Simultaneous Road Surface and Centerline Extraction From Large-Scale Remote Sensing Images Using CNN-Based Segmentation and Tracing

Yao Wei, Kai Zhang, and Shunping Ji, *Member, IEEE*

***Abstract*— Accurate and up-to-date road maps are of great importance in a wide range of applications. Unfortunately, automatic road extraction from high-resolution remote sensing images remains challenging due to the occlusion of trees and buildings, discriminability of roads, and complex backgrounds. To address these problems, especially road connectivity and completeness, in this article, we introduce a novel deep learning-based multistage framework to accurately extract the road surface and road centerline simultaneously. Our framework consists of three steps: boosting segmentation, multiple starting points tracing, and fusion. The initial road surface segmentation is achieved with a fully convolutional network (FCN), after which another lighter FCN is applied several times to boost the accuracy and connectivity of the initial segmentation. In the multiple starting points tracing step, the starting points are automatically generated by extracting the road intersections of the segmentation results, which then are utilized to track consecutive and complete road networks through an iterative search strategy embedded in a convolutional neural network (CNN). The fusion step aggregates the semantic and topological information of road networks by combining the segmentation and tracing results to produce the final and refined road segmentation and centerline maps. We evaluated our method utilizing three data sets covering various road situations in more than 40 cities around the world. The results demonstrate the superior performance of our proposed framework. Specifically, our method's performance exceeded the other methods by 7% and 40% for the connectivity indicator for road surface segmentation and for the completeness indicator for centerline extraction, respectively.**

***Index Terms*— Convolutional neural network (CNN), remote sensing images, road extraction, segmentation, tracing.**

I. Introduction

ACCURATE and up-to-date road maps are of great importance in a wide range of applications, including urban planning, disaster management, vehicle navigation, and autonomous driving. Until now, time-consuming and labor-intensive manual work has been necessary to construct and update high-quality road networks. Thanks to the recent rapid development of Earth observation and remote sensing technology, though, considerable attention is being given to extracting roads automatically from high-resolution remote sensing images. Deep learning techniques, in particular, which have been successfully applied to image classification, semantic segmentation, object detection, and many other tasks in computer vision, offer a promising avenue for automatic road extraction from remote sensing images. However, the complex backgrounds of remote sensing imagery can cause the road extraction to suffer, such as overlapping of viaducts and occlusion of trees and tall buildings. Additionally, some land covers, such as bare soil, parking lots, and rivers, may share similar textures and structures with roads, making them difficult to discriminate. All these situations have prevented achieving automatic high-quality road extraction from remote sensing imagery.

During the past few decades, a variety of road extraction approaches have been advanced from different viewpoints, which can be generally divided into two categories, road surface segmentation and road centerline extraction. On the one hand, road surface segmentation mainly aims to produce a binary mask map where each pixel is labeled as either road or nonroad. On the other hand, road centerline extraction focuses on the topology and connectivity of road networks, which is commonly conducted by line tracking or thinning from road mask maps.

Conventional statistics and machine learning methods, such as artificial neural network (ANN), support vector machine (SVM), and maximum likelihood (ML), have been widely utilized in road surface extraction. For example, Kirthika and Mookambiga [1] applied ANN to extract road surfaces from satellite images using the texture and spectral information. Das *et al.* [2] proposed a multistage framework that includes four probabilistic SVMs and a series of postprocessing methods to extract roads from multispectral satellite images. Ünsalan and Sirmacek [3] proposed a framework to estimate road centerlines and a graph-based network to refine road segments. Wegner *et al.* [4] proposed a higher order conditional random field (CRF) [5] model to detect road network.

Recently, convolutional neural networks (CNNs) [6]–[9] have become prominent in semantic segmentation of visual

Manuscript received December 13, 2019; revised March 11, 2020; accepted April 26, 2020. Date of publication May 14, 2020; date of current version November 24, 2020. This work was supported by the National Key Research and Development Program of China under Grant 2018YFB0505003. *(Corresponding author: Shunping Ji.)*

Yao Wei and Shunping Ji are with the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China (e-mail: weiyao@whu.edu.cn; jishunping@whu.edu.cn).

Kai Zhang was with the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan 430079, China. He is now with ENSTA ParisTech, 91120 Palaiseau, France (e-mail: zhangkai11@whu.edu.cn).

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

Digital Object Identifier 10.1109/TGRS.2020.2991733

images. After Long *et al.* [10] proposed a fully convolutional network (FCN), which achieved pixel-wise segmentation by replacing fully connected layers with convolutional layers in a CNN, other FCN structures have been extensively studied, such as DeconvNet [11], SegNet [12], U-Net [13], and DeepLab [14], [15], for pursuing better semantic segmentation performance in close range and medical images.

Inspired by the successful applications of deep learning methods in semantic segmentation, some studies have introduced CNNs, especially FCNs, into road surface extraction. The network proposed by Mnih and Hinton [16] included millions of neurons to extract features representing roads. Zhong *et al.* [17] applied an FCN to extract roads and buildings from remote sensing images simultaneously. Panboonyuen *et al.* [18] presented a deep convolutional encoder–decoder network for road detection, followed by a CRF [5] to increase the spatial accuracy via filling gaps between road segments. Zhang *et al.* [19] combined a residual network with U-Net, which reduced the number of training parameters. He *et al.* [20] integrated the Atrous spatial pyramid pool (ASPP) [21] with the encoder–decoder network to extract fine features of roads. Yang *et al.* [22] designed a recurrent CNN (RCNN) unit to explore detailed low-level spatial characteristics. Zhou *et al.* [23] developed an encoder–decoder network named D-LinkNet, which consists of LinkNet [24] and dilated convolution [25]. Zhang and Wang [26] combined an efficient dense connection [27] with dilated convolution layers for a large receptive field.

As it is challenging to extract road centerlines directly from remote sensing images, most of the conventional centerline extraction methods to date [28], [29] are implemented by two steps: road surface detection and road centerline extraction. Huang and Zhang [30] presented a framework for road centerline extraction by integrating multiscale information with an SVM. An integrated method for urban main-road centerline extraction was introduced by Shi *et al.* [31], which incorporated spectral–spatial classification, local weighted regression, and tensor voting. Mattyus *et al.* [32] estimated road topology assisted by the initial segmentation results and inferred the missing connections based on the shortest path search algorithms. Cheng *et al.* [33] proposed cascaded networks to predict road surface and centerline segmentation maps; however, their centerline extraction was determined by the road segmentation. Lu *et al.* [34] presented a multitask learning framework which contained a road surface extraction network and a road centerline extraction network in parallel modes where both subtasks were limited in the pixel-wise segmentation level without considering any topological or structural information.

More recently, a few CNN-based studies have been introduced in road centerline tracing from remote sensing images, which were proved to significantly improve the accuracy. Ventura *et al.* [35] designed a CNN that predicted local connectivity between the central pixel and the border points of an input image patch and inferred the global topology of road networks by iterating this local connectivity. Bastani *et al.* [36] proposed a CNN-based iterative search method called RoadTracer to construct road network graphs from aerial images. Starting from a given point on the road, the decision was made between stepping back to the previous node and walking a fixed distance at an angle inferred by the CNN.

Although remarkable improvements have been made in road extraction by recent deep learning-based approaches, the problem is far from solved. Segmentation methods, such as the most recent D-LinkNet [23], can detect most of the road surfaces but produce poor connectivity results due to their neglect of the structural and topological information about the road. The situation is similar for two other methods [33], [34], which output road centerlines only through segmentation. On the other hand, tracing methods can preserve road connectivity better but may lack of completeness as the search is affected by the conditions of the starting and currently traced points. The most recent RoadTracer [36] only tracks a road network graph from one given starting point, which lacks automation and inevitably results in incompleteness in complicated scenes covering isolated roads. The problem to be addressed at this point, therefore, is how to combine the road features extracted by segmentation and tracing methods to constrain each other and benefit their complementary advantages. The novel framework we present in this article does indeed address this problem. Our approach is based on several CNNs and a fusion method to extract road surfaces and centerlines simultaneously to construct road networks with much better accuracy, connectivity, and completeness.

The main contributions of our work are summarized as follows.

1) A new multistage framework is proposed for simultaneous road surface and centerline extraction from remote sensing imagery, which aggregates both the semantic and topological information of road networks by combining the strengths of CNN-based segmentation and tracing. To our knowledge, it is the first integrated framework for simultaneous road surface segmentation and centerline tracing.
2) The boosting strategy is introduced to enhance the road segmentation results by applying multiple segmentation networks, which learn from the failed cases of previous segmentation incrementally to connect the broken segments in the initial masks. Moreover, a novel and light encoder–decoder structure is designed for boosting segmentation.
3) An improved iterative search algorithm guided by a CNN-based decision function is introduced to centerline tracing that starts tracing from multiple intersection points, which are automatically derived from the road segmentation maps predicted from the boosting segmentation, which was proved to improve both automation and completeness of centerline maps.
4) Finally, an empirical fusion method is introduced to produce the final refined road surface and centerlines through fusing the segmentation and centerline maps from the preceding steps.

The remainder of this article is arranged as follows. Section II introduces our proposed framework. In Section III, we illustrate our experimental results and evaluations on

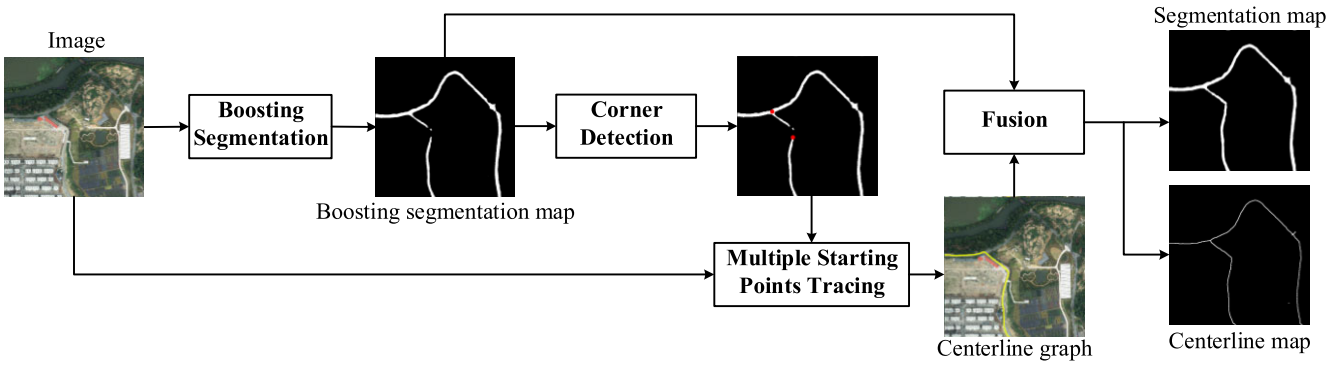


Fig. 1. Flowchart of the proposed framework for simultaneous road surface and centerline extraction.

high-resolution remote sensing images from dozens of cities over the world and compare our methods with the most current methods in road segmentation and centerline tracing. We further discuss the ramifications of our approach in Section IV and present our conclusions in Section V.

II. Multistage Framework for Road Surface and Centerline Extraction

Our proposed multistage framework for road surface and centerline extraction, which is illustrated by the flowchart in Fig. 1, has three main stages: 1) boosting segmentation; 2) multiple starting points tracing; and 3) fusion. First, the remote sensing image is initially segmented by a mainstream FCN method, which is followed by a series of boosting segmentation steps with another FCN we designed. Using the road mask maps predicted from the boosting segmentation, a corner detection method is applied to discover road intersections and other distinctive points. Second, a multiple starting points tracing is developed for tracing the topographical road centerline networks starting from the extracted road points. Finally, the results of road surface segmentation and centerline tracing are merged through a fusion process to produce fine segmentation and centerline maps. The implementation details are presented in Sections II-A–II-C.

A. Boosting Segmentation

In this article, we treat road surface extraction as an image semantic segmentation task that is realized through CNN-based initial and boosting segmentation. In our proposed method, D-LinkNet [23], which won the DeepGlobe 2018 Road Extraction Challenge [37], is applied to extract an initial and coarse road segmentation map. We found that many of the extraction process problems occur in the initial segmentation map, especially the discontinuities among road segments, which are caused by the utilized algorithm itself or by occlusion, which are tackled with a boosting strategy we developed. In an iterative manner, our boosting segmentation works as a mender to fill the gaps and to connect the fragmented road segments that existed in previous road segmentation maps.

Our boosting segmentation is motivated by AdaBoost [38], which concentrates on converting several weak classifiers into a strong one. The results of the strong classifier are generated by weighted voting from all the weak classifiers. By introducing and adapting the AdaBoost strategy to road segmentation refinement, a novel network named Boosting Segmentation Network (BSNet) is proposed here to work as a weak classifier in boosting segmentation.

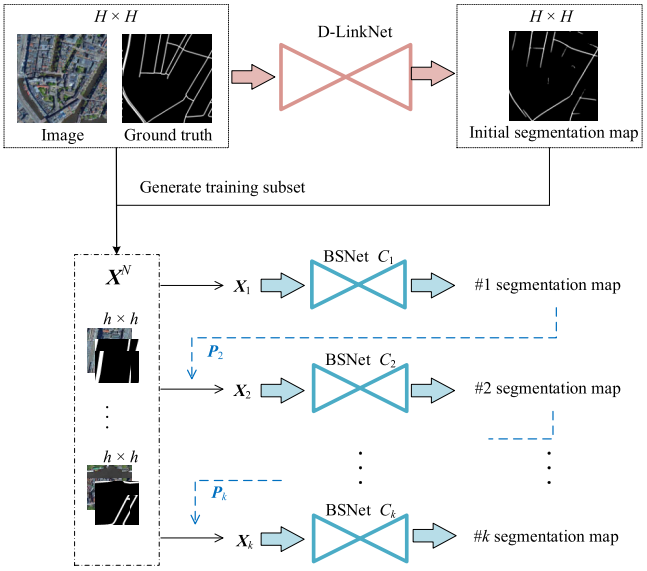


Fig. 2. Training procedure of boosting segmentation.

In Fig. 2, the initial segmentation is implemented with D-LinkNet. A subset (i.e., a set of patches ($h \times h$) cropped from the initial image ($H \times H$) and the corresponding ground truth) is produced as the input of the BSNet for refinement. We only crop out those patches whose intersection-over-union (IoU) between the initial segmentation map and the ground truth is lower than 0.7. Also, we discard the patches without roads to alleviate imbalance between the number of positive and negative samples. Finally, a subset $\boldsymbol{X}^N$ is obtained, where N represents the number of cropped patches (called data items in AdaBoost), and $\boldsymbol{X}_j$ represents the subset for training each BSNet C_j, where j is the number of iterations whose maximum is k.

As shown in Fig. 2, the first BSNet C_1 is fed with the $\boldsymbol{X}_1$ (i.e., $\boldsymbol{X}^N$). After training, it is evaluated on the $\boldsymbol{X}^N$. A series of

measures (using the following equations) are then calculated for obtaining the voting weight of the current model and updating the probability of each data item $x^i \in \boldsymbol{X}^N$ for the next iteration.

The probability of x^i is initialized by

$$P^i_{j=1} = \frac{1}{N}. \tag{1}$$

For BSNet C_j ($j > 1$), the training subset $\boldsymbol{X}_j$ is first generated through random resampling according to the probability P^i_j of each data item x^i, whereas the evaluation set is fixed with the whole $\boldsymbol{X}^N$. After model training and prediction on the evaluation set, the IoU of each data item x^i is calculated, where the error rate of x^i is defined as follows:

$$\varepsilon^i_j = 1 - \mathrm{IoU}^i_j. \tag{2}$$

The error rate of C_j is

$$\varepsilon_j = \sum_{i=1}^{N} \left(\varepsilon^i_j \times P^i_j\right). \tag{3}$$

The voting weight of C_j is defined as follows:

$$\omega_j = \log\left(\frac{1-\varepsilon_j}{\varepsilon_j}\right). \tag{4}$$

Then, the probability of x^i is updated for the next iteration as follows:

$$P^i_{j+1} = \begin{cases} P^i_j \times \dfrac{\varepsilon_j}{1-\varepsilon_j}, & \mathrm{IoU}^i_j > T_{\mathrm{IoU}} \\ P^i_j, & \text{otherwise} \end{cases} \tag{5}$$

where T_{IoU} is a predefined threshold which is set to 0.7.

A normalization step is followed,

$$P^i_{j+1} = \frac{P^i_{j+1}}{\sum_{i=1}^{N} P^i_{j+1}}. \tag{6}$$

The detailed boosting segmentation procedure is also given in Algorithm 1.

Algorithm 1 Boosting Segmentation

Input: training dataset $\boldsymbol{X}^N$ containing data item x with probability P and corresponding label y'. A predefined threshold T_{IoU}. The maximum number of BSNet is k and BSNet is noted as C.

Initialize:
  For $i \in [1, N]$
  $P^i_{j=1} = 1/N$
For $j = 1, 2, \ldots, k$
  Randomly select $\boldsymbol{X}_j$ from $\boldsymbol{X}^N$ according to P^i_j
  Training:
    Train C_j on dataset $\boldsymbol{X}_j$
  Evaluation:
    $y^i_j = C_j(x^i)$, $x^i \in \boldsymbol{X}^N$
    $IoU^i_j = \mathrm{IoU}\ (y^i_j, y'^i_j)$
    Error rate $\varepsilon^i_j = 1 - IoU^i_j$
    If $\varepsilon_j > 0.5$ **then** $k = j - 1$; stop
Updating:
    Foreach (x^i, y'^i) in $\boldsymbol{X}^N$
      Weight update $\beta_j = \varepsilon_j/(1-\varepsilon_j)$
    If $IoU^i_j > T_{\mathrm{IoU}}$ **then** $P^i_{j+1} = \beta_j P^i_j$
      Else $P^i_{j+1} = P^i_j$
    Normalize probability; $P^i_{j+1} = P^i_{j+1}/\sum_i P^i_{j+1}$
Inference:
Foreach x^i in test dataset:
$y^i = \sum_{j=1}^{k} \left(\log\frac{1}{\beta_j}\right) C_j(x^i)$

BSNet is implemented with a light and efficient segmentation network to learn how to fix the imperfect results of previous segmentation iteratively. As illustrated in Fig. 3, BSNet adopts an encoder–decoder architecture. The encoder part utilizes ResNet-34 [9] model pretrained on ImageNet [39] data set to accelerate the training procedure. It has four down-sampling layers with the input size of 256 × 256. The last three feature maps are down-sampled in the same size as half of the last feature map before they are concatenated, the process for which is denoted in Fig. 3 with blue arrows. The dilated 3 × 3 convolution layers with dilation rates of 1, 2, and 4, both in cascade and parallel modes and named the dilated block, are applied to enlarge the receptive field and to preserve the spatial information. As shown in Fig. 3, the receptive field of each path is different in order to combine the features from different scales. From top to bottom, the receptive fields are 15, 7, 3, and 1. Each convolution layer is followed by a ReLU activation except the last convolution layer which uses sigmoid activation. In order to reduce the number of computation parameters and to preserve the spatial correlation among the pixels, we replaced the conventional transposed convolution with the Data-dependent Upsampling (DUpsampling) [40] in the decoder part, which recovers the feature maps from the lowest resolution to the original scale.

The DUpsampling block is a project matrix for transforming the feature map to the final prediction result. As shown in Fig. 4, the input image size is $h \times w$, the sampling ratio is r, and the size of the last feature map F of the encoder is $h/r \times w/r \times Q$. A convolution layer with a 1 × 1 kernel is utilized to reshape it to $h/r \times w/r \times Q'$, where $Q' = r \times r$. For each 1 × 1 × Q' grid in the reshaped feature map F', it is reshaped to its corresponding spatial size $r \times r \times 1$. Finally, all the blocks consist of a feature map Y' equaling the size of the original input.

Instead of training along with the complete BSNet, the DUpsampling block was pretrained by minimizing the reconstruction loss L_{rec}. We defined W as a matrix to transform F to the final output Y' and Z as the inverse operator which transformed Y' to F. By denoting the ground truth as Y, the reconstruction loss can be defined as follows:

$$L_{\mathrm{rec}} = \sum_{Y} \|Y - Y'\|^2 = \sum_{Y} \|Y - WZY\|^2. \tag{7}$$

Once W was pretrained before each BSNet iteration and kept fixed, BSNet was trained and optimized with the loss function L_{seg}, consisting of the summation of a dice coefficient loss L_{dice} and a binary cross entropy (BCE) loss L_{BCE}, which can

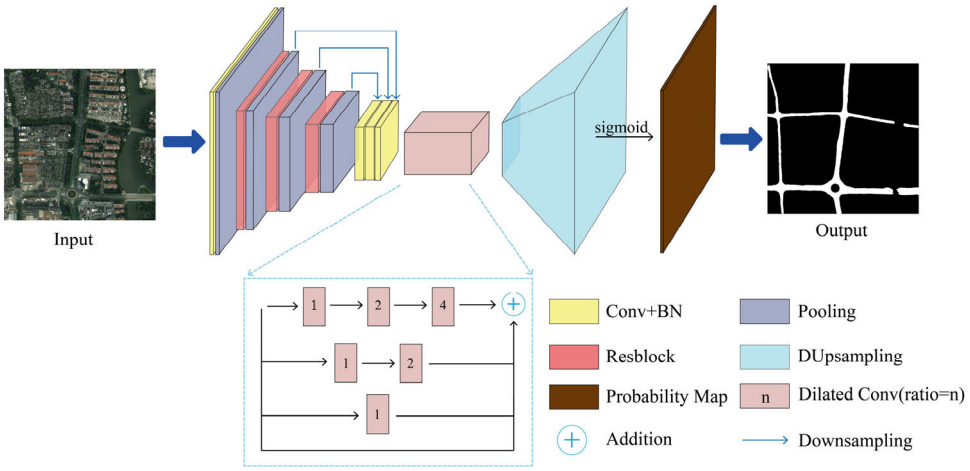


Fig. 3. Structure of the BSNet. Pooling indicates 2× max pooling.

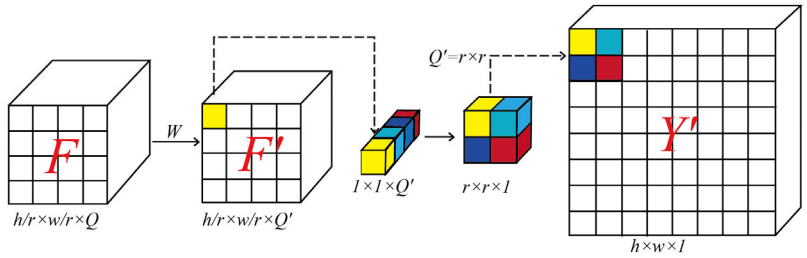


Fig. 4. DUpsampling block which upsamples the lowest feature map F at a size of $h/r \times w/r \times Q$ to the final segmentation map at a size of $h \times w \times 1$.

be defined as follows:

$$L_{\text{dice}} = 1 - \frac{\sum_{i=1}^{w}\sum_{j=1}^{h}|\text{Pred}_{ij} \cap \text{GT}_{ij}|}{\sum_{i=1}^{w}\sum_{j=1}^{h}(|\text{Pred}_{ij}| + |\text{GT}_{ij}|)} \tag{8}$$

$$L_{\text{BCE}} = -\sum_{i=1}^{w}\sum_{j=1}^{h}|\text{GT}_{ij} \times \log \text{Pred}_{ij} + (1 - \text{GT}_{ij}) \times \log(1 - \text{Pred}_{ij})| \tag{9}$$

$$L_{\text{seg}} = L_{\text{dice}} + L_{\text{BCE}} \tag{10}$$

where Pred is the prediction, GT is the ground truth, and w and h represent the width and height of the image, respectively.

The weighted result of these BSNets, which is called boosted segmentation map, R_{boost}, is obtained by

$$R_{\text{boost}} = \frac{\sum_{j=1}^{k}\omega_j R_j}{\sum_{j=1}^{k}\omega_j} \tag{11}$$

where k is the total number of BSNets, and ω_j and R_j constitute the voting weight calculated by (4) and the segmentation probability map of the jth BSNet C_j, respectively.

An integrated strategy which takes advantage of probability information from both the initial segmentation map and the boosted segmentation map is introduced to obtain the boosting segmentation results. Specifically, the initial segmentation map, R_{ini}, is formulated as a base map, and the boosted segmentation map, R_{boost}, is added into the base map at the pixels in case the sum of them is larger than a threshold T_{seg}, resulting in probability segmentation map, R_{seg}. The algorithm can be described as follows:

$$R_{\text{seg}} = \text{Norm}(R_{\text{ini}} + R_{\text{boost}}[(R_{\text{ini}} + R_{\text{boost}}) > T_{\text{seg}}]) \tag{12}$$

where $R\,[f]$ is an operator on the segmentation map R and if f is TRUE at a pixel in R, the corresponding pixel value remains unchanged, otherwise it is set to zero; and Norm $(\cdot)$ is the normalization step to obtain the probability segmentation map ranging from 0 to 1.

Finally, the probability map is binarized through the OTSU [41] algorithm, which identifies the threshold automatically to classify the foreground and background by maximizing the separability of the categories in gray levels.

B. Multiple Starting Points Tracing

Road centerline tracing aims to reconstruct the global topology of road networks. Our algorithm, which is called the multiple starting points tracer (MSP-Tracer), was developed from a baseline method called RoadTracer [36], which searches road centerlines starting from a known point on a road and constructs road networks iteratively. The key component is a CNN-based decision function. At each step of tracing, the CNN is utilized to decide either to walk a fixed distance at an angle or stop and step back to the previous vertex in the search tree. Then, the searching window centered on the current point is updated and conveyed to the network for the next prediction. Vertices (points on roads) and edges (line segments connecting adjacent points) are added to a path list as

the search proceeds, and the road network graph is constructed until all the points in a searching stack are explored.

Our algorithm attempts to fix the two distinct drawbacks of RoadTracer. The first drawback is that the starting point is manually determined, which may be not applicable in real applications and lowers the automation of the algorithm. Its second drawback is that starting the search from a single point easily can be affected by obstacles and may omit many other roads as well, especially in large-scale remote sensing images. To overcome these problems, our MSP-Tracer algorithm traces the centerlines from multiple starting points, which are automatically generated based on previous road segmentation results rather than user interaction.

A corner detector called Good Features To Track [42] was applied to detect the road junctions as starting points for tracing. Before applying the detector on the segmentation map, we skeletonized the road segmentation mask to a one-pixel width. The scoring function of the Good Features To Track operator is defined as follows:

$$R = \min(\lambda_1, \lambda_2) \tag{13}$$

where λ_1 and λ_2 are eigen values of matrix M (14), which is a weighted covariance matrix with I representing the gradient image:

$$M = \sum_{x,y} w(x, y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix}. \tag{14}$$

After the multiple starting points were extracted, a CNN-based decision function (decision_func) was applied to tracing the road centerlines. The network and training process are similar to the RoadTracer [36]. As illustrated in Fig. 5, the input layer is a $d \times d$ sliding window centered on the current point S_{top} in a stack and consists of the red, green and blue (RGB) channels of the image patch, the currently constructing graph G, and the ground truth road graph G^*, which was only used in training and replaced with a blank graph during inference. The output layer consists of two components: an action component that decides either walk or stop, $O_{\text{action}} =< O_{\text{walk}}, O_{\text{stop}} >$; and an angle component that decides which angle to walk toward, O_{angle}. The network also outputs an intermediate segmentation result, O_{seg}, which is a coarse thumbnail of the segmentation map to constrain the training process. Therefore, the training process is optimized by the square loss of angles and actions as well as the cross-entropy loss between the predicted thumbnail and the ground truth.

In the inference stage, the MSP-Tracer is conducted by tracking the road network from each point in a starting points list. First, one point is randomly chosen from the starting points list and the MSP-Tracer starts tracing from this point by following the output of the pretrained CNN-based decision function step-by-step until the searching stops. Then, another point is chosen as the next starting point from the list and is pushed into the searching stack. Considering the computing expense, we included an adaptive starting point decision (ASPD) strategy which dynamically chooses the next starting point according to the earlier explored graph. Specifically, the starting point of the following tracing is determined by checking whether the earlier explored graph is outside a bounding box centered at a candidate starting point, which is removed from the list if the bounding box intersects with the previous constructed graph. The MSP-Tracer terminates when all the starting points are explored. A graph that records the nodes and edges is obtained by tracing these centerlines. The inference procedure is also given in Algorithm 2.

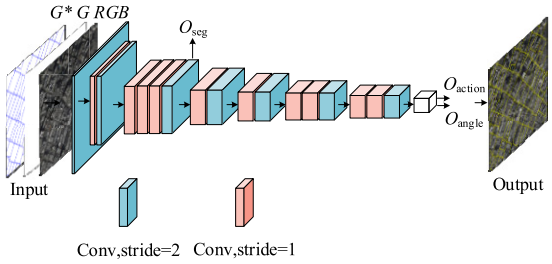


Fig. 5. Architecture of CNN-based decision function. It outputs an action component O_{action} and an angle component O_{angle}; O_{seg} is a coarse thumbnail of the segmentation map to constrain the training.

Algorithm 2 Multiple Starting Points Tracing (MSP-Tracer)

Input: starting points list V, window W, graph $G = \Phi$, vertex stack $S = \Phi$, move distance D, bounding box B.
while V is not empty, **do**
  randomly pick V_i from V
  $S = V_i$
  initialize W_i centered at V_i
  if G intersect with W_i; **break**
  else
    while S is not empty, do
      action, α = decision_func (G, S_{top}, Image)
      $u = S_{\text{top}} + (D \cos\alpha,\ D \sin\alpha)$
      if action == stop or u is outside B then
        pop S_{top} from S
      else
      add vertex u to G
      add an edge (S_{top}, u) to G
      push u onto S
      end if
    end while
  end if
  remove V_i from V
return G

C. Fusion

Pixel-wise road segmentation tends to produce many isolated and discontinuous road segments due to ignoring the structural and topological information of roads and the occlusions from backgrounds, whereas centerline graphs derived from tracing approaches are influenced by the locations of the starting points. There may be no starting point assigned on an isolated road, and our MSP-Tracer greatly advances the original single start point RoadTracer. We attempted to combine the road features extracted by the above two approaches to constrain each other and to benefit their complementary

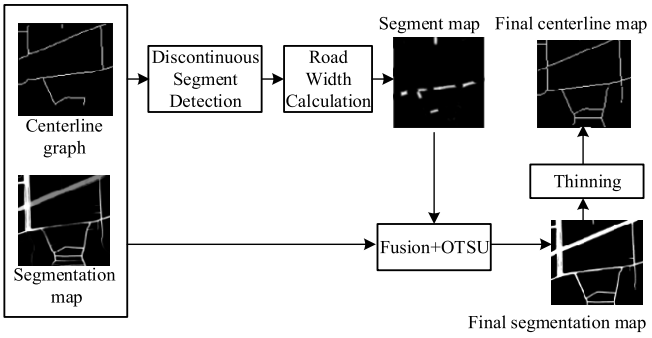


Fig. 6. Process of the fusion method.

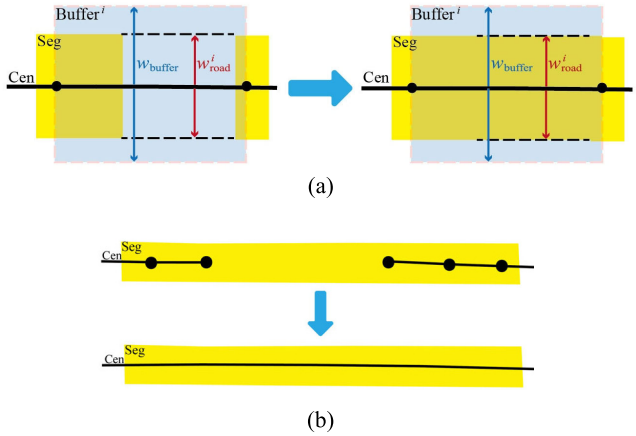


Fig. 7. Two examples to show the advantages of the fusion method. (a) Connectivity of segmentation map (yellow) can be enhanced by the buffered centerline (light blue). (b) Completeness of centerline graph (black lines) can be improved by segmentation map.

advantages. Our proposed fusion method aggregates both the semantic and topological information of road networks. The process is shown in Fig. 6.

The first step in our fusion method is to detect the discontinuous segments in the boosting segmentation map with the help of the centerline graph inferred from the MSP-Tracer. The centerline graph that records the nodes and edges is split into multiple segments with fixed lengths, and it judges, for each segment, whether there is an intersection with the binary segmentation map. The segment is considered as "discontinuous" if it is partially covered by the segmentation map.

In the second step, a centerline map is generated with a certain road width at the discontinuous segments. The optimal road width is automatically inferred; and a buffer with an empirical width w_{buffer} is created for each discontinuous segment i. The intersection of the buffer area and the segmentation map, denoted as area^i, is obtained, and the road width, w^i_{road}, of segment i then is calculated as follows:

$$w^i_{\text{road}} = \frac{\text{area}^i}{\text{length}^i} = \frac{\text{num}(\text{Seg} \cap \text{Buffer}^i)}{\text{num}(\text{Seg} \cap \text{Cen}^i)} \tag{15}$$

where Seg refers to the segmentation map, Buffer^i refers to the buffer area of segment i on the centerline graph, Cen^i is the single-pixel width segment i on the centerline graph, and num $(\cdot)$ counts the number of road pixels in the intersection area. Then, the centerline graph is rasterized to obtain the centerline segment map by expanding the inferred width. Note that only those "discontinuous segments" need to be expanded.

The third step is the fusion of the centerline segment map, denoted as R_{cen}, and the probabilistic segmentation map R_{seg} derived from boosting segmentation to an integrated probability map R_{fuse}. The map R_{fuse} is obtained under the rule

$$\begin{aligned} R_{\text{fuse}} = {} & R_{\text{seg}}[(R_{\text{seg}} + R_{\text{cen}}) \le (T_{\text{fuse}} + 1)] \\ & + R_{\text{cen}}[(R_{\text{seg}} + R_{\text{cen}}) > (T_{\text{fuse}} + 1)] \end{aligned} \tag{16}$$

where $R\,[f]$ is an operator on the map R. At each pixel in R, if f is TRUE, the pixel value remains unchanged; otherwise, the pixel value is set to 0. T_{fuse} equals to the threshold of the OTSU algorithm for binarizing the segmentation map R_{seg}, which is adaptive to each input image.

The last step generates the final road segmentation and centerline map from R_{fuse}. R_{fuse} is binarized with the OTSU algorithm [41], which discovers the optimal threshold automatically, to produce a binary map (i.e., the final segmentation map). Morphological thinning algorithm [43] is performed on the segmentation map to obtain the final centerline map.

Two obvious advantages of our fusion method, which fuses the results from boosting segmentation and centerline tracing, are described in Fig. 7. First, the discontinuous segments on the segmentation map can be connected through the topology information provided by centerline graph; and as shown in Fig. 7(a), the segment i on the centerline graph, with road width w^i_{road} calculated by (15), helps connect the two separate road surfaces (yellow). Second, the segmentation map helps connect the broken centerlines or create a centerline in isolated roads. As shown in Fig. 7(b), the two centerlines are correctly bridged by (16) and the following thinning algorithm.

Note that in the fusion step, we only process the partially overlapped centerlines and segmentation maps. If they are fully overlapped, both are inferred well, but if they do not overlap, the segmentation map will probably create centerlines according to (16).

III. Experiment and Results

A. Data Sets

We performed our experiments on three diverse data sets: 1) the Massachusetts data set [44]; 2) the Shaoxing data set; and 3) the Cities data set [36].

The Massachusetts data set, which is publicly available, contains aerial images with at least 1500 × 1500 pixel size and 1-m resolution together with the corresponding 7-pixel-width segmentation ground truth collected from OpenStreetMap (OSM) [45] that covers about 2600 km^2 in Massachusetts. After manually excluding some damaged images, we preprocessed the data set by merging and cropping it into 256 image tiles of 1024 × 1024 pixels. Among them, 192 images were separated for training and the remaining 64 images were used for testing. Corresponding centerline

ground truths were obtained by skeletonizing the pixel level segmentation annotations.

The Shaoxing data set was captured from Shaoxing City, which is a watery city in China with many lakes and rivers, making it challenging to extract complete and accurate road networks. There are 532 1024 × 1024 aerial images and corresponding segmentation annotations with variable road widths that are close to the real width. The corresponding ground resolution of a pixel is 0.6 m. A total of 372 images were used for training and 160 for testing. We obtained the centerline ground truth by skeletonizing the pixel level segmentation annotations.

In terms of the Cities data set, we collected satellite images from Google Earth [46] with 60 cm/pixel resolution, and the centerline ground truth was obtained from OSM covering the urban core of 37 cities across six countries. For each city, the centerline graph covered a region of approximately 24 km^2 around the city center. The data set was divided into a training set of 25 cities and a test set of 12 other cities. Those images also were cropped into 1024 × 1024 tiles, resulting in 1600 images for training and 768 images for testing. The ground truth for road segmentation was obtained by rasterizing the road centerline with a constant width of 8 pixels.

B. Evaluation Metrics

To assess the quantitative performance in both road surface and centerline extraction, seven benchmark metrics were introduced. Four metrics were employed in the road surface segmentation evaluation and the other three metrics were utilized in the road centerline extraction evaluation.

1) Road Segmentation Metrics: Recall, precision, and IoU, which are commonly used as the evaluation indicator for semantic segmentation, were the metrics adopted to estimate the segmentation accuracy at the pixel level. IoU, which is an overall metric offering a tradeoff between recall and precision, refers to the ratio between the intersection of the road pixels predicted by the algorithm and the true-positive pixels and the result of their union.

The equal-width road mask generated from the OSM data in both the Massachusetts and Cities data sets adversely affected the pixel-based metrics as road width varies. Thus, we used relaxed metrics based on the "buffer method" suggested by Mnih and Hinton [47]. The relaxed recall and precision introduced a buffer. Within the range of φ pixels from any positively labeled pixel of the ground truth, each pixel predicted as positive was considered correctly classified.

Another critical issue in road segmentation is the connectivity of the roads. It makes little sense to obtain only isolated road segments. Therefore, we provided a metric called connectivity (Conn), which reflects the connectivity and topology of road networks at the local scale. Specifically, the ground truth centerline graph is split in multiple segments of equal length, and the segment totally covered by the predicted segmentation map is considered as the connected road segments. The calculation of the connectivity is as follows:

$$\text{Conn} = \frac{2N_{\text{conn}}}{N_{\text{gt}} + N_{\text{pred}}} \tag{17}$$

where N_{conn} is the number of connected segments, and N_{gt} and N_{pred} are the total number of segments on the ground truth graph and skeletonized prediction graph, respectively.

2) Centerline Extraction Metrics: Completeness, correctness, and quality, which were introduced by Wiedemann *et al.* [48], were utilized to assess the performance of the centerline extraction algorithms. Completeness (Comp) is a variant of recall, which is the percentage of the reference road centerline that lies within a buffer of width ρ around the extracted centerlines, and correctness (Corr) is a variant of precision and is the percentage of the extracted road centerlines that lies within a buffer of width ρ around the reference centerlines. Quality (Qual) is an overall metric which combines Comp and Corr. They are described as follows:

$$\text{Comp} = \frac{\text{length of matched reference}}{\text{length of reference}} \tag{18}$$

$$\text{Corr} = \frac{\text{length of matched extraction}}{\text{length of extraction}} \tag{19}$$

$$\text{Qual} = \frac{\text{length of matched extraction}}{\text{length of extraction+length of unmatched reference}}. \tag{20}$$

C. Implementation Details

1) Boosting Segmentation: For the initial segmentation, we implement data augmentation, which includes image horizontal flip, vertical flip, diagonal flip, color jittering, shifting, and scaling to expand the data set size. The weights of the BCE loss and dice loss are equal. The Adam optimizer [49] was selected as the network optimizer. The learning rate is initially set at 2e-4 and divided by 5 while the training loss stops decreasing up to three continuous epochs. The batch size during the training phase is fixed as two on 1024 × 1024 tiles. We use test time augmentation (TTA) in prediction, which includes image horizontal flip, vertical flip, and diagonal flip (predicting each image $2 \times 2 \times 2 = 8$ times) also on 1024 × 1024 tiles, and then produced an initial segmentation map in both the binary and probability formats. To obtain a boosted segmentation map, we use two BSNets in the boosting segmentation. The training subsets for BSNets consisted of 256 × 256 patches which are seamlessly cropped from the initial segmentation binary mask. The IoU of each patch then is calculated; and only when the IoU is lower than 0.7 will the patch be chosen as the training subset. For example, in the Massachusetts data set, 470 image patches of 256 × 256 pixels were chosen from 256 images of 1024 × 1024 pixels and conveyed to the BSNet as a training subset. Data augmentation is applied in BSNet. The DUpsampling ratio is 16 and the lowest resolution of the feature map (i.e., the output feature map from the ResNet-34) is 16 × 16. The learning rate is 5e-3. The previous ten epochs are taken as a warm-up and after that, the learning rate is updated each epoch polynomial. The total training epoch is set at 200, but the network will terminate early if it stops decreasing in six continuous epochs. TTA is utilized in testing phase as well. To obtain the aggregation

of the initial and boosted segmentation map, we summed the probability map and compared it with T_{seg}, which is set at 1.

2) Multiple Starting Points Tracing: A larger image preserves better road completeness and connectivity. Before tracing, we detect corners from 8192 × 8192 binary maps, which are merged from 1024 × 1024 masks. For each merged map, we generated a maximum of 100 points with a minimum distance of 400 pixels for the adjacent corners. For road centerline tracing, we set the search window size $d = 256$ pixels. In the ASPD algorithm, we set the radius of the search bounding box as 60 pixels, which is three times that of each road segment length. The batch size was set at four and the loss function includes three parts with equal weight, action loss, angle loss, and cross-entropy loss between the predicted thumbnail and the ground truth. We used the Adam optimizer and trained about 400 epochs. The initial learning rate was 1e-5 and was updated every 100 epochs. Similar to RoadTracer [36], the output angle contained 64 evenly distributed directions and the angle with the maximum probability was selected as the moving direction; a threshold was set at 0.4 for action output, which meant that if O_{walk} was above the threshold, then the point walked 20 pixels at each step. Otherwise, it stopped and stepped back to the previous node.

3) Fusion: The buffer width w_{buffer} was set at 11 pixels in order to make the buffer a little wider than the road on the segmentation map.

All the algorithms were implemented based on PyTorch [50], and the experiments were conducted at a NVIDIA GTX1060 with 6-GB memory.

D. Comparison of Road Segmentation Methods

We compared our proposed algorithms with U-Net [13], ResUnet [19], ASPP-Unet [20], RCNN-Unet [22], and LinkNet [24], as well as D-LinkNet [23]. The relaxed recall, precision, IoU, and Conn of each method were computed. The buffer width φ was set at 4 pixels. Please note that the "buffer width" is the distance between the edges and the centerline (i.e., a half width of the buffer) as defined in many related studies.

Our results for different segmentation methods on different data sets are presented in Table I. It can be seen first that, compared with U-Net, ResUnet, ASPP-Unet, RCNN-Unet and LinkNet, D-LinkNet achieved the best performance in both IoU and Conn. Second, both our boosting segmentation and fusion methods exceeded D-LinkNet considerably. Taking the results of D-LinkNet as a baseline, the IoU improved 1.6% and Conn improved 3.0% when using our boosting segmentation on the Massachusetts data set. When our fusion method was used, IoU improved 2.0% and Conn improved 4.3%. On the Shaoxing data set, the results for boosting segmentation increased by 0.7% in IoU and 1.0% in Conn compared to D-LinkNet. Our fusion method outperformed D-LinkNet by 2.5% in Conn. In terms of the Cities data set, where prediction was performed on 12 cities around the world, the IoU and Conn of our boosting segmentation were 2.4% and 6.3% higher than the baseline; the IoU and Conn of fusion method were 2.7% and 7.8% higher than the baseline, respectively. The improvement for IoU was not as significant, but Conn greatly improved after we introduced the boosting strategy and the fusion with the centerline graphs. This is critical progress as connectivity is the key indicator toward achieving automation of road extraction and a better indicator than IoU for progress toward semiautomatic road extraction, where manual work is mainly spent on fixing holes and breaks between extracted road segments.

In Fig. 8, we show our qualitative comparison of different road segmentation methods on different data sets. There are six rows and ten columns of subfigures. It can be seen from the third column to the seventh column that U-Net, ResUnet, ASPP-Unet, RCNN-Unet, and LinkNet had difficulties distinguishing the homogenous regions from the real road regions. Among them, ResUnet performed the poorest on the Cities data set. As shown in the eighth column, D-LinkNet eliminated most of the false positives and false negatives but still experienced discontinuities due to the shadows caused by trees and buildings. In contrast, our boosting segmentation achieved more coherent road areas and much smoother road boundaries than D-LinkNet, which demonstrates that our boosting segmentation is more robust against occlusions. Moreover, after integrating the centerline results by the MSP-Tracer, our fusion method further improved the connectivity. In terms of the Shaoxing data set, illustrated in the third and fourth rows, our fusion strategy was able to connect the gaps and obtained more structured road networks.

E. Comparison of Centerline Extraction Methods

Table II contains the quantitative comparison of the different road centerline extraction methods on different data sets for completeness (Comp), correctness (Corr), and quality (Qual). The buffer width ρ was set at 4 pixels. Compared with RoadTracer, our results after applying multiple starting points (MSP-Tracer) significantly improved the topology. The Comp, Corr, and Qual improved 21.0%, 19.5%, and 14.8%, respectively, on the Shaoxing data set. The improvement on the other data sets was relatively small as they had less isolated roads.

Our fusion strategy radically outperformed RoadTracer. For example, the Qual of our fusion method exceeded that of RoadTracer by 48.8%, 50.0%, and 37.1% for the Massachusetts data set, Shaoxing data set, and Cities data set, respectively. The Comp and Corr scores also showed significant improvement.

Fig. 9 shows a visual comparison of different road centerline extraction methods on the Massachusetts data set, Shaoxing data set, and Cities data set. For better visualization, the images have been cropped. As illustrated in the first column, RoadTracer performed well on road connectivity but produced incomplete road networks from large-scale remote sensing images due to the limitation of a single starting point. By comparing the first and second columns, it can be clearly seen that our MSP-Tracer achieved more complete road networks than RoadTracer and efficiently eliminated the viaduct and river blocking problems. Considering the complementary character-

TABLE I

ROAD SEGMENTATION RESULTS, WHERE THE VALUES IN BOLD ARE WITH THE BEST PERFORMANCE

,

Dataset	Method	Recall	Precision	IoU	Conn
Massachusetts	U-Net [13]	0.7518	0.7738	0.6735	0.6245
	ResUnet [19]	0.7484	0.8292	0.6998	0.6164
	ASPP-Unet [20]	0.7655	0.8011	0.7061	0.7015
	RCNN-Unet [22]	0.7786	0.7814	0.7067	0.7009
	LinkNet [24]	0.7697	**0.8451**	0.7311	0.6784
	D-LinkNet [23] (Baseline)	0.8121	0.8267	0.7662	0.7810
	Boosting Segmentation (Ours)	0.8455	0.7992	0.7821	0.8110
	Fusion (Ours)	**0.8588**	0.7847	**0.7865**	**0.8238**
Shaoxing	U-Net [13]	0.5677	0.7007	0.4563	0.4276
	ResUnet [19]	0.6173	0.6494	0.4666	0.2992
	ASPP-Unet [20]	0.6869	0.6532	0.5087	0.5028
	RCNN-Unet [22]	0.6484	0.6870	0.5025	0.5002
	LinkNet [24]	0.6428	0.7882	0.5497	0.4961
	D-LinkNet [23] (Baseline)	0.7253	**0.8055**	0.6144	0.6411
	Boosting Segmentation (Ours)	**0.7798**	0.7571	**0.6218**	0.6515
	Fusion (Ours)	0.7775	0.7519	0.6178	**0.6663**
Cities	U-Net [13]	0.4674	0.4961	0.3384	0.2450
	ResUnet [19]	0.2979	0.6413	0.2615	0.1366
	ASPP-Unet [20]	0.5514	0.5917	0.4283	0.3858
	RCNN-Unet [22]	0.5560	0.5925	0.4325	0.3908
	LinkNet [24]	0.4638	0.6514	0.3926	0.3374
	D-LinkNet [23] (Baseline)	0.5998	**0.6708**	0.4981	0.4769
	Boosting Segmentation (Ours)	0.6794	0.6241	0.5222	0.5394
	Fusion (Ours)	**0.6907**	0.6173	**0.5247**	**0.5545**

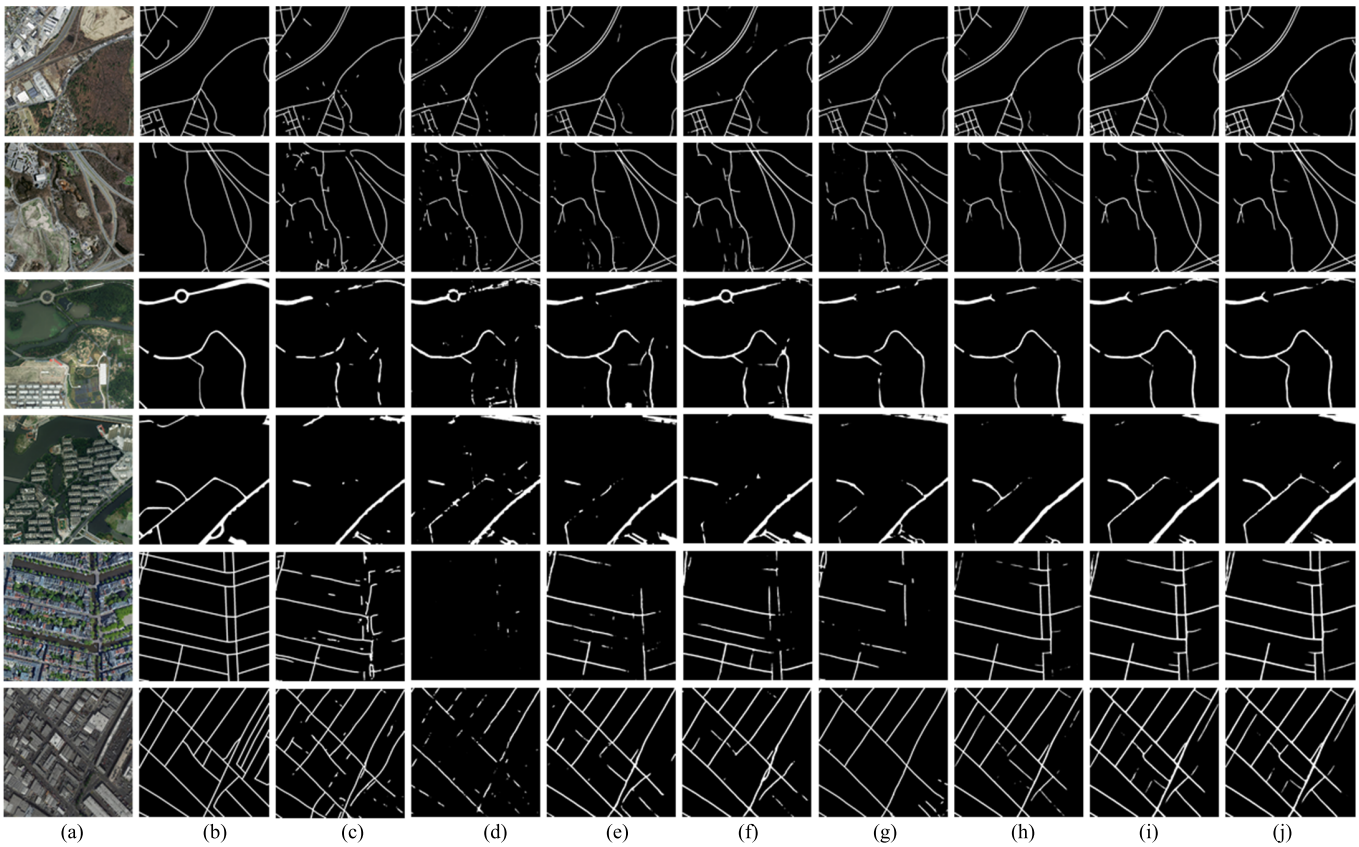


Fig. 8. Qualitative results of different road segmentation methods on different data sets. (From top to bottom) Every two consecutive rows represent the performance for the Massachusetts data set, Shaoxing data set, and Cities data set. (a) Image. (b) Ground truth. (c) U-Net. (d) ResUnet. (e) ASPP-Unet. (f) RCNN-Unet. (g) LinkNet. (h) D-LinkNet. (i) Boosting Segmentation. (j) Fusion.

istics between road segmentation and centerline tracing, our fusion method extracted a more accurate road network and performed much better as far as completeness. For example, in the left-bottom corner of the bottom image (from the Cities data set), several roads were totally missed by the MSP-Tracer but were well repaired by the segmentation results.

Fig. 9. Visual results of different road centerline extraction methods on the (top) Massachusetts data set, (middle) Shaoxing data set, and (bottom) Cities data set. We overlay the inferred graph (yellow) over ground truth from OSM data (light blue). (a) RoadTracer. (b) MSP-Tracer. (c) Fusion.

TABLE II

ROAD CENTERLINE RESULTS, WHERE THE VALUES IN BOLD ARE WITH THE BEST PERFORMANCE

Dataset	Method	Comp	Corr	Qual
Massachusetts	RoadTracer [36]	0.435	0.513	0.308
	MSP-Tracer (Ours)	0.488	0.552	0.343
	Fusion (Ours)	**0.889**	**0.882**	**0.796**
Shaoxing	RoadTracer [36]	0.168	0.413	0.150
	MSP-Tracer (Ours)	0.378	0.607	0.298
	Fusion (Ours)	**0.823**	**0.757**	**0.650**
Cities	RoadTracer [36]	0.229	0.371	0.170
	MSP-Tracer (Ours)	0.264	0.416	0.195
	Fusion (Ours)	**0.701**	**0.693**	**0.541**

TABLE III

RESULTS OF BOOSTING SEGMENTATION WITH DIFFERENT NUMBERS OF BSNETS ON THE CITIES DATA SET

Dataset	k	Recall	Precision	IoU	Conn
Cities	0	0.5998	**0.6708**	0.4981	0.4769
	1	0.6726	0.6307	0.5215	0.5391
	2	**0.6794**	0.6241	**0.5222**	**0.5437**
	3	0.6736	0.6286	0.5212	0.5412
	4	0.6741	0.6295	0.5218	0.5411
	5	0.6738	0.6298	0.5218	0.5410

IV. DISCUSSION

In this section, we focus on the impacts of a few tunable parameters (the number of BSNets and the buffer width of the centerline) in our framework for simultaneous road surface and centerline extraction. In addition, the influence of the sample quality on the road extraction is discussed.

Our boosting segmentation includes a series of BSNets. When the number of BSNet was 2, the segmentation model outperformed the other road segmentation methods on different data sets quantitatively (Table I) and qualitatively (Fig. 8). Here, we evaluate the effects of the different number of BSNets on the performance of road surface segmentation for the Cities data set. The comparison is described in Table III, where the best performance is denoted in bold type and the second best is underlined. Compared with D-LinkNet ($k = 0$), the first boosting segmentation ($k = 1$) obtained 2.3% IoU and 6.3% Conn improvement. When k was increased, the performance of the boosting segmentation changed only slightly, and the highest IoU and Conn were reached when $k = 2$. The results proved that the performance of the model was robust to parameter k.

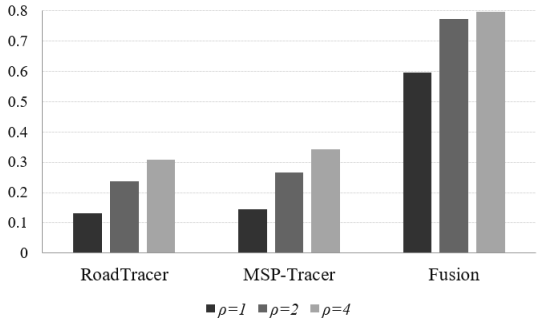


Fig. 10. Results of different centerline extraction methods with different buffer widths, $\rho = 1, 2, 4$, on the Massachusetts data set.

We introduced a buffer-based evaluation for the road centerlines as it was difficult to directly compare the pixel difference between the extracted centerline and the ground truth. We discussed earlier the influence of different buffer widths ρ on the performance of centerline extraction methods. Fig. 10 shows the performances of the different centerline extraction methods with different buffer widths (1, 2, and 4 pixels) on the Massachusetts data set. Our results indicate that as the buffer width grew, the improvement increased. Our fusion method performed better than RoadTracer [36] and MSP-Tracer on Qual and achieved a 40% improvement on Qual compared to RoadTracer. Also, when buffer width ρ changed from 2 to 4, the performance of the fusion slightly improved.

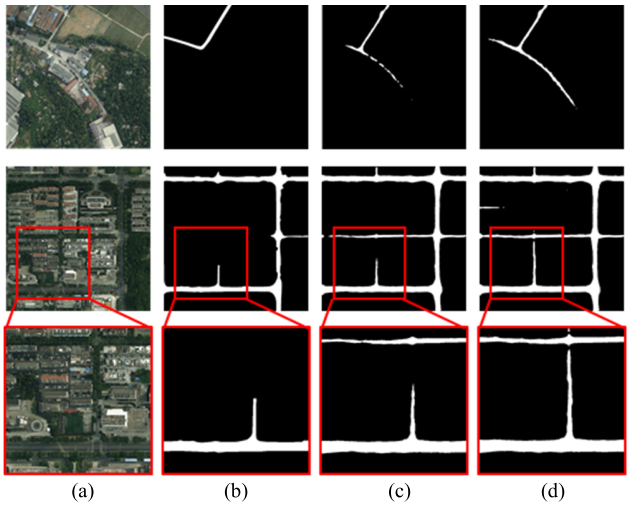


Fig. 11. Lack of road labels leads to the biased precision score on the Shaoxing data set. (a) Image. (b) Ground Truth. (c) D-LinkNet. (d) Fusion. The third row is the local close-ups of the second row.

Therefore, we stopped at 4 as a wider width indicated a more relaxed constraint for the indicators.

The ground truth of the samples heavily impacted the process of model training and the quantitative assessment of accuracy. The ground truths for our road data sets were derived from manually labeled GIS maps or OSM data. The quality suffered from annotations at different levels of detail (LoDs), outdated geospatial databases, and the complexity of various road types. Fig. 11 shows two examples from the Shaoxing data set where our method was able to predict narrow roads under trees and was more robust against occlusions than the D-LinkNet. However, due to the missing corresponding ground truth, the precision score of our method was lower than that of D-LinkNet.

Our road extraction method continues to be refined as well as our rule-making for the different levels of road annotations and corresponding finer evaluation criteria, which jointly are advancing the automation of road extraction.

V. Conclusion

In this article, a novel CNN-based multistage framework was proposed for simultaneous road surface and centerline tracing from remote sensing images instead of treating them separately as most of the current road extraction methods do. This multistage framework presents a coarse-to-fine road extraction approach. Based on the initial segmentation results, a boosting strategy was introduced to improve the segmentation accuracy, which especially increased the connectivity of road segments by learning the complementary information of previous segmentation maps and labels with an efficient encoder–decoder network. Then, an improved road centerline tracing method, which tracks road centerlines from multiple starting points that are automatically derived from the segmentation maps, was proposed to construct a more complete road network. Finally, the centerline graph rasterized with an adaptive width on the discontinuous segments of a segmentation map was fused with the road surface segmentation map to obtain the final road segmentation maps and centerline networks.

Our method was evaluated on three diverse data sets and proved to be superior than other current road segmentation and centerline extraction methods as far as extraction accuracy and especially road connectivity and completeness.

Our future work will address two aspects. First, due to the limitations of high-quality labels, it would be interesting to study a semisupervised model which can detect road and centerlines by using a smaller amount of training samples. Second, the regularization of road networks (i.e., using more structured polylines or polygons to fit the networks extracted by CNNs) would be a key step toward achieving accuracy that matches that of manual road delineation, which currently lacks investigation.

References

[1] A. Kirthika and A. Mookambiga, "Automated road network extraction using artificial neural network," in *Proc. Int. Conf. Recent Trends Inf. Technol. (ICRTIT)*, Jun. 2011, pp. 1061–1065.

[2] 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.

[3] 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.

[4] 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.

[5] Y. Zhang and T. Chen, "Efficient inference for fully-connected CRFs with stationarity," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1–9.

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

[7] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. ICLR*, 2015, pp. 1–14.

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

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

[10] 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.

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

[12] V. Badrinarayanan, A. Kendall, and R. Cipolla, "SegNet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[13] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.*, Oct. 2015, pp. 234–241.

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

[15] L. C. Chen, G. Papandreou, F. Schroff, and H. Adam, "Rethinking atrous convolution for semantic image segmentation," 2017, *arXiv:1706.05587*. [Online]. Available: https://arxiv.org/abs/1706.05587

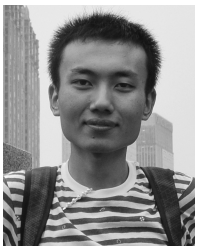
[16] V. Mnih and G. E. Hinton, "Learning to detect roads in high-resolution aerial images," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 210–223.

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

[18] T. Panboonyuen, K. Jitkajornwanich, S. Lawawirojwong, P. Srestasathiern, and P. Vateekul, "Road segmentation of remotely-sensed images using deep convolutional neural networks with landscape metrics and conditional random fields," *Remote Sens.*, vol. 9, no. 7, p. 680, 2017.
[19] Z. Zhang, Q. Liu, and Y. Wang, "Road extraction by deep residual U-net," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 5, pp. 749–753, May 2018.
[20] H. He, D. Yang, S. Wang, S. Wang, and Y. Li, "Road extraction by using atrous spatial pyramid pooling integrated encoder-decoder network and structural similarity loss," *Remote Sens.*, vol. 11, no. 9, p. 1015, 2019.
[21] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1904–1916, Sep. 2015.
[22] X. Yang, X. Li, Y. Ye, R. Y. K. Lau, X. Zhang, and X. Huang, "Road detection and centerline extraction via deep recurrent convolutional neural network U-Net," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 9, pp. 7209–7220, Sep. 2019.
[23] L. Zhou, C. Zhang, and M. Wu, "D-LinkNet: LinkNet with pretrained encoder and dilated convolution for high resolution satellite imagery road extraction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 192–196.
[24] A. Chaurasia and E. Culurciello, "LinkNet: Exploiting encoder representations for efficient semantic segmentation," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, Dec. 2017, pp. 1–4.
[25] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proc. ICLR*, 2016, pp. 1–13.
[26] Z. Zhang and Y. Wang, "JointNet: A common neural network for road and building extraction," *Remote Sens.*, vol. 11, no. 6, p. 696, 2019.
[27] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2261–2269.
[28] R. Liu *et al.*, "Multiscale road centerlines extraction from high-resolution aerial imagery," *Neurocomputing*, vol. 329, pp. 384–396, Feb. 2019.
[29] D. Costea, A. Marcu, E. Slusanschi, and M. Leordeanu, "Roadmap generation using a multistage ensemble of deep neural networks with smoothing-based optimization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 210–214.
[30] 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, Apr. 2009.
[31] 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.
[32] G. Mattyus, W. Luo, and R. Urtasun, "DeepRoadMapper: Extracting road topology from aerial images," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3458–3466.
[33] 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.
[34] X. Lu *et al.*, "Multiscale and multitask deep learning framework for automatic road extraction," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 11, pp. 9362–9377, Nov. 2019.
[35] C. Ventura, J. Pont-Tuset, S. Caelles, K.-K. Maninis, and L. Van Gool, "Iterative deep learning for road topology extraction," 2018, *arXiv:1808.09814*. [Online]. Available: http://arxiv.org/abs/1808.09814
[36] F. Bastani *et al.*, "RoadTracer: Automatic extraction of road networks from aerial images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4720–4728.
[37] I. Demir *et al.*, "DeepGlobe 2018: A challenge to parse the Earth through satellite images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 172–181.
[38] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *J. Comput. Syst. Sci.*, vol. 55, no. 1, pp. 119–139, Aug. 1997.
[39] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[40] Z. Tian, T. He, C. Shen, and Y. Yan, "Decoders matter for semantic segmentation: Data-dependent decoding enables flexible feature aggregation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3126–3135.
[41] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Trans. Syst., Man, Cybern.*, vol. 9, no. 1, pp. 62–66, Jan. 1979.
[42] J. Shi and Tomasi, "Good features to track," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Oct. 1994, pp. 593–600.
[43] T. Y. Zhang and C. Y. Suen, "A fast parallel algorithm for thinning digital patterns," *Commun. ACM*, vol. 27, no. 3, pp. 236–239, Mar. 1984.
[44] V. Mnih, "Machine learning for aerial image labeling," Ph.D. dissertation, Univ. Toronto, Toronto, ON, Canada, 2013. [Online]. Available: https://www.cs.toronto.edu/~vmnih/docs/Mnih_Volodymyr_PhD_Thesis.pdf
[45] M. Haklay and P. Weber, "OpenStreetMap: User-generated street maps," *IEEE Pervas. Comput.*, vol. 7, no. 4, pp. 12–18, Oct. 2008.
[46] *Google Earth*. [Online]. Available: https://www.google.com/earth
[47] V. Mnih and G. Hinton, "Learning to label aerial images from noisy data," in *Proc. 29th Int. Conf. Mach. Learn. (ICML)*, 2012, pp. 203–210.
[48] C. Wiedemann, C. Heipke, and H. Mayer, "Empirical evaluation of automatically extracted road axes," in *Proc. Empirical Eval. Techn. Comput. Vis.*, 1998, pp. 172–187.
[49] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," in *Proc. ICLR*, 2015, pp. 1–15.
[50] *PyTorch*. [Online]. Available: http://pytorch.org/

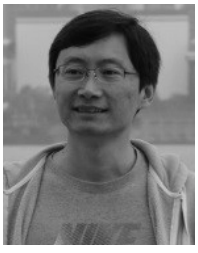
Yao Wei received the B.S. degree in geographic information science from the China University of Petroleum, Qingdao, China, in 2018. She is pursuing the M.S. degree with the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, China.

Her research interests include remote sensing image processing and machine learning.



Kai Zhang received the B.E. degree in remote sensing science and technology from Wuhan University, Wuhan, China, in 2019. He is pursuing the Engineering degree in communication science and technology with ENSTA ParisTech, Palaiseau, France.

His research interests include image processing and machine learning.



Shunping Ji (Member, IEEE) received the Ph.D. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2007.

He is a Professor with the School of Remote Sensing and Information Engineering, Wuhan University. He has co-authored more than 50 articles. His research interests include photogrammetry, remote sensing image processing, mobile mapping system, and machine learning.