHERE THE VALUES IN BOLD ARE THE BEST AND THE VALUES UNDERLINED ARE THE SECOND BEST. IT SHOULD BE NOTED THAT THE LAST COLUMN IS THE AVERAGE PERFORMANCE OF ALL IMAGES IN TEST SET

	Image 1			Image 2			Image 3			Avg.(test set)		
	COM	COR	Q	COM	COR	Q	COM	COR	Q	COM	COR	Q
Huang-S	0.9154	0.8246	0.7662	0.9137	0.8981	0.8279	0.9244	0.7488	0.7056	0.9173	0.8027	0.7754
Shi-S	0.8504	**0.9613**	0.8237	0.8631	0.9478	0.8507	0.7583	**0.9893**	0.7522	0.8476	**0.9571**	0.8175
Cheng-S	0.9690	0.8554	0.8324	0.9789	0.8891	0.8725	0.9635	0.8627	0.8352	0.9648	0.8639	0.8407
FCN-S	0.9397	0.8027	0.7859	0.9624	0.8437	0.8254	0.8459	0.9756	0.8284	0.9307	0.8478	0.8067
Unet-S	**0.9848**	0.8303	0.8204	**0.9942**	0.8432	0.8374	**0.9696**	0.7953	0.7761	**0.9708**	0.8326	0.8186
Baseline-S	0.9476	0.8807	0.8630	0.9781	0.8907	0.8817	0.9573	0.8836	0.8719	0.9537	0.8861	0.8613
Ours-S	0.9427	0.9367	**0.9124**	0.9634	**0.9487**	**0.9146**	0.9487	0.9416	**0.8960**	0.9418	0.9214	**0.8884**

Fig. 4 shows the comparing results of different methods in visual performance. It should be noted that Huang-S and Cheng-S are superpixel-based methods, Shi-S is pixel-based method, and others are based on DCNN. As we shall see, Shi-S is sensitive to the occlusions of cars and trees [see blue areas in Fig. 4(c)]. Huang-S and Cheng-S can alleviate the effectiveness of occlusions to some extent; thus, they can achieve more coherent results, while these methods tend to bring in more FPs [see the red areas of close-ups in Fig. 4(b) and (d)]. FCN-S can achieve relatively coherent road areas, while it tends to be sensitive to the occlusions and brings in some FPs. This is because FCN-S resizes the feature maps back to the input size with simple upsampled predictions (see the architecture of FCN-S in Fig. 3), which may be not enough to deal with occlusions. To alleviate this shortcoming, Unet-S introduces convolutional operations after upsampled predictions (see the architecture of Unet-S in Fig. 3). Thus, Unet-S shows better robustness to the occlusions than FCN-S. Though FCN-S and Unet-S can achieve satisfactory results, these methods bring in some FPs [see the red areas in Fig. 4(e) and (f)]. To overcome this shortcoming, the Baseline-S and Ours-S utilize the pooling–unpooling operations, which downsize the feature maps and memorize the maximum activation positions simultaneously. Both of them show robustness against the occlusions, while Ours-S brings in less FPs than the Baseline-S [see the close-ups in Fig. 4(g) and (h)]. From Fig. 4, it shows that Ours-S can achieve more satisfactory and coherent road detection results than other comparing methods. Besides, our approach is more robust against the occlusions of cars and trees.

Table III summarizes the quantitative performance of road detection results with different comparing methods. Compared with COM and COR, Q is an overall metric. In Table III, the first three columns are the performance of three sampled images, and the last column is the average performance of all the images in test set. In each metric term, the value in bold is the best, while the underlined value is the second best. As Table III shows, though some comparing algorithms achieve

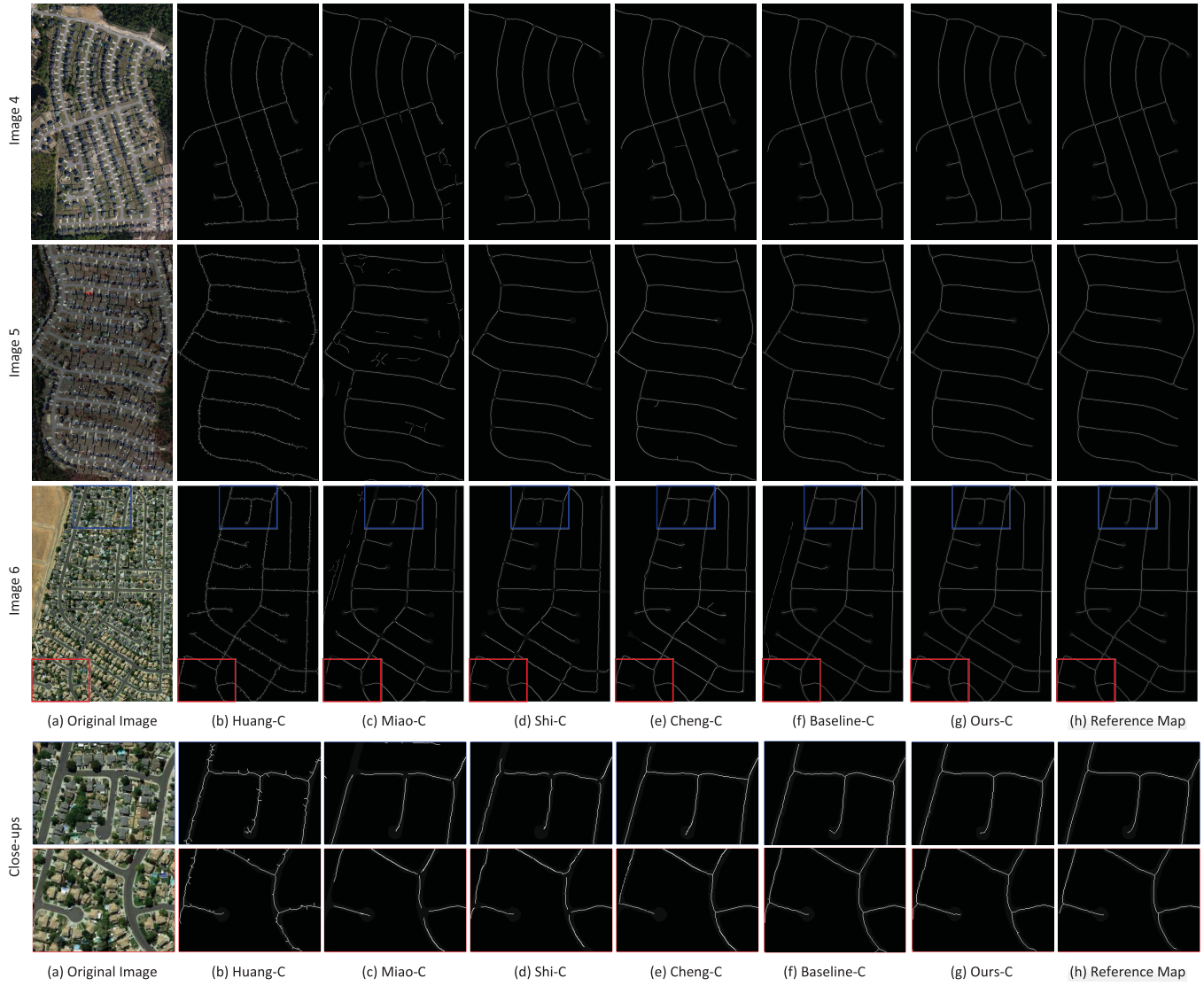


Fig. 5. Visual comparisons of road centerline extraction results with different comparing algorithms. There are five rows and eight columns of subfigures. The first three rows illustrate the results of three sampled images, and the last two rows illustrate the close-ups of the corresponding regions of the red and blue rectangles in the third row. (a) Original image. (b) Result of Huang-C [13]. (c) Result of Miao-C [10]. (d) Result of Shi-C [18]. (e) Result of Cheng-C [14]. (f) Result of Baseline-C. (g) Result of Ours-C. (h) Result of the reference map.

decent performance in either COM (e.g., Unet-S) or COR (e.g., Shi-S), while their overall metric (i.e., Q) is unsatisfying, Ours-S achieves relatively satisfactory performances both in COM and COR, and thus it generally achieves the best result. Specifically, the average Q of Ours-S is about 3% higher than the Baseline-S, which demonstrates that the more supervised information in the CasNet help the Ours-S achieve better performance than the Baseline-S. Apart from Baseline-S, the average Q of Ours-S is nearly 5% higher than the other best comparing method (i.e., Cheng-S), which demonstrates the validity of Ours-S on road detection task.

E. Comparison of Road Centerline Extraction

Fig. 5 presents the visual performance of road centerline among different comparing algorithms. To better express the contrast effects, the road segmentation reference map is also included in Fig. 5. In Fig. 5, we set the extracted centerline results of comparing methods as 255, the backgrounds as 0, and the segmentation reference area as 40 in gray value. To facilitate comparison, some close-ups are displayed, which are in the last two rows of Fig. 5. It can be seen from Fig. 5, though Huang-C can produce relatively complete road centerline network, it brings in small spurs (see the close-ups in Huang-C) around the centerline, which greatly reduce the smoothness and correctness of the road network. Miao-C can produce relatively smooth road centerline network, while it has two shortcomings. First, it introduces some FPs (the white lines that are out of the road reference map). It is because the homogenous regions are classified as road class in Miao-C. However, in Fig. 5, some forest areas and bare land in original images are also homogenous regions too; thus, it is hard to distinguish them from the real road regions. Second, as a regression-based centerline extraction algorithm, Miao-C cannot link the centerline well around the intersection areas. Like

TABLE IV

QUANTITATIVE COMPARISONS AMONG DIFFERENT METHODS ON CENTERLINE EXTRACTION, WHERE THE VALUES IN BOLD ARE THE BEST AND THE VALUES UNDERLINED ARE THE SECOND BEST. IT SHOULD BE NOTED THAT THE LAST COLUMN IS THE AVERAGE PERFORMANCE OF ALL IMAGES IN TEST SET

	Image 4			Image 5			Image 6			Avg.(test set)		
$\rho = 2$	COM	COR	Q	COM	COR	Q	COM	COR	Q	COM	COR	Q
Huang-C	0.9750	0.8140	0.7974	0.9638	0.7223	0.7033	**0.9667**	0.7467	0.7279	0.9586	0.7381	0.7027
Miao-C	0.9295	0.8155	0.7680	0.8845	0.7047	0.6469	0.8935	0.7241	0.6673	0.8964	0.7178	0.6735
Shi-C	0.9376	0.9119	0.8674	0.9401	0.9198	0.8739	0.8639	0.8486	0.7509	0.8933	0.9069	0.8194
Cheng-C	0.9602	0.9104	0.8857	0.9896	0.9067	0.8876	0.9458	0.8236	0.7827	0.9307	0.8963	0.8362
Baseline-C	0.9418	0.9007	0.8716	0.9274	0.9108	0.8537	0.9326	0.7909	0.7638	0.9244	0.8735	0.8209
Ours-C	**0.9793**	**0.9459**	**0.9217**	**0.9970**	**0.9655**	**0.9532**	0.9573	**0.9428**	**0.9176**	**0.9635**	**0.9542**	**0.9183**
$\rho = 1$	COM	COR	Q	COM	COR	Q	COM	COR	Q	COM	COR	Q
Huang-C	0.8778	0.7328	0.6648	0.8376	0.6278	0.5597	0.8463	0.6538	0.5844	0.8509	0.6837	0.6471
Miao-C	0.8575	0.7523	0.6687	0.7568	0.6203	0.5212	0.7858	0.6369	0.5427	0.8113	0.6735	0.6218
Shi-C	0.8750	0.8696	0.7735	0.8608	0.8451	0.7435	0.7562	0.7336	0.6124	0.8486	0.8417	0.7463
Cheng-C	0.8734	0.8478	0.7635	0.9251	0.8487	0.7834	0.8484	0.7573	0.6672	0.8769	0.8394	0.7631
Baseline-C	0.9071	0.8730	0.8146	0.8935	0.8862	0.8367	0.8869	0.7483	0.6908	0.8904	0.8530	0.7857
Ours-C	**0.9556**	**0.9408**	**0.9070**	**0.9837**	**0.9561**	**0.9320**	**0.9481**	**0.9217**	**0.8917**	**0.9487**	**0.9275**	**0.8963**

Miao-C, Shi-C can extract smooth road centerline, while it also fails to extract road centerline around the intersection areas. Cheng-C can produce relatively smooth and complete road centerline network, and it can extract centerline well around the intersections, while as an NMS-based centerline extraction algorithm, its centerline width may be more than single-pixel width. One common shortcoming for all the above-mentioned comparing methods is that there are some deviations between the extracted centerline and the real centerline. Basline-C can extract smooth road centerline network with single-pixel width, while it brings in some FPs and discontinuity. This is because the Basline-C has much less supervised information (only the centerline supervision) than the Ours-C; it is hard to extract the complete road centerline network well. As the close-ups shows, Ours-C can detect the smooth and complete centerline well; meanwhile, our extracted centerline is more similar to the real centerline than other comparing methods.

Table IV summarizes the quantitative performances of road centerline extraction results with different methods. To better illustrate the contrast effects among different methods, we evaluated the quantitative performances with different buffer widths (i.e., $\rho = 1$ and $\rho = 2$). It can be seen that apart from our method, Shi-C, Cheng-C, and Baseline-C generally achieve better results than other comparing methods. However, in general, Ours-C achieves almost all of the best performance (i.e., the bold values in Table IV) when $\rho = 2$. It should be noted that Ours-C outperforms the second best approach (Cheng-C) by a large margin (about 8% in Q value). To better explore the deviation between the extracted centerline and real centerline among different methods, we also evaluated the performance when $\rho = 1$. As we shall see, Ours-C achieves better performance by a larger margin (i.e., at least 13% in Q value) than all the comparing methods. By comparing the quantitative performance with different buffer widths and the visual performance in Fig. 6, we can see that, apart from Ours-C and Baseline-C, the performance of other comparing methods draws dramatically with the decrease of buffer width. However, in general, Ours-C decreases very little with the decrease of buffer width. It demonstrates that the extracted centerline of Ours-C is more similar to real centerline than the comparing methods.

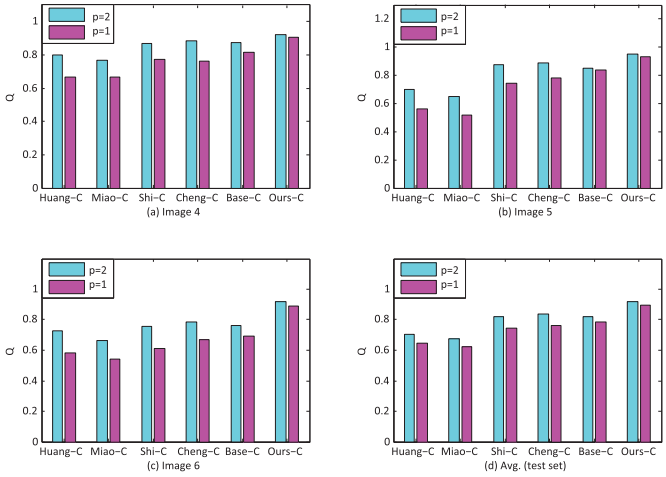


Fig. 6. Visual comparisons of road centerline extraction results with different buffer widths (i.e., $\rho = 1$ and $\rho = 2$). These numeric values are from Table IV. The first three subfigures are the quantitative performance of three sampled images, and the last subfigure is the average performance of all the images in test set.

F. Comparison of Centerline Extraction Algorithms

To better explore the contrasting effects among different centerline extraction algorithms based on the same detection result, we provided the visual and quantitative performance of different centerline extraction algorithms in Figs. 7 and 8 and Table V.

Fig. 7 exhibits the visual performance among different centerline extraction algorithms. To ensure a fair comparison, all the centerline extraction algorithms employ the same road detection results (i.e., the results produced by Ours-S) to obtain the final centerline network. As Miao-C and Shi-C are all the regression-based centerline extraction algorithm, thus in

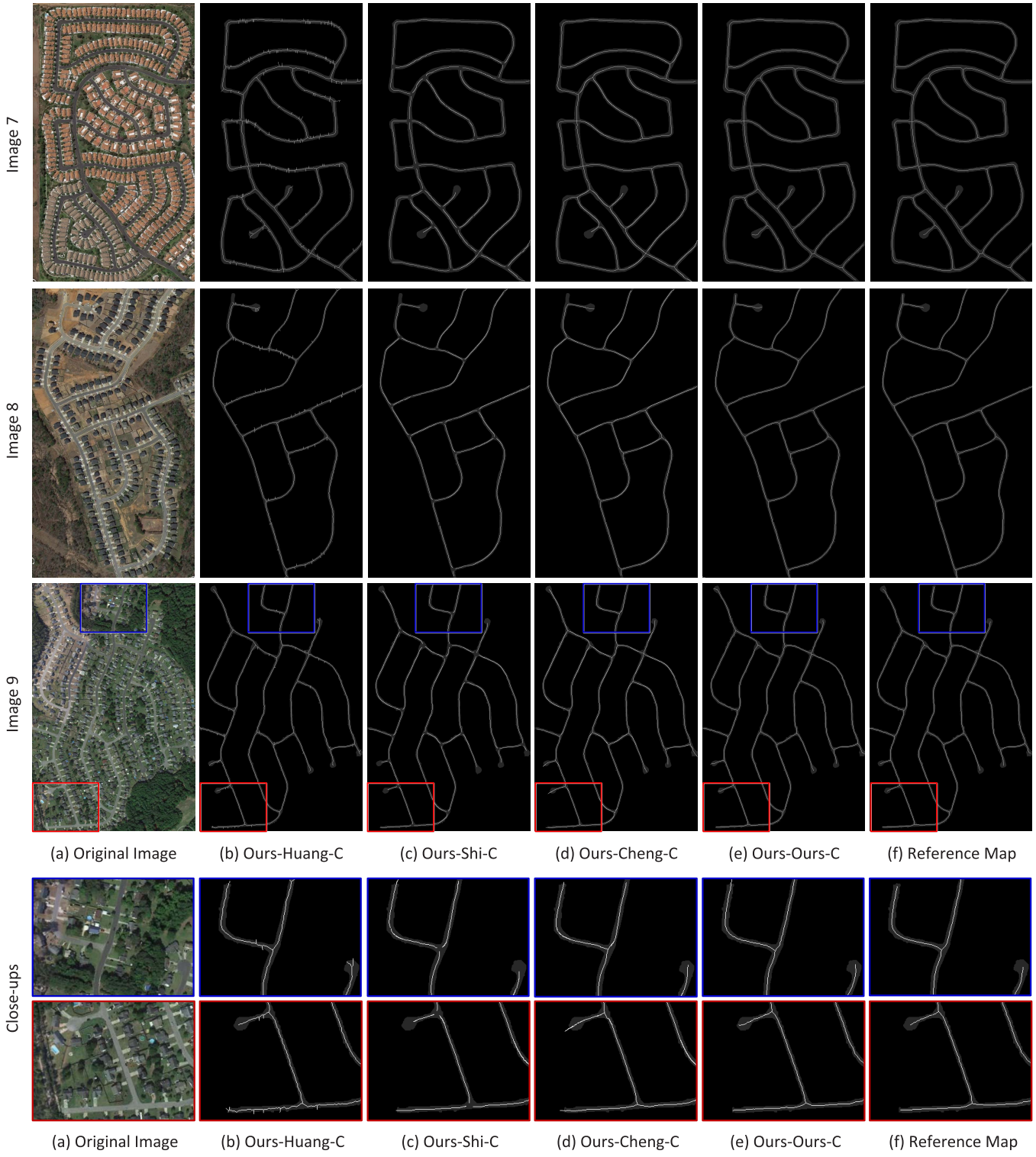


Fig. 7. Visual comparisons of centerline extraction results among different road centerline extraction algorithms with the same road detection results (i.e., extracted result by Ours-S). There are five rows and six columns of subfigures. The first three rows illustrate the results of three sampled images, and the last two rows illustrate the close-ups of the corresponding regions of the red and blue rectangles in the third row. (a) Original image. (b) Result of Ours-Huang-C [13]. (c) Result of Ours-Shi-C [18]. (d) Result of Ours-Cheng-C [14]. (e) Result of Ours-Ours-C. (f) Result of the reference map.

this experiments, we only utilize one of them (i.e., Shi-C). As we shall see, although there are still some spurs in Ours-Huang-C, their number has been much less than Huang-C in Fig. 5. It is because Ours-S produces much smoother road boundaries and less FPs than Huang-S. All these factors lead Ours-Huang-C to produce less spurs than Huang-C. As for Shi-C and Cheng-C, because their original road segmentation algorithms can produce relatively smooth boundaries and small

TABLE V

QUANTITATIVE COMPARISONS AMONG DIFFERENT CENTERLINE ALGORITHMS, WHERE THE VALUES IN BOLD ARE THE BEST AND THE VALUES UNDERLINED ARE THE SECOND BEST. IT SHOULD BE NOTED THAT THE LAST COLUMN IS THE AVERAGE PERFORMANCE OF ALL IMAGES IN TEST SET

	Image 7			Image 8			Image 9			Avg.(test set)		
$\rho = 2$	COM	COR	Q	COM	COR	Q	COM	COR	Q	COM	COR	Q
Ours-Huang-C	0.9589	0.8120	0.7782	0.9755	0.8587	0.8406	0.9384	0.9179	0.8963	0.9413	0.8479	0.8109
Ours-Shi-C	0.9234	0.8607	0.8161	0.9540	0.8732	0.8419	0.9467	0.8976	0.8517	0.9327	0.8810	0.8396
Ours-Cheng-C	**0.9682**	0.8695	0.8443	0.9704	0.8812	0.8637	**0.9918**	0.8855	0.8797	**0.9738**	0.8805	0.8634
Ours-Ours-C	0.9674	**0.9463**	**0.9269**	**0.9764**	**0.9428**	**0.9289**	0.9670	**0.9508**	**0.9346**	0.9635	**0.9542**	**0.9183**
$\rho = 1$	COM	COR	Q	COM	COR	Q	COM	COR	Q	COM	COR	Q
Ours-Huang-C	0.8508	0.7855	0.7028	0.8715	0.8277	0.7392	0.9176	0.8719	0.8086	0.8763	0.8309	0.7534
Ours-Shi-C	0.8115	0.8518	0.7419	0.8857	0.8309	0.7641	0.8916	0.8561	0.7724	0.8719	0.8674	0.7730
Ours-Cheng-C	0.8392	0.8614	0.7604	0.8933	0.8415	0.7784	0.9189	0.8633	0.7926	0.8997	0.8637	0.7908
Ours-Ours-C	**0.9527**	**0.9397**	**0.9157**	**0.9653**	**0.9384**	**0.9183**	**0.9547**	**0.9431**	**0.9276**	**0.9487**	**0.9275**	**0.8963**

number of FPs, it is not obvious to see some evident visual improvement for Ours-Shi-C and Ours-Cheng-C, although they do have some promotions in performance (see Table V). Compared with all the modified methods (i.e., Ours-Huang-C, Ours-Shi-C, and Ours-Cheng-C), Ours-C produces much similar road centerline network to the real centerline network. It demonstrates that our centerline extraction network and the thinning algorithm can produce much better centerline network than state-of-the-art centerline extraction algorithms.

The quantitative performances of different centerline extraction algorithms are summarized in Table V and Fig. 8. In Table V, we displayed numerical performance of three sampled images and average performance of all the images in test sets with modified methods. By comparing the average performances in Tables IV and V, we can see that all the modified methods (i.e., Ours-Huang-C, Ours-Shi-C, and Ours-Cheng-C) achieve better quantitative performance than their original methods (i.e., Huang-C, Shi-C, and Cheng-C) to some degree. Specifically, Ours-Huang-C achieves about 10% improvement in Q than Huang-C, and the other two gain about 3% improvement in Q than their original methods. To better illustrate this improvement in visual effects, Fig. 8 shows intuitive comparisons between original methods and modified methods with buffer width $\rho = 1$ and $\rho = 2$. In Fig. 8, we also display numerical performance of ours though these two values are the same. As we can see, Ours-Huang-C achieves bigger improvements than the other two comparing methods. Although all the modified methods have achieved a certain degree of numerical promotion, our method surpasses the second best method by a large margin (i.e., 5% higher when $\rho = 2$ and 10% higher when $\rho = 1$).

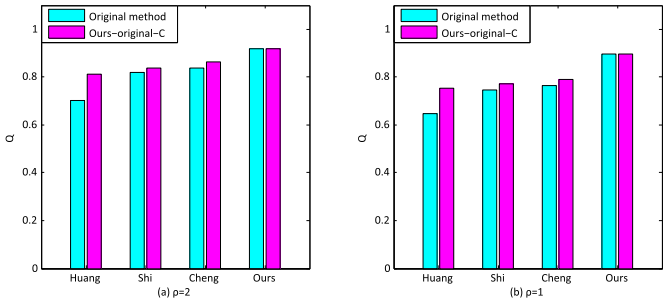


Fig. 8. Visual comparisons of different centerline extraction algorithms with the same road detection results. "Original method" denotes the centerline extraction result with the detection result produced by its corresponding road detection algorithm. "Ours-original-C" denotes the centerline extraction result with the extracted road detection result produced by Our-S. (a) Visual comparisons between the modified methods and original methods with $\rho = 2$. (b) Visual comparisons with $\rho = 1$.

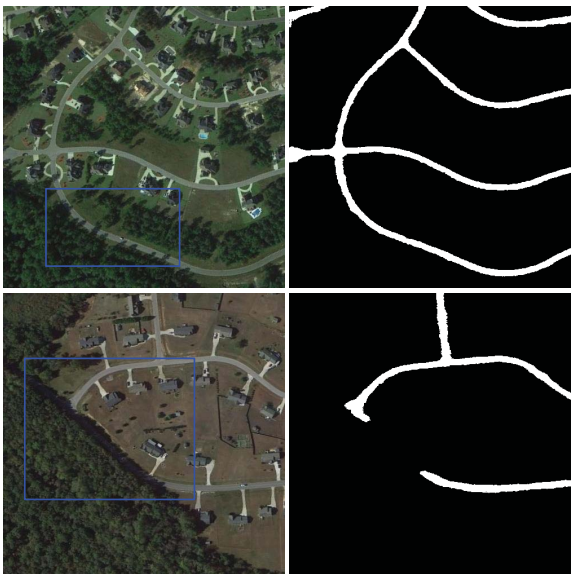


Fig. 9. Some road detection results of CasNet under the occlusions of trees.

G. Time Comparison

Table VI illustrates the comparison of average running time among different methods. It consists of the running time of road detection stage and centerline extraction stage. All the experiments are conducted at a single NVIDIA K20 GPU with 4-GB memory. It should be noted that our method is based on Caffe [48] and MATLAB, and other comparing methods are based on mixed programming of MATLAB and C++. As can be seen from Table VI, Miao takes less time than other comparing methods in the road detection stage, while ours is only about one third of Miao's method. Huang and Cheng take more time in this stage, because they are multiscale-based methods. In the centerline extraction stage,

TABLE VI

TIME COMPARISONS OF DIFFERENT METHODS. HERE, THE TIME IS MEASURED IN SECONDS. **DT** REFERS TO THE AVERAGE ROAD DETECTION TIME, AND **E**t REFERS TO THE AVERAGE CENTERLINE EXTRACTION TIME

	Image 10		Image 11		Image 12	
size	1316*738		999*692		1232*735	
	Dt (s)	**Et** (s)	**Dt** (s)	**Et** (s)	**Dt** (s)	**Et** (s)
Huang	60.43	0.03	28.32	0.02	60.95	0.03
Miao	5.42	71.09	3.49	57.28	4.71	58.86
Shi	19.91	72.18	14.35	58.14	16.38	60.83
Cheng	59.24	58.93	46.98	45.34	56.62	46.57
Ours	1.46	0.93	1.33	0.82	1.42	0.87

Huang takes the least time, while it produces spurs around the road centerline, which greatly reduce the smoothness and correctness of centerline network. Apart from Huang, other comparing algorithms cost more than 60 times running time of our extraction algorithm. Therefore, it demonstrates that CasNet achieves better performance by a large margin, as well as less running time than other state-of-the-art methods.

VII. CONCLUSION AND DISCUSSION

In this paper, a novel cascaded end-to-end convolutional neural network (CasNet) is proposed to perform the road detection task and road centerline extraction task simultaneously. Specifically, a covolutional neural network is introduced to extract consistent road area result. Based on the feature maps produced by the first network, a centerline extraction network is proposed to formulate the road centerline extraction task as two-class classification problem. After that, a thinning algorithm is utilized to obtain smooth, complete, and single-pixel width road centerline network. It should be noted that the CasNet is trained via an end-to-end strategy with two separated data sets, i.e., road detection data set and road centerline data set. Extensive experiments verify the advantages of CasNet: 1) in terms of both quantitative and visual performances, CasNet achieves extraordinarily more smooth and consistent road detection results than all the comparing methods; 2) CasNet achieves better performance than state-of-the-art methods in the road centerline extraction task; and 3) in terms of time complexity, CasNet is much faster than all the comparing methods. Specifically, it is about 25 times faster than the second fastest method (i.e., Huang). Moreover, as another contribution, a large and challenging road centerline extraction data set for VHR remote sensing images will be publicly available for further studies. It contains 224 original images, road segmentation reference maps, and their corresponding centerline reference maps. To the best of our knowledge, this is the biggest road centerline data set so far.

As Fig. 9 shows, to some extent, the CasNet shows robustness against the occlusions of trees (see the area of the blue rectangle in the first row of Fig. 9). However, for the continuous and large area of occlusions (see the area of the blue rectangle in the second row of Fig. 9), CasNet cannot detect the road well. We will leave it for the future work to incorporate more high-level semantic information to extract those occlusion areas well.

REFERENCES

[1] Q. Li, L. Chen, M. Li, S. Shaw, and A. Nuchter, "A sensor-fusion drivable-region and lane-detection system for autonomous vehicle navigation in challenging road scenarios," *IEEE Trans. Veh. Technol.*, vol. 63, no. 2, pp. 540–555, Feb. 2014.

[2] R. Bonnefon, J. Desachy, P. Dhérété, and J. Desachy, "Geographic information system updating using remote sensing images," *Pattern Recognit. Lett.*, vol. 23, no. 9, pp. 1073–1083, 2002.

[3] J. B. Mena, "State of the art on automatic road extraction for GIS update: A novel classification," *Pattern Recognit. Lett.*, vol. 24, no. 16, pp. 3037–3058, 2003.

[4] M.-F. A. Fortier, D. Ziou, C. Armenakis, and S. Wang, "Survey of work on road extraction in aerial and satellite images," *Tech. Rep.*, vol. 24, no. 16, pp. 3037–3058, 2003.

[5] 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.

[6] A. Kaur and E. R. Singh, "Various methods of road extraction from satellite images: A review," *Int. J. Res.*, vol. 2, no. 2, pp. 1025–1032, 2015.

[7] M. Song and D. Civco, "Road extraction using SVM and image segmentation," *Photogramm. Eng. Remote Sens.*, vol. 70, no. 12, pp. 1365–1371, 2004.

[8] Q. Zhang and I. Couloigner, "Benefit of the angular texture signature for the separation of parking lots and roads on high resolution multi-spectral imagery," *Pattern Recognit. Lett.*, vol. 27, no. 9, pp. 937–946, 2006.

[9] 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.

[10] 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. 3, pp. 583–587, May 2013.

[11] M. Wang and R. Li, "Segmentation of high spatial resolution remote sensing imagery based on hard-boundary constraint and two-stage merging," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 9, pp. 5712–5725, Sep. 2014.

[12] Y. Zhong, Q. Zhu, and L. Zhang, "Scene classification based on the multifeature fusion probabilistic topic model for high spatial resolution remote sensing imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 11, pp. 6207–6222, Nov. 2015.

[13] 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.

[14] 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.

[15] 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.

[16] D. Chaudhuri, N. K. Kushwaha, and A. Samal, "Semi-automated road detection from high resolution satellite images by directional morphological enhancement and segmentation techniques," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 5, no. 5, pp. 1538–1544, Oct. 2012.

[17] W. Shi, Z. Miao, Q. Wang, and H. Zhang, "Spectral–spatial classification and shape features for urban road centerline extraction," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 4, pp. 788–792, Apr. 2014.

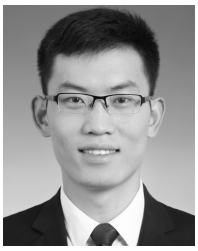
[18] W. Shi, Z. Miao, and J. Debayle, "An integrated method for urban main-road centerline extraction from optical remotely sensed imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 52, no. 6, pp. 3359–3372, Jun. 2014.

[19] A. Sironi, V. Lepetit, and P. Fua, "Multiscale centerline detection by learning a scale-space distance transform," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2697–2704.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Neural Inf. Process. Syst.*, 2012, pp. 1106–1114.

[21] 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.

[22] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Semantic image segmentation with deep convolutional nets and fully connected CRFs." Unpublished paper, 2014. [Online]. Available: https://arxiv.org/abs/1412.7062
[23] H. Noh, S. Hong, and B. Han, "Learning deconvolution network for semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1520–1528.
[24] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation." Unpublished paper, 2015. [Online]. Available: https://arxiv.org/abs/1511.00561
[25] L. Zhang, G. Xia, T. Wu, L. Lin, and X. Tai, "Deep learning for remote sensing image understanding," *J. Sensors*, vol. 2016, pp. 1–2, Jan. 2016.
[26] K. Nogueira, O. A. B. Penatti, and J. A. dos Santos, "Towards better exploiting convolutional neural networks for remote sensing scene classification," *Pattern Recognit.*, vol. 61, pp. 539–556, Jan. 2017.
[27] F. Zhang, B. Du, and L. Zhang, "Scene classification via a gradient boosting random convolutional network framework," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 3, pp. 1793–1802, Mar. 2016.
[28] S. Chen, H. Wang, F. Xu, and Y.-Q. Jin, "Target classification using the deep convolutional networks for sar images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4806–4817, Aug. 2016.
[29] F. Hu, G.-S. Xia, J. Hu, and L. Zhang, "Transferring deep convolutional neural networks for the scene classification of high-resolution remote sensing imagery," *Remote Sens.*, vol. 7, no. 11, pp. 14680–14707, 2015.
[30] F. Zhang, B. Du, L. Zhang, and M. Xu, "Weakly supervised learning based on coupled convolutional neural networks for aircraft detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 9, pp. 5553–5563, Sep. 2016.
[31] Y. Liu, Y. Zhong, F. Fei, and L. Zhang, "Scene semantic classification based on random-scale stretched convolutional neural network for high-spatial resolution remote sensing imagery," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2016, pp. 763–766.
[32] J. M. Alvarez, T. Gevers, Y. LeCun, and A. M. Lopez, "Road scene segmentation from a single image," in *Proc. 12th Eur. Conf. Comput. Vis.*, 2012, pp. 376–389.
[33] V. Mnih and G. E. Hinton, "Learning to detect roads in high-resolution aerial images," in *Proc. 11th Eur. Conf. Comput. Vis.*, 2010, pp. 210–223.
[34] 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.
[35] 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.
[36] P. Gamba, F. Dell'Acqua, and G. Lisini, "Improving urban road extraction in high-resolution images exploiting directional filtering, perceptual grouping, and simple topological concepts," *IEEE Geosci. Remote Sens. Lett.*, vol. 3, no. 3, pp. 387–391, Jul. 2006.
[37] C. Unsalan and B. Sirmacek, "Road network detection using probabilistic and graph theoretical methods," *IEEE Trans. Geosci. Remote Sens.*, vol. 50, no. 11, pp. 4441–4453, Nov. 2012.
[38] 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.
[39] A. Sironi, V. Lepetit, and P. Fua, "Projection onto the manifold of elongated structures for accurate extraction," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 316–324.
[40] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 886–893.
[41] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[42] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, 2015.
[43] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.
[44] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus, "Deconvolutional networks," in *Proc. 23rd IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2528–2535.
[45] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, Jan. 2014.
[46] O. Russakovsky *et al.*, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.
[47] J. Dai, K. He, and J. Sun, "Instance-aware semantic segmentation via multi-task network cascades," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3150–3158.
[48] Y. Jia *et al.*, "Caffe: Convolutional architecture for fast feature embedding," in *Proc. ACM Int. Conf. Multimedia*, 2014, pp. 675–678.
[49] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1026–1034.
[50] Google. (2015). *Google Earth*. [Online]. Available: http://www.google.cn/intl/zh-CN/earth/
[51] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors." Unpublished paper, 2012. [Online]. Available: https://arxiv.org/abs/1207.0580
[52] N. Navab, J. Hornegger, W. M. Wells III, and A. F. Frangi, "Medical image computing and computer-assisted intervention-(MICCAI)," in *Proc. 18th Int. Conf. (III)*, vol. 9351. Munich, Germany, Oct. 2015.
[53] C. Heipke, H. Mayer, C. Wiedemann, and O. Jamet, "Evaluation of automatic road extraction," in *Proc. Int. Arch. Photogram. Remote Sens.*, 1997, pp. 47–56.
[54] B. Wessel and C. Wiedemann, "Analysis of automatic road extraction results from airborne sar imagery," in *Proc. ISPRS Arch.*, 2003, pp. 105–110.



Guangliang Cheng received the B.S. degree from the China University of Petroleum, Qingdao, China, in 2012. He is currently pursuing the Ph.D. degree with the Institute of Automation, Chinese Academy of Sciences, Beijing, China.

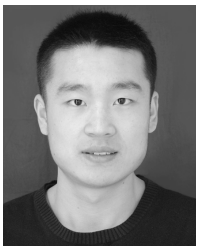
His research interests include pattern recognition, machine learning, remote sensing image processing, and deep learning.

Ying Wang received the B.S. degree from the Nanjing University of Information Science and Technology, China in 2005, the M.S. degree from the Nanjing University of Aeronautics and Astronautics, China, in 2008, and the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 2012.

He is currently an Associate Professor with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His research interests include computer vision, pattern recognition, and remote sensing.

Shibiao Xu received the B.S. degree in information engineering from the Beijing University of Posts and Telecommunications, Beijing, China, in 2009, and the Ph.D. degree in computer science from the Institute of Automation, Chinese Academy of Sciences, Beijing, in 2014.

He is currently an Assistant Professor with the Institute of Automation, Chinese Academy of Sciences. His current research interests include image-based 3-D scene reconstruction and scene semantic understanding.



Hongzhen Wang received the B.S. degree from the Ocean University of China, Qingdao, China, in 2013. He is currently pursuing the Ph.D. degree with the Institute of Automation, Chinese Academy of Sciences, Beijing, China.

His research interests include object detection, semantic segmentation, and deep learning.

Shiming Xiang (M'13) received the B.S. degree in mathematics, the M.S. degree from Chongqing University, Chongqing, China, in 1993 and 1996, respectively, and the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2004.

From 1996 to 2001, he was a Lecturer with the Huazhong University of Science and Technology, Wuhan, China. He was a Post-Doctoral Researcher with the Department of Automation, Tsinghua University, Beijing, China, until 2006. He is currently a Professor with the Institute of Automation, Chinese Academy of Sciences. His research interests include pattern recognition and image processing.

Chunhong Pan received the B.S. degree in automatic control from Tsinghua University, Beijing, China, in 1987, the M.S. degree from the Shanghai Institute of Optics and Fine Mechanics, Chinese Academy of Sciences, Beijing, in 1990, and the Ph.D. degree in pattern recognition and intelligent system from the Institute of Automation, Chinese Academy of Sciences, in 2000.

He is currently a Professor with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His research interests include computer vision, image processing, computer graphics, and remote sensing.