DeepWindow: Sliding Window Based on Deep Learning for Road Extraction From Remote Sensing Images

Renbao Lian and Liqin Huang

***Abstract*—The road centerline extraction is the key step of the road network extraction and modeling. The hand-craft feature engineering in the traditional road extraction methods is unstable, which makes the extracted road centerline deviated from the road center in complex cases and even results in overall extracting errors. Recently, the road centerline extraction methods based on semantic segmentation employing deep neural network greatly outperformed the traditional methods. Nevertheless, the pixel-wise labels for training deep learning models are expensive and the postprocess of road segmentation is error-prone. Inspired by the work of human pose estimation, we propose DeepWindow, a novel method to automatically extract the road network from remote sensing images. DeepWindow uses a sliding window guided by a CNN-based decision function to track the road network directly from the images without the prior of road segmentation. First of all, we design and train a CNN model to estimate the road center points inside a patch. Then, the road seeds are automatically searched patch by patch employing the CNN model. Finally, starting from seeds, our method first estimates the road direction using a Fourier spectrum analysis algorithm and then iteratively tracks the road center-line along the road direction guided by the CNN model. In our method, the CNN model is trained by point annotations, which greatly reduces the training costs comparing to those in semantic model training. Our method achieves comparable performance with the state-of-the-art road extraction methods, and extensive experiments indicate that our method is robust to the point deviation.**

***Index Terms*—Deep learning, remote sensing images, road extraction, sliding window.**

I. Introduction

ROADS are important objects in geographic information systems as man-made objects. Road networks are applied in many aspects of social life, such as vehicle navigation, traffic management, map updating, geological disaster emergency, and humanitarian aid. In modern life, people request higher demand for the update speed and accuracy of road information than before. High-resolution remote sensing imagery is an important avenue to automatically infer the road networks. However, the obscures by vehicles, trees, and buildings make the automatic road extraction very difficult [1]. How to improve the efficiency and accuracy of road network extraction is a hot issue at present.

Road extraction from remote sensing images can be divided into two levels: road region segmentation and road network extraction. Road region segmentation is to classify each pixel in the image into roads and nonroads, while road network extraction is to obtain the road centerlines and their connectivity. In most tasks, the classification of each pixel is not the ultimate goal of road extraction. The road network can provide more comprehensive information, as it provides connectivity between different pixels and the road network can be directly used in many applications [2].

The road network extraction methods mainly include template matching, shortest path, semantic segmentation combining postprocessing, and road tracking based on the patched model. Template matching methods [3]–[7] are the classic road extraction methods in which several patches are cropped within a certain direction range along the tracking direction, and the best matching patch is determined according to the hand-crafted matching rules, and the center of the patch is regarded as the road central point, and finally, the adjacent road central points are connected to obtain the road centerline. These methods work well in good conditions. However, when the road is covered by noise or the color changes dramatically, a large number of matching errors will emerge. The shortest path methods [8], [9] search the path between two seeds with the minimum cost. The fast-matching method [10] is a classic optimization algorithm for solving the shortest path problems. The algorithm accumulates the reciprocal of gradient as the cost from the starting point to the end point, and then traces the path of the minimum cost from the end point to starting point. Thus, the extracted paths are inclined to the road edges. Miao *et al*. improved the shortest path method to extract the road centerline by computing the shortest path twice, but greatly increase the computation cost, and also fail to track the correct path when the situation is complicated. Some extraction errors of template matching, shortest path, and improved shortest path are shown in Fig. 1 from (b) to (d), respectively. Template matching algorithms likely cause the problems

Manuscript received January 1, 2020; revised March 2, 2020; accepted March 22, 2020. Date of publication April 20, 2020; date of current version May 21, 2020. This work was supported in part by Education and Scientific Research Project for Middle-aged and Young Teachers in Fujian Province under Grant JT180595 and in part by Key Program of Fostering Young Scientific Research Talents in Fujian Jiangxia University under Grant JXZ2018002 and Grant JXZ2016001. *(Corresponding author: Liqin Huang.)*

Renbao Lian is with the College of Physics and Information Engineering, Fuzhou University, Fuzhou 350008, China, and with the Digital Fujian, Internet-of-Things Key Lab of Information Collection and Processing in Smart Home, Fuzhou 350108, China, and also with the College of Electronics and Information Science, Fujian Jiangxia University, Fuzhou 350108, China (e-mail: luoshao@163.com).

Liqin Huang is with the Collage of Physics and Information Engineering, Fuzhou University, Fuzhou 350008, China (e-mail: hlq@fzu.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2020.2983788

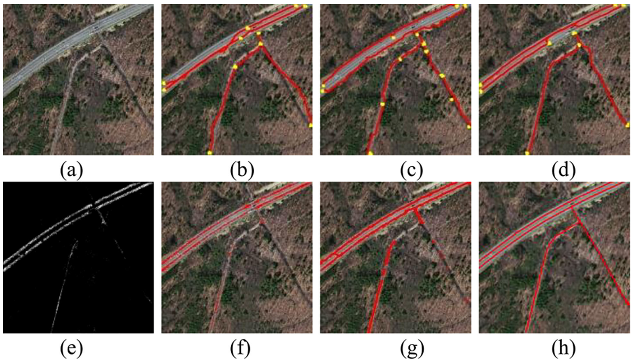


Fig. 1. Problems exist in current road extraction methods. (a) Input image. (b)–(d) Effects of traditional road tracking methods that are template matching method, shortest path method, improved geodesic distance method respective, where yellow points represent manually specified seed points, and red is the extracted roads. (e)–(g) Effects of semantics segmentation combining postprocessing, which are the semantic segmentation, the binarized segmentation overlaid on the image, and the result of postprocessing respective. (h) Effect of our method.

of crossing roads when roads are adjacent and matching errors in the place where roads are covered by noise. The methods of segmentation combining postprocessing [11]–[14] first divide pixels into road and nonroad, and then extract the road centerlines by heuristic rules. Road segmentation can be acquired by traditional segmentation algorithms, but better results can be obtained by deep learning models [15]–[18], and then the key decisions on how road segments are interconnected are delegated to the error-prone postprocessing stage that relies on heuristics algorithms [1]. It is difficult to guarantee the universality of these algorithms. Moreover, these methods rely seriously on the road segmentation which almost determines the result of road extraction. Fig. 1(e)–(g) show a case that the segmentation containing numerous gaps leads to poor graph connectivity. The road tracking methods based on the patched models require precise annotations (such as GPS trajectory, pixel-wise labels) to train the CNN model and to predict the global road probability map for guiding the tracking process, which increases the cost of model training.

Given the problems of existing road tracking methods, we propose DeepWindow, an approach that uses a patched CNN model for tracking the road network from remote sensing images. Different from previous ones, our method extracts the accurate road centerlines directly from images without the guidance of road segmentation. Most importantly, we do not need to prepare a precise road mask. Each training sample is only labeled by a few points, which greatly reduces the cost of model training. Besides, our algorithm can auto search tracking seeds, which makes the tracking algorithm fully automated. Specifically, we train a CNN model in advance, which outputs a probability map containing the confidence of each pixel as a road central point in a patch. Then, the CNN model is used to seek road seeds automatically, which are the starting points of our road tracking method. We also use the CNN model to infer the confidence map of each patch centered at each sliding step and pick out some points by NMS [19] as the candidate road central points from the map. The optimal point is determined according to the local characteristics of roads. Finally, we connect the selected point and previous road central point using a straight line and step forward with a fixed distance. The above process is executed iteratively, and finally, a complete and fine road network is tracked, as shown in Fig. 2. More samples are available in,[1] The contributions of this article are summarized as follows.

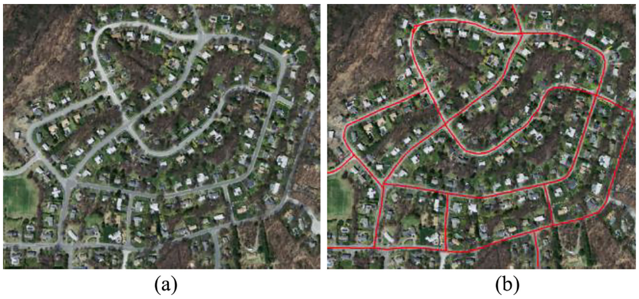


Fig. 2. Road network extraction from aerial images. (a) Input image. (b) Road extraction via our method.

1) A patched-based road center estimation model is designed and trained by point annotations, which predicts the road central points in a local patch.
2) An algorithm of road direction estimation is proposed to increase the automation of our tracking process, which estimates the road direction according to the Fourier spectrum of canny edges.
3) An algorithm of automatic seed searching is implemented, which makes the tracking process fully automated combining with the road direction estimation algorithm.
4) A large and challenging road patch dataset with manually sampled road center points for road extraction will be publicly available for further studies of weakly supervised learning.

II. Related Work

Road extraction from remote sensing images can be divided into two levels: road area segmentation and road network extraction. Road area segmentation classifies each pixel of the image into roads or nonroads. Road network extraction is to obtain the road centerlines or boundaries, and, finally obtain the topology of the network.

Road area segmentation: Road area segmentation is the preliminary task of road extraction, which can be used as guidance of the topology delineation. Manually classifying the road areas is generally time-consuming and contains abundant errors introduced by the operator [20]. In recent years, deep learning technologies have made great progress in computer vision, and many of them have been introduced into road area segmentation in remote sensing images. Mnih *et al.* [21] and Rezaee *et al.* [22] proposed a patch-based deep neural network to label road regions in aerial images. These methods infer the road map of an image patch by patch. These algorithms are computation consuming, because the correlation of adjacent pixels is not fully utilized, and the overlapping patches centered at adjacent pixels are computed repeatedly. Zhong *et al.* [23] introduced the fully convolutional network into the segmentation of the road area

[1] Online. [Available]: https://github.com/rob-lian/DeepWindow

and achieved dense end-to-end reasoning. But the simple linear interpolation upsampling of FCN made the model performance poor. Evolved from FCN, UNet has a symmetric encoder–decoder structure and the decoder uses parameter-learnable deconvolution, which makes the semantic segmentation more accurate. Zhang *et al.* [17] and Alexander *et al.* [24] proposed an improved UNet network for road area segmentation, which achieved higher performance using ResNet as its encoder. To expand the feature reception field without sacrificing the feature space resolution, the literature [16], [18], [25], [26] designed a UNet-like network with atrous spatial pyramid pooling (ASPP) to further improve the accuracy of road segmentation.

Road network extraction: Compared with road segmentation, road network provides more extensive and meaningful information [2]. It is also the ultimate goal of the road extraction task and can be used to update and optimize road database [27]. Cao *et al.* [28] proposed a method that locates the rough road positions using GPS initially and then adjusts the GPS coordinates to the road center or extracts the road shape exploiting the intensity of the local region. With the help of OpenStreetMap (OSM), Mattyus *et al.* [13] regard road topology extraction as a parametric Markov random field reasoning problem of road location and width, and the extracted vector data, in turn, optimizes OSM data. The above works are devoted to the refining of the road network using the existing vector data. In paper [1], a dynamic graph construction method is proposed to track the road network iteratively. Under the guidance of OSM, this method trains a decision network by dynamically generating training samples. Then an iterative road tracking is implemented depending on the direction estimation inferred by the decision network according to the local region. Other works are devoted to road centerlines extraction based on the road segmentation. Miao *et al.* [11] presented a Gaussian mixed model for extracting the centerlines from the segmented road regions. The road regions are cut into different road segments that are fitted with a Gaussian mixture model, and the long axis of each Gaussian ellipse is the initial centerline of the corresponding road segment. Finally, the SCMS [29] algorithm is used to adjust the initial centerline to the exact position. Mattyus *et al.* [13] proposed an approach that directly estimates road topology from aerial images, taking advantage of the latest developments in deep learning to have an initial segmentation of the aerial images. Ventura *et al.* [2] provided an iterative tracking method for road topology extraction under the guidance of initial road segmentation, in which a CNN network was trained to predict the local connectivity among the central pixel of an input patch and its border points. There are many other studies devoted to extracting road topology directly from remote sensing road images without any auxiliary information (e.g., GPS, DEM, and segmentation), and template matching is a family of classical methods [5], [7], [30]–[32]. Lin *et al.* [32] employed a least-squares rectangular template matching to track the road axis with lane markings in urban areas. An adaptive circular template was proposed by Lian *et al.* [7], which automatically adjusts the seeds to road center and extracts road topology by iterative interpolate template matching. Most of the traditional template matching methods are designed by artificial specified descriptors and matching rules which cannot fulfill the various complicated conditions in remote sensing imagery.

III. Methodology

This section presents our approach which traces the road network with a sliding window, as shown in Fig. 3. More precisely, when a seed point (e.g., O) and marching direction (e.g., $\vec{d} = \overrightarrow{e_1 e_2}$) are initiated automatically, our approach crops a patch from the image along the direction $\vec{d}$ with a step forward (e.g., e_1). The trained road center estimation model outputs the road center probability map taking the patch as input. We pick some peaks above a hard threshold (we set the threshold to 0.05 for all experiments, see Section IV-C) and finally retain the one (e.g., $\hat{O}$) which is closest to the direction of the road extension, note that roads present slowly curved stripes in a small patch. Then, the points of O and $\hat{O}$ are simply connected by a straight line because the shape of the road is a straight line in a sufficiently local area expect for the T-junction or L-turn which will be discussed later. After that, our approach repeats the process along the direction of $\overrightarrow{O\hat{O}}$.

Section III-A presents a modified stacked hourglass networks that learn to predict the road center probability of each pixel in the patch. The model is used to find initial seeds and also embed into the tracking process as kernel decision function to estimate the road central points in a local patch. The seeds searching are introduced in Section III-B. Section III-C describes the road direction estimation based on a patch centered at the seed. The road network is tracked by a sliding window, as explained in Section III-D.

A. Road Center Estimation Model

As mentioned above, the function of the model is to estimate the road central point of a given patch. To solve this issue, we consider the road central point as a special point in the context of a patch. Our goal is to design a deep neural network to infer the possible road central points according to the local texture. We found that special points estimation has already been researched for decades in the human pose estimation field. To estimate the human pose, the position of human joints such as head, neck, shoulders, and elbows need to be located first which are regarded as the special points according to the local context of the image. These points are encoded as heatmaps with Gaussians center on them. Inspired by the idea, we also encode the road central points with Gaussians if there are roads in the patch as the ground truth for model training. We find in the experiments that the model is indeed capable of learning the relation between the road central point and the local texture.

A research team of Michigan University proposed a stacked hourglass networks for human pose estimation [33]. This model employees a fully convolutional network to output the precise pixel position of a human key point for a given single RGB image and uses multiscale features to capture the spatial position information of each joint point of the human body. The kernel of the network is designed like an hourglass, and the top-down to bottom-up is repeated to infer the position of the joint points

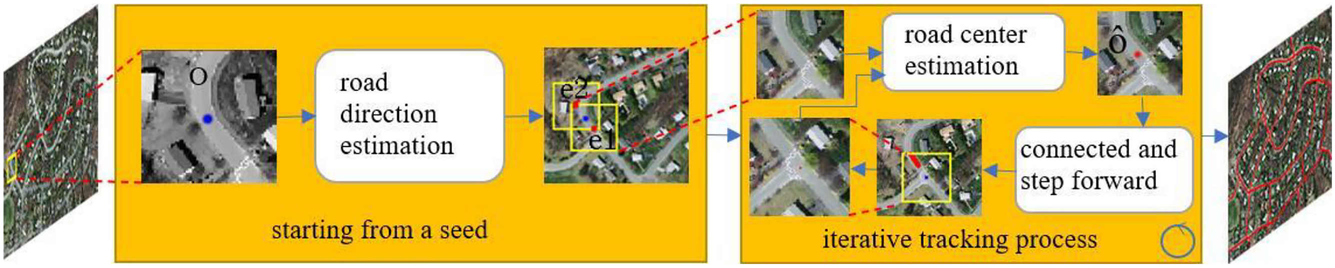

O
road direction estimation
e2
e1
starting from a seed
road center estimation
ô
connected and step forward
iterative tracking process

Fig. 3. Iterative tracking process starting from a seed that is automatically searched.

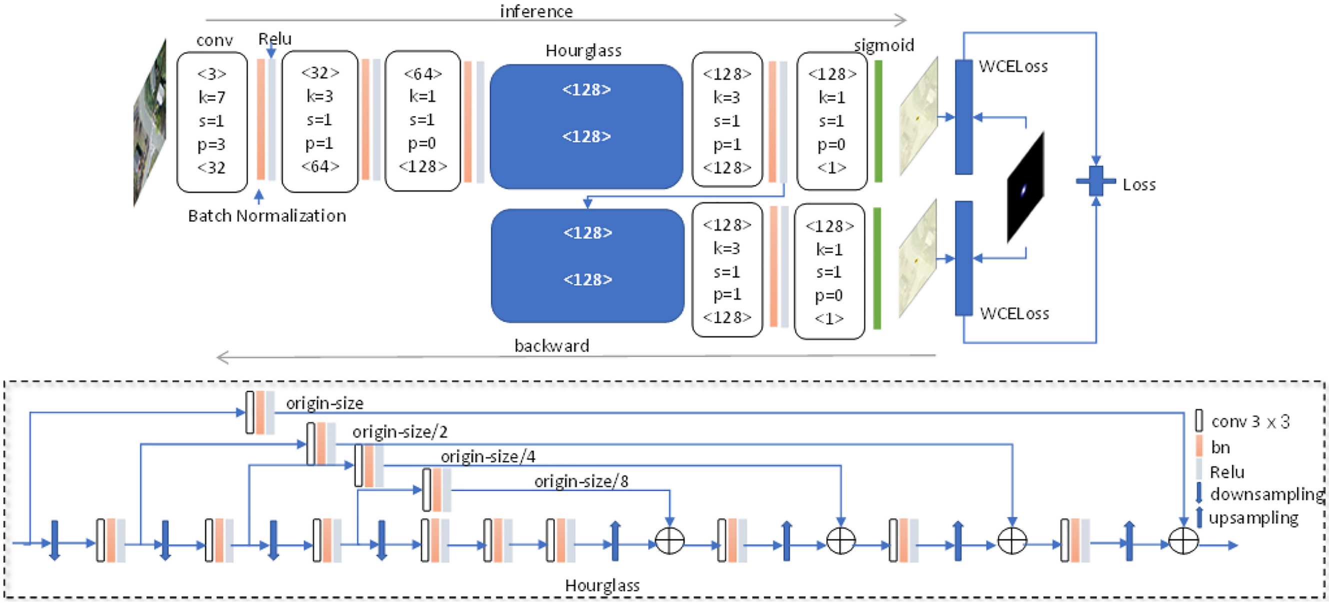


Fig. 4. Architecture of the road center estimation model.

of the human body. Each top-down to bottom-up structure is an hourglass module. Many subsequent human pose estimation methods evolved from the hourglass networks structure. It can be said that the hourglass networks structure has been approved by the industry. We also take the architecture of stacked hourglass networks to build our road center estimation model, as shown in Fig. 4. More specifically, we stack two hourglass models in the network and replace the first convolutional layer as kernel size 7×7, the stride 1 and padding 3 pixels to keep the resolution consistent as the input patch. Furthermore, we reduce the depth of the output of each hourglass model to 1, that is, we force the network to encode the probability of each pixel in one channel no matter how many road centers in the patch. The dotted box in Fig. 4 details our modified hourglass model, in which the input is gradually downsampled four times and each scale keeps the consistent number of features across the whole hourglass.

We propose a weighted cross-entropy loss (WCELoss) for the model training, which can be written as (1), where N is the number of pixels in a sample patch, y_i is the ith value in the ground truth, which is close to 1 near the road centers and close to 0 in other position, see Section IV-A for detail, and $\hat{y}_i$ is the ith value of output map. e^{y_i} is the weight factor to penalize the error occurs near the road centers. λ is a scale factor, which is used to control the ratio of the loss coming from the errors near the road centers. The simple idea behind the WCELoss is to consider the problem of extreme imbalance labels in supervised training. All the labels in the ground truth are almost zeros except for a small number of nonzeros around the road centers, which depresses the output of the model. In other words, the model can achieve lower loss by predicting the outputs approaching zeros if the error of each position in the patch is measured equally. We can increase λ to raise the global output of the model, but the increment of global output does not affec the judging of the road centers because we choose them by screening the peaks of the output, therefore, λ is fixed to 1 in our experiments

$$\text{WCELoss} = \frac{1}{N}\sum_{i=1}^{N} e^{y_i}\left[\lambda y_i \log\left(\hat{y}_i\right) + \left(1 - y_i\right)\log\left(1 - \hat{y}_i\right)\right]. \tag{1}$$

B. Automatic Searching of Tracking Seeds

Once the road central point estimation model is trained, we can automatically find the seeds for road tracking. Patches with the resolution 64×64 are consecutively cropped from the image from left to right and top to bottom without overlap, and the road central points of each patch are inferred using the road central point estimation model. For one patch, we only keep one point as the seed whose confidence is the max and exceeds a certain threshold. It is noteworthy that we empirically set the confidence threshold to a high value to ensure the precision of seeds. For a 1500×1500 image, there may be numerous seeds found automatically, while many of them may not be used. When we start tracking from a seed, many automatically searched seeds will be visited during the tracking process and these visited seeds will be discarded later.

C. Road Direction Estimation

Usually, most tracking algorithms used to track filaments need the starting point and the initial tracking direction. In order to reduce the artificial intervention, we estimate the tracking direction automatically. According to the priori of a road in high-resolution remote sensing image, the edges of a road are obvious and sharp contrast to the texture of the surrounding environment. Especially in the scenario of this article, the road direction is estimated only at the locations of road seeds based on a small patch, in which the road edges are more obvious and with fewer interferences. Thus, the road features in these local areas are naturally more prominent. We can estimate the main direction robustly.

It is ubiquitous that meaningful structures are formed by or appear over textured surfaces, which could be regular, near-regular, or irregular [34]. The texture is directional, and the direction of the texture is a regional concept. It is meaningless to talk about directionality for an isolated pixel. In other words, the road direction refers to the direction of the regular texture inside a patch. The regularity and periodicity of the regular texture make it possible for texture primitives to embody some characteristics on the whole, such as direction. Edges have great influence on human vision, and the predominant direction of the edges represents the orientation of the texture. From the perspective of Fourier spectrum, for a regular texture with the same directionality, its energy in Fourier spectrum is concentrated on a line passing through the origin, and the direction of the line is perpendicular to the orientation of the texture. More specifically, canny [35] is used to scan the edges of the grayscale patch after bilateral filtered [36], and then the scanned patch is converted into discrete Fourier spectrum. Finally, the direction of the principle spectrum energy $\theta_{\max}$ is computed according to (2) and (3), in which *w* is the patch width, fft_img is the Fourier spectrum of the canny edges of the patch, and the spectrum energy *E* is calculated every 1 degree. The orthogonal direction of $\theta_{\max}$ is regarded as the direction of the road, as shown in Fig. 5

$$E(\theta)|_{\theta\in[0,\pi)} = \sum_{r=1}^{w/2} \mathrm{fft\_img}\left[\frac{w}{2} + r \times \sin\theta, \frac{w}{2} + r \times \cos\theta\right] \tag{2}$$

$$\theta_{\max} = \operatorname{argmax}(E). \tag{3}$$

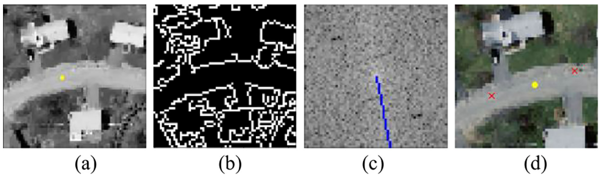


Fig. 5. Predominant direction estimation. Left: a bilateral filtered image patch with the superposition of initial seed. Middle: edges scanned by canny and its principle spectrum energy direction marked by a blue line. Right: Road direction estimated by our method and the two red crosses are the extension seeds for our sliding window algorithm.

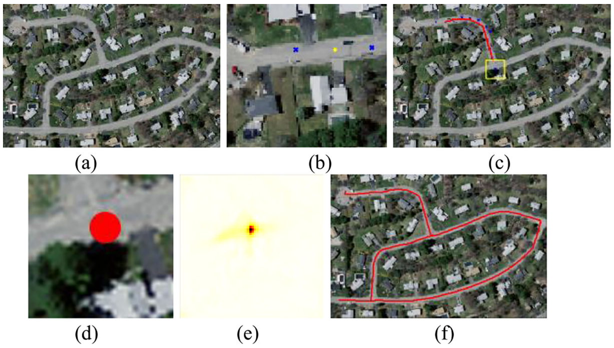


Fig. 6. Sliding window for road topology tracking. (a) Input image. (b) Seed is shown in yellow dot and the extension points estimated by the local spectrum analysis are shown in blue crosses. (c) Current step of the sliding window where the corresponding patch (yellow rectangle) is cropped to predict road central points. (d) Cropped patch and the predicted road central point. (e) Corresponding road center probability map inferred by our road center estimation network. (f) Final result.

D. Iterative Topology Tracking

We regard the topology tracking as a sliding window process depending on the patch-level road center estimation model. First, numerous seeds are searched automatically using the road center estimation model. Starting from one of the seeds, a patch with resolution 64×64 centered at the seed is cropped. Then, the predominant direction of the road in the patch is estimated by the spectrum analysis algorithm. After that, the window steps forward with a fixed distance along the road direction, and the patch corresponding to the window is cropped and fed into the road center estimation model that will output a probability map (the output of the second hourglass) with the same size as the input patch. The probability of each pixel in the map means the confidence of the pixel in the patch as a road center. We regard the location p_c with the modest probability, which is closest to the sliding direction as the most likely road center. Finally, p_c is connected to the previous road center p_v using a straight line.

Then, the window steps forward with the fixed distance again along the direction of vector $\overrightarrow{p_c p_v}$ like inertia and repeat the above process. On the other hand, during the sliding window process, If the maximum value of the output probability is below a certain threshold, which means that the road center estimation model cannot find a road center in the patch. In this situation, we stop the current tracking process and pop another seed to restart tracking if the seed is unvisited. A tracking sample is shown in Fig. 6

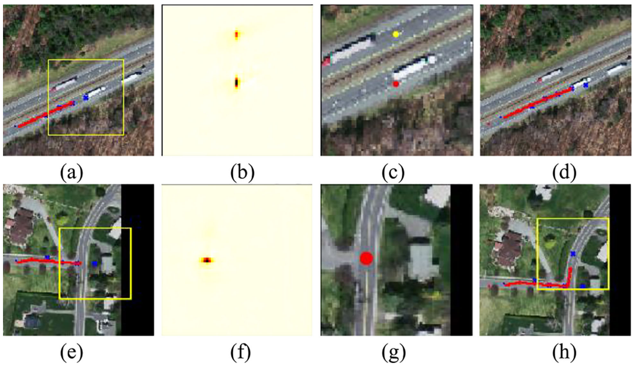


Fig. 7. First-row: Illustration of the solution when model outputs multi points. From left to right: The current state of sliding window, the road center probability of the patch, the estimated road centers marked by dots, and the final linking result. Second-row: Solution when sliding window encounters the T-junction. From left to right: The current state of sliding window, the road center probability of the patch, the estimated road centers marked by dot, turning to the perpendicular direction.

There are several other situations that we need to pay attention during the tracking process. When multiple parallel roads are close to each other, the cropped patch will contain more than one road. The road center estimation model will output multiple road centers with high confidence. At this moment, as shown in the first row of Fig. 7, we choose the point which is closest to the extension direction of the current topology as the optimal point because road direction changes slowly in the local area according to the road priors. Other points are pushed into the seed stack as new seeds. Another case is that when the tracking process encounters a T-junction or a sharp turn, the road central point predicted by the model will be very close to the previous one. At this situation, we can judge that the road has been tracked to the end or the sliding window encounters a T-junction, as shown in the second row of Fig. 7. To continue tracking, we try to find the road center in the perpendicular direction against the current tracking direction. The tracking process will continue if a road center is found in the perpendicular direction. For clarity, our algorithm is shown in Table I.

IV. Experiments

To evaluate our approach, we carry out extensive experiments on two publicly available datasets, which are Massachusetts roads dataset [37] and a dataset collected from *Google Earth* [38]. The Massachusetts roads dataset is an aerial imagery dataset containing 1108 images for training, 14 images for validation, and 49 images for testing. Each image consists of 1500 × 1500 pixels with spatial resolution 1 m. The Google Earth dataset contains 224 VHR images that manually labeled the road segmentation reference maps and corresponding centerline reference maps. The original images in Google Earth dataset are with a spatial resolution of 1.2 m per pixel and there are at least 600 × 600 pixels in each image. As Fig. 8 shows, the images in Massachusetts roads dataset are under complex backgrounds and occlusions of cars and trees, and the road segmentations constructed from maps are suffered from omission and registration noise [39]. Although the reference maps in Google earth dataset are accurate, the brightness of different images varies greatly and the color of different road parts is different as black and white in an image. All these problems make the road extraction task very challenging.

TABLE I
SLIDING WINDOW FOR ROAD NETWORK TRACKING

Input:

Initial seed stack.

Procedure:

Step 1: A seed is popped from the seed stack called p_s.

Step 2: A patch with resolution 64×64 center at p_s is cropped. According the patch, the predominant direction of the road is estimated by spectrum analysis. Then we obtain two extension points before and after p_s along the road direction. The two pairs of seed and extension point are pushed into the extension stack.

Step 3: Pops a pair of seed and extension point from the extension stack if the stack is not empty, or else go to step 8.

Step 4: Crops the patch centered the extension point and fed into road center estimation model which will output the road center probability map. Picks some peaks from the probability which are above a hard threshold or go to step 3 if none the peaks are fulfill the threshold.

Step 5: Choses the point (e.g. p_0) which is closest to the sliding direction with a modest distance to the previous road center (the distance threshold is set to 2 pixels) and go to step 6. If the point is not existing, we push the two pairs of (p_0, p_1) and (p_0, p_2) to the extension stack and go to step 3. Note that $\overrightarrow{p_1, p_2}$ is perpendicular to the current sliding direction.

Step 6: Links p_0 to previous road center p_v if p_0 does not encounter previous visited regions with two successive times, or else goes to step 3. This rule gives our algorithm a chance to cross the intersections which are visited before. Add the path (p_v, p_0) to road network.

Step 7: Steps forward with a fixed distance and reach a new point called p_e, and pushes the pair of (p_0, p_e) into the extension stack, goes to step 3.

Step 8: Stops the current tracking process, goes to step 1 if the seed stack is not empty or else output the tracked network.

Output:

Road network.

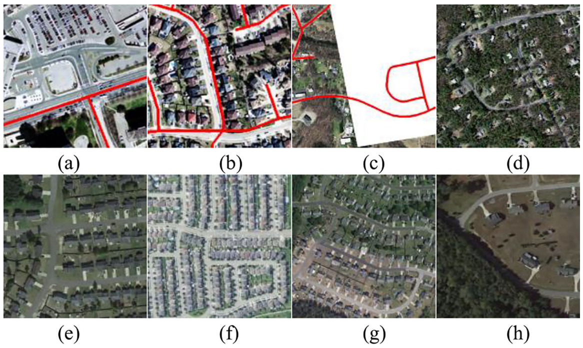


Fig. 8. Illustration of some representative images in two datasets. (a)–(d) Examples from Massachusetts Roads Dataset. (e)–(h) Samples from Google Earth. (a) Example of omission noise. (b) Example of registration noise. (c) Example with large invalid area. (d) and (h) Examples of Occlusions of trees. (e) and (f) show the brightness contrast. (g) Example of different color on one road.

A. *Preparing Training Samples*

To train the road center estimation model, taking Massachusetts roads dataset as an example, we randomly cut 50 patches with the resolution of 64 × 64 from each image in the training set, and finally, obtain 55 400 training samples. In the

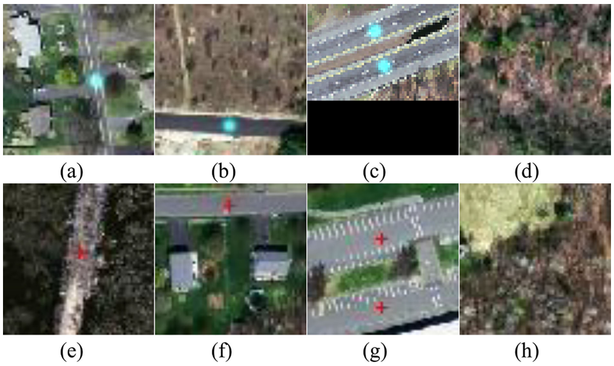


Fig. 9. Some training samples and visual prediction results of the road center estimation model applied to patches from the Massachusetts Roads dataset. First-row: Some training samples shown in the composition of the patch and ground truth. Second-row: Some of the prediction samples, and the road centers are marked by red cross.

Massachusetts roads dataset, Mnih *et al.* intentionally placed large blank areas in many images in order to study the problem of annotation noise, as shown in Fig. 8(c). In order to reduce the impact of these anomalous annotations, we discard the continuous blank parts of each image, i.e., sampling from its valid regions. To ensure the diversity of training samples, we ensure the road existence in half of the samples and do not guarantee the road existence for another half. More importantly, the road center points must not be fixed in the patch center, or the model will learn the location characteristics, which makes the model has a strong preference to regard the patch center as a road center. For the first half samples, we first skeletonize the ground truth mask corresponding to the training image, randomly pick a point p within a certain range away from the skeleton. Then, we crop a patch from the training image centered at p. Meanwhile, we crop the same patch from the ground truth and capture the central point of each road segment in it. After that, we generate a mask of Gaussians centered at each road central point as the ground truth of the patch. Another half of the samples are randomly cut from the valid region of the training images, and then generate the training ground truth with the same rule mentioned above. These samples guarantee that the model can output a lower confidence map when there are no roads in the patch, which plays an important role in judging the end of the tracking process. In the same way, we cut 50 patches from each validation image for model evaluation. Some training samples are shown in Fig. 9. It is worth noting that the sample preparation described here is to automatically obtain point annotations from the global mask to avoid manual labeling. However, we need to manually mark the road central points in a new dataset because we can easily obtain a large number of remote sensing images but struggle to acquire the precise pixel-wise labels. The samples from Google Earth dataset are generated in the same rules, but we adjust the brightness of origin images using histogram equalization before cropping the training samples, which reduces the impact of brightness differences.

B. Training Details

We construct the CNN model by stacking two hourglass modules according to the paper [33], refer to Section III-A for detail. The model used in Massachusetts roads dataset is trained for 250 epochs with a min-batch size of 64 using Root Mean Square Prop [40] and 1000 epochs with a min-batch size of 32 using the same optimizer for Google Earth dataset. The learning rate is fixed to 1e-5, the parameter alpha equals 0.99, momentum equals 0. The weights of convolutional kernels are initialized with the normal distribution with a mean of 0 and standard deviation of $\sqrt[2]{2/n}$ where n is the number of the parameters of the convolutional kernels. We train the CNN network on a single NVIDIA RTX2080Ti GPU.

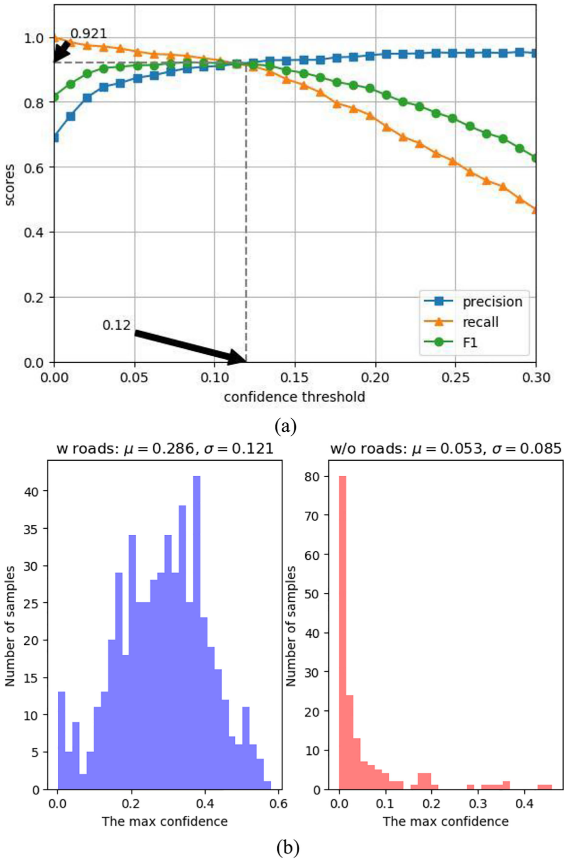


Fig. 10. (a) Relationship of the CNN performance and the confidence threshold, in which a pixel with the peak confidence above the threshold is regarded as a road central point. (b) Statistics of max road center confidence of validation samples grouped by with roads and without roads.

C. Model Evaluation

Taking Massachusetts roads dataset, for example, the trained road center estimation model (trained by manual sampled points set, see Section IV-D) gets 92.1% F1-score on the validation set when the confidence threshold is set to 0.12, as shown in Fig. 10. More specially, we regard the inference of the road centers as correct if all of the points (selected by NMS [19]) are located on road mask (the ground truth in Massachusetts roads dataset), and we regard the patch as road nonexistence if none of the peaks exceeds 0.12. It worthy to note that we cannot calculate the precision as the ratio between the correctly predicted centers and all the predicted centers because there are a lot of samples without roads. The calculation of recall also has the same concern. We regard a prediction is true positive if a sample has roads in the patch and all the predicted points fall on the roads, and false positive if any of the predicted points are outside the roads. Meanwhile, we define a prediction is a true negative if the patch has no roads and all the confidence peaks are below 0.12, otherwise false negative. The precision, recall, and F1 score are calculated by (4). It can be seen from Fig. 10(a) that when the threshold is set between 0.04 and 0.15, F1-score remains high and stable, indicating that our model is not sensitive to the confidence threshold, which also means that the setting of confidence threshold can be more flexible in the process of road tracking. We also infer each validation patch and count the max outputs group by roads and nonroads, respectively. The statistics show that the mean of the max outputs of the patches with roads inside is 0.286 and 0.053 to those without roads, and the standard deviations of the max values are 0.121 and 0.085, respectively, as shown in Fig. 10(b). The statistics indicate that there is a wide confidence gap between the patches with roads and those without roads. These statistics further provide the basis for us to set the confidence threshold to judge the road existence during the road tracking, and we fix the confidence threshold to 0.1 in the following experiments:

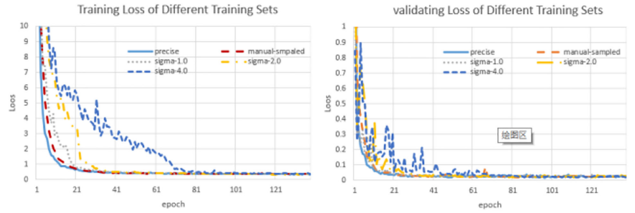
$$\text{precision} = \frac{TP}{TP + FP},$$
$$\text{recall} = \frac{TP}{TP + FN},$$
$$F1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}. \tag{4}$$

In Google Earth dataset, the road center model gets the best F1-score of 92.0% on the validation set when the confidence threshold is set to 0.1.

The second row of Fig. 9 shows the predicted results of some validation samples. We found that the model indeed learned the features of the road inside the patch. Our model can accurately predict the location of the road center regardless of the road direction, curvature, modest tree occlusion, or similarity with the background. In particular, if there are multiple road segments in the patch, the model can output the central point of each road target separately, which provides a more robust discriminant basis for the following road tracking.

D. Point Sensitivity Evaluation

As mentioned in Section IV-A, the training samples are generated from the skeletons of the ground truth mask. For each training patch, an undirect graph is constructed to calculate the center point sets, in which the nodes are the pixels of the skeletons corresponding to the patch and the edges connect between the adjacent pixels. The center point set consists of the nodes with eccentricity equal to the radius in each connected component. Finally, we select a point from each center point set for the corresponding road segment in the patch. However, the center points need to be placed without the guidance of ground truth in practice. It is difficult to manually locate the exact center points of the road segments. In order to evaluate the impact of the inaccuracy of road center points, we intentionally deviate the road center point from the road center. First, an exact center point p_c is selected as mentioned above. Second, p_c is vibrated on both sides of its original position along the road skeleton, and the vibration amplitude subjects to normal distribution of 1/4 road length, named p_v. Third, p_v is further deviated from the road skeleton in the orthogonal direction of road in accordance with the same distribution, and the deviation amplitude is 1/2 of road width, called p_{vv}. Finally, a 2-D Gaussians mask centered at p_{vv} is generated as the ground truth of the patch. The parameter in normal density function determines the overall deviation of sampled points. In our experiments, we set σ to 1.0, 2.0, and 4.0 and the average deviation is 3.75, 5.08, and 5.90 pixels, respectively. Moreover, we implement a program[2] to sample the road center points from the original images manually. The manually sampled point dataset will be publicly available for further studies. We stipulate that in a small patch, if there are multiple roads, a point needs to be placed in each road, which is roughly the center position of the road measured by human eyes. We trained the CNN models from scratch separately on the different road center point sets. Fig. 11 shows the training and validating loss curves, which means that all the trained models converge to the same state. From these comparative experiments, we conclude that the CNN model is not sensitive to the accuracy of road center points. More importantly, we found that the model trained on the manual sampled points set performs better. The explanation is that when the computer automatically generates the samples, it only considers the center of the road skeleton, but neglects to consider the situation of multiple road segments overlapping each other in a patch, and at the same time, it also regards the patches as positive samples when the roads in the patch are totally covered by trees, which causes a degree of confusion of the model.



(a) (b)

Fig. 11. Training and validating loss curves are shown in (a) and (b), respectively. The “precise” curve in (a) shows the training loss trained on the precise points. The “hand-sampled” curve in (a) shows the training loss on the points sampled manually. The legends ‘sigma-x’ represents the training loss on the deviated points auto-sampled by normal distribution in which σ is set to 1.0, 2.0, and 4.0, respectively.

E. Patch Size Evaluation

In the process of tracking the road network, the texture inside the small patch is the sole basis for our model to predict the road center points. Therefore, the more obvious the road features, the more it can help the model determine the road center points. The features of the road are generally reflected by the internal characteristics of the road and the background around the road.

[2] Online. [Available]: https://github.com/rob-lian/DeepWindow/PointSampleTool

TABLE II
COMPREHENSIVE COMPARISON OF DIFFERENT PATCH SIZE

Size	Ratio	Roads	W/wo	F1-score	$F1_R$
32x32	0.441	1.022	4.468	0.908	0.821
48x48	0.314	1.032	4.680	0.909	0.872
64x64	0.248	1.067	8.174	0.920	0.932
80x80	0.201	1.073	6.726	0.928	0.904
96x96	0.170	1.133	5.054	0.917	0.841

"Size" denotes the patch size; "ratio" equals the width of the road divided by patch width; "roads" represents the average road segments in a patch; "F1-score" is the tradeoff precision and recall of the trained model performed in valid patch set. $F1_R$ is the road tracking performance using the trained model. "W/wo" stands for the ratio of the mean value of the max outputs group by the patches with roads and without roads inside.

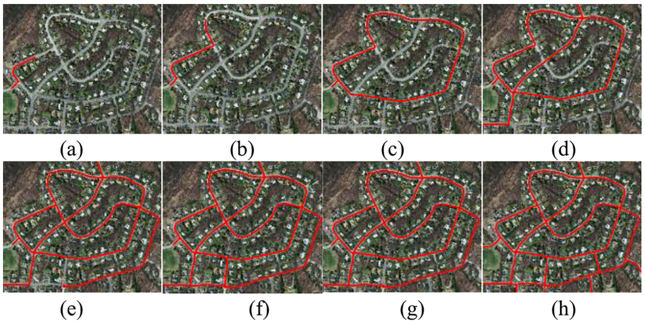


Fig. 12. Evolution of the road network by sliding window. The progress is displayed from left to right and from top to bottom.

Therefore, the picture inside a patch should contain as much information as possible about the road itself and the background around the road, but the optimal ratio of road and background in a patch needs to be verified by experiments. To this end, we fixed the spatial resolution of the image and verified the F1-score and the road tracking performance of the model by changing the size of the patch on Google Earth road set, because the data set has accurate road segmentation ground truth, which is shown in Table II. It should be noted that all the scores in Table II are achieved by the models that are all trained by automatically sampled point sets with the same setting except for the patch size, e.g., 1000 epochs, 32 min-batch size, and 1e-5 learning rate. As can be seen from Table II, The ratio of the road in the patch must be appropriate to highlight the features of the road, so that the model can better determine the existence of road (see w/wo index) and infer the position of the road center (see F1-score index). Higher w/wo means the model has a better distinction of road and background, and higher F1-score denotes the better performance of road center prediction. But, larger patch size means more road segments in a patch, although the model can identify the road center points in the patch, our algorithm is agnostic to the connection of these road center points in the absence of the prior of the road map, the simple straight connection introduces a lot of FPs, which leads to low $F1_R$ index. To sum up, the size of the patch should consider both the density of the road in the image and the width of the road itself. Generally, in practice, we simply set the patch size to 5–7 times the road width for convenience.

V. COMPARISON

A. Comparing Algorithms

To verify the performance, the proposed method is compared with three state-of-the-art road extraction methods. The basic information about these methods is summarized as follows.

1) Ours(acc): The CNN model is trained on the points automatically calculated with the accurate center position.
2) Ours($\sigma = x$): The CNN model is trained on the points automatically calculated with deviation subject to normal distribution parameterized by σ.
3) Ours(manual): The CNN model is trained on the manually sampled points.
4) CasNet: Cheng *et al.* [38] proposed a cascaded end-to-end convolutional neural network to simultaneously cope with the road detection and centerline extraction tasks. In their method, one UNet-like network was used for road detection followed by another UNet-like network for road centerlines extraction.
5) Ventura: Ventura *et al.* [2] designed a CNN model that predicts the local connectivity among the central pixel of an input patch and its border points. The global topology of the road network was inferred by iterating this local connectivity prediction guided by the global road segmentation.
6) ASPP-UNet: He *et al.* [16] improved the standard UNet by integrating ASPP, in which a structural similarity loss was combined for the first time with the BCE loss to train the network.

It should be noted that we take the output of the second network in CasNet and the skeleton of the road area predicted by ASPP-UNet as the final results.

B. Evaluation Metrics

First, we use the F1 metric, a classic measure that evaluates the precision and recall of segmentation. In our theme, the precision represents the ratio between the number of pixels correctly tracked as road and the total number of pixels are tracked as roads. The recall refers to the fraction between the pixels correctly detected as road and the total pixels labeled as a road in the ground truth, and the F1 is the tradeoff of precision and the recall. Inspired by Ventura *et al.* [2], we further evaluate the connectivity performance, and we also use the F1 measure combining precision and connectivity, because high connectivity can be obtained by classifying all pixels into the road. The tradeoff measure can prevent this from happening [2]. For the rest of the paper, $F1_C$ represents the F1 measure of precision and connectivity, and $F1_R$ stands for the F1 measure of precision and recall.

C. Comparison of Road Extraction Algorithms

According to the description of tracking algorithm in Section III-D, we carried out experiments on the Massachusetts road set and Google Earth using the CNN models trained by the different point datasets which are mentioned in Section IV-D. Fig. 12 demonstrates the tracking evolution of our algorithm. Fig. 13 shows some qualitative results in comparison with

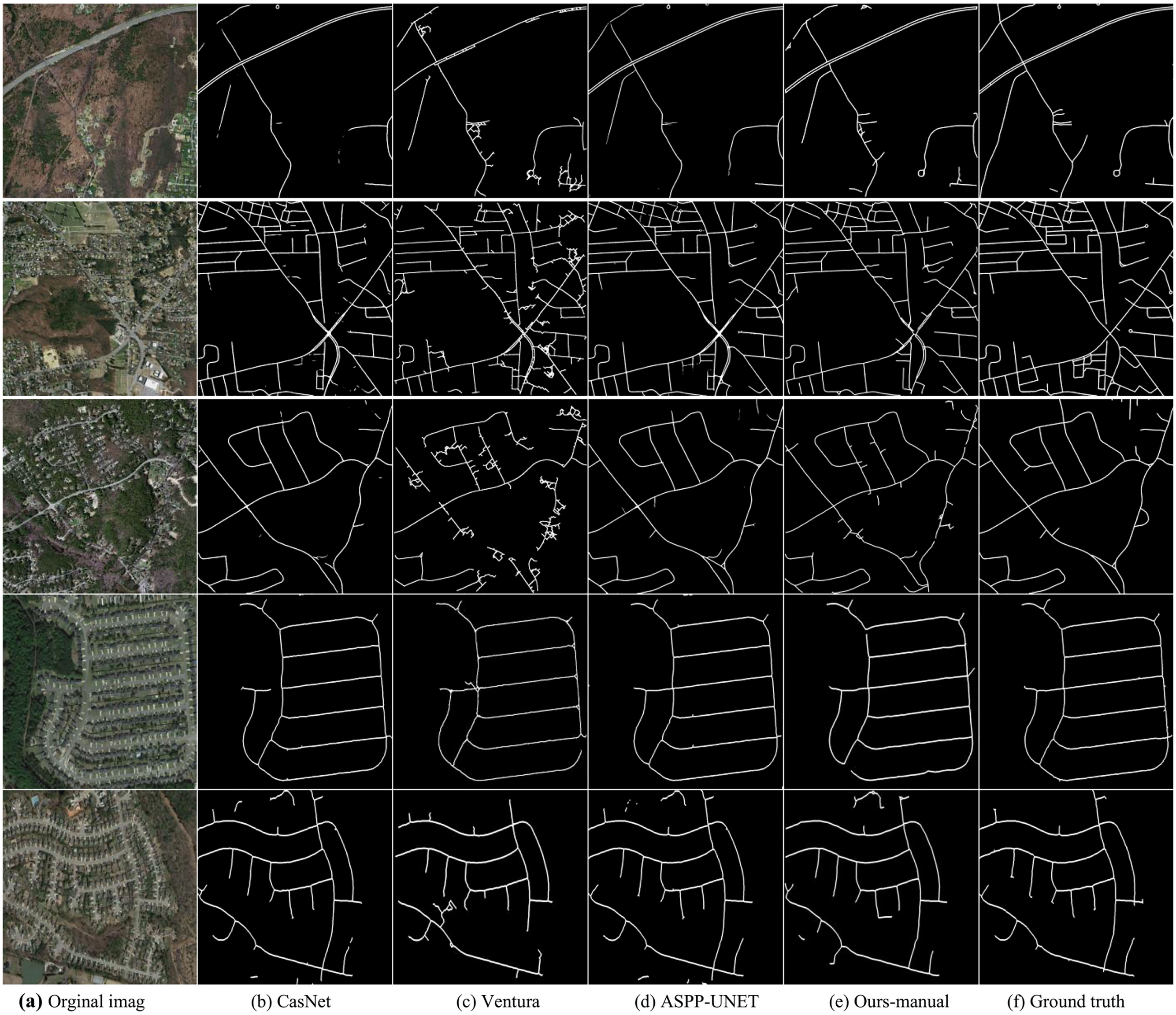


Fig. 13. Qualitative comparisons of road extraction results with different comparing algorithms. The first three rows are from the Massachusetts road dataset, the last two rows are from Google Earth. (a) Original image. (b) CasNet. (c) Ventura. (d) ASPP-UNET. (e) Ours-manual. (f) Ground truth.

other state-of-the-art methods. Though CasNet has two cascaded UNet, the final road centerlines are heavily dependent on the segmentation of the first network. Moreover, the standard UNet does not work well in such a complex situation as the Massachusetts road set, which results in more missing of the final road centerline. The ASPP-UNet achieves much better results taking advantage of ASPP's ability to extract multiscale features. Ventura *et al*. introduce much more FPs because it connects multiple points detected in a small piece, and these points maybe not really connected, which is especially worse in the parallel roads. On the other hand, the result of Ventura depends on the segmentation of VGG net, which does not perform well in the Massachusetts road set, and results in some missing in the complex position, e.g., covered by trees. Our algorithm obtains similar results compared to the best method in most situations. However, as can be seen from Fig. 14, our algorithm performs worse at certain intersections with large spacing, because at these positions, the large road ratio in the patch weakens the road features and makes our model regard these places as parking lots or the roofs of large buildings, which is why our algorithm has a low connectivity index.

To evaluate the effectiveness of the DeepWindow on road extraction, quantitative comparisons with the other three state-of-the-art methods are summarized in Table III. As can be seen from Table III, we conduct comparative experiments on two databases and achieve competitive performances compared to the state-of-the-art methods. To better illustrate the robustness of our model, we also evaluate the quantitative performances of the models trained by different accuracy point datasets, including manually sampled point datasets. It can be seen that the performance decreases as the accuracy of the data decreases, but the reduction is marginal, which means our method is not sensitive to the accuracy of the points. The most important is that our method performs best when the training dataset is

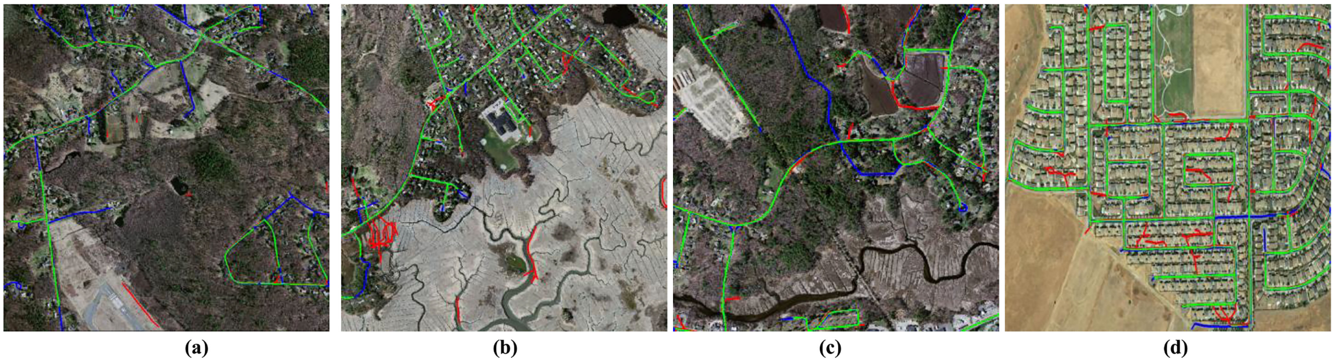


Fig. 14. Some false examples, where green, red, and blue represent correct, error, and missing, respectively.

TABLE III
QUANTITATIVE COMPARISONS AMONG DIFFERENT METHODS ON ROAD CENTERLINES EXTRACTION, WHERE THE VALUES IN BOLD ARE THE BEST AND THE VALUES UNDERLINED ARE THE SECOND BEST

DS	Method	P	R	C	$F1_R$	$F1_C$	Super vision
Massachusetts	CasNet	85.6	79.3	65.3	82.3	74.1	Mask
	Ventura	77.6	80.5	70.5	79.0	73.9	Mask
	ASPP-UNet	**86.3**	**83.1**	**72.6**	**84.7**	**78.9**	Mask
	Ours(acc)	81.2	82.5	70.1	81.8	75.2	Points
	Ours (σ=1.0)	80.5	82.1	69.9	81.3	74.8	
	Ours (σ=2.0)	79.3	81.5	68.7	80.4	73.6	
	Ours (σ=4.0)	78.4	80.8	68.2	79.6	72.9	
	Ours(manual)	82.3	82.7	70.4	82.5	75.9	
Goolg Earth	CasNet	94.7	95.6	95.1	95.9	94.9	Mask
	Ventura	90.4	94.1	93.6	92.2	92.0	Mask
	ASPP-UNet	**96.3**	**95.9**	**97.6**	**96.1**	**97.0**	Mask
	Ours(acc)	92.6	93.8	90.8	93.2	91.7	Points
	Ours (σ=1.0)	91.7	92.9	91.2	92.3	91.4	
	Ours (σ=2.0)	91.4	92.3	90.7	91.8	91.0	
	Ours (σ=4.0)	90.2	91.3	90.1	90.7	90.1	

manually sampled, which guarantees the practical feasibility of our method. Although our algorithm does not achieve the best results compared with the fully supervised semantic segmentation algorithms, but the gap is small. The advantage of our method is that we only need to train a road central point estimation model using weak annotations, which greatly reduces the training cost and extracts the road network directly from images without the auxiliary information such as road segmentation.

However, our method also has disadvantages, e.g., it fails to extract the roads when they are seriously covered by trees or mistakes road-like objects as roads because the information inside a small patch is the only basis of our judgment. Fig. 14 shows some false examples where the roads seriously covered by trees are missing and some road-like objects are wrongly traced limited by the small receptive field of the sliding window.

VI. CONCLUSION

In this article, a patch-based road central point estimation model is proposed for the prediction of the road central points in a patch. Based on the model, we present a fully automatic road network tracking method in a sliding window mode. Most importantly, the proposed method discards the guidance of the global road segmentation and, thus, our method is free from the large amount of pixel-wise annotations that are required by the training of the semantic segmentation model. Moreover, for training the road central point estimation model, we only need to point out the central point of each road segment in the training patches, which greatly reduces the labeling cost. Experiments show that our method can accurately track the road centerlines even if they are interfered by noise. The tracking process of our method needs a certain amount of time unlike the methods based on the semantic segmentation model which output the global road segmentation in a flash; however, the mode of the point supervised training combining the iterative tracking process provides a practicable scheme for the weakly supervised training of the semantic segmentation model for road extraction. Finally, our method fails to extract the road segments when they are fully covered by noise, which will be improved in future studies.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their valuable comments.

REFERENCES

[1] F. Bastani *et al.*, "RoadTracer: Automatic extraction of road networks from aerial images," in *Proc. IEEE/CVF Conf. Comput. Vision Pattern Recognit.*, 2018, pp. 4720–4728.

[2] C. Ventura *et al.*, "Iterative deep learning for network topology extraction," in *Proc. British Mach. Vision Conf.*, 2018.

[3] S.-R. Park and T. Kim, "Semi-auto road extraction algorithm from IKONOS images using template matching," in *Proc. 22nd Asia Conf. Remote Sens.*, 2001, pp. 1209–1213.

[4] L. Xiangguo *et al.*, "Integration method of profile matching and template matching for road extraction from high resolution remotely sensed imagery," in *Proc. Int. Workshop Earth Observ. Remote Sens. Appl.*, 2008, pp. 1–6.

[5] X. Lin and Z. Liu, "Semi-automatic extraction of ribbon roads from high resolution remotely sensed imagery by T-shaped template matching," *Geomatics Inf. Sci. Wuhan Univ.*, vol. 7147, no. 3, pp. 293–296, 2009.

[6] X. Lin, J. Shen, and Y. Liang, "Semi-automatic road tracking using parallel angular texture signature," *Intell. Automat. Soft Comput.*, vol. 18, no. 8, pp. 1009–1030, 2012.

[7] L. Renbao, W. Weixing, and L. Juan, "Road extraction from high-resolution remote sensing images based on adaptive circular template and saliency map," *Acta Geodaetica Et Cartographica Sinica*,vol. 47, no. 7, pp. 62–70, 2018.

[8] Z. Miao *et al.*, “A semi-automatic method for road centerline extraction from VHR images,” *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 11, pp. 1856–1860, Nov. 2014.

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

[10] C. L. D, “Multiple contour finding and perceptual grouping using minimum paths,” *J. Math. Imag. Vis.*, vol. 14, no. 3, pp. 225–236, 2001.

[11] Z. Miao *et al.*, “Use of GMM and SCMS for accurate road centerline extraction from the classified image,” *J. Sensors*, vol. 2015, 2015, Art. no. 784504.

[12] R. Liu *et al.*, “Road centerlines extraction from high resolution images based on an improved directional segmentation and road probability,” *Neurocomputing*, vol. 212, no. C, pp. 88–95, 2016.

[13] G. Mattyus, W. Luo, and R. Urtasun, “Deeproadmapper: Extracting road topology from aerial images,” in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 3458–3466.

[14] M. Maboudi *et al.*, “Object-based road extraction from satellite images using ant colony optimization,” *Int. J. Remote Sens.*, vol. 38, no. 1, pp. 179–198, 2017.

[15] L. Gao *et al.*, “Road extraction from high-resolution remote sensing imagery using refined deep residual convolutional neural network,” *Remote Sens.*, vol. 11, no. 5, 2019, Art. no. 552.

[16] H. He *et al.*, “Road extraction by using atrous spatial pyramid pooling integrated encoder-decoder network and structural similarity loss,” *Remote Sens.*, vol. 11, no. 9, 2019, Art. no. 1015.

[17] 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.

[18] 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*, 2018, pp. 192–1924.

[19] A. Neubeck and L. V. Gool, “Efficient non-maximum suppression,” in *Proc. Int. Conf. Pattern Recognit.*, 2006, pp. 850–855.

[20] Y. Xu *et al.*, “Road extraction from high-resolution remote sensing imagery using deep learning,” *Remote Sens.*, vol. 10, no. 9, pp. 1461–1476, 2018.

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

[22] M. Rezaee and Y. Zhang. “Road detection using deep neural network in high spatial resolution images,” in *Proc. Joint Urban Remote Sens. Event*, 2017, pp. 1–4.

[23] Z. Zhong *et al.*, “Fully convolutional networks for building and road extraction: Preliminary results,” in *Proc. Geosci. Remote Sens. Symp.*, 2016, pp. 1591–1594.

[24] A. V. Buslaev *et al.*, “Fully convolutional network for automatic road extraction from satellite imagery,” in *CVPR Workshops*, pp. 207–210, 2018.

[25] X. Gao *et al.*, “An end-to-end neural network for road extraction from remote sensing imagery by multiple feature pyramid network,” *IEEE Access*, vol. 6, pp. 39401–39414, 2018.

[26] A. Wulamu *et al.*, “Multiscale road extraction in remote sensing images,” *Comput. Intell. Neurosci.*, vol. 2019, 2019, Art. no. 2373798.

[27] G. Mattyus “Enhancing road maps by parsing aerial images around the world,” in *Proc. IEEE Int. Conf. Comput. Vision*, 2015, pp. 1689–1697.

[28] C. Cao and Y. Sun, “Automatic road centerline extraction from imagery using road GPS data,” *Remote Sens.*, vol. 6, no. 9, pp. 9014–9033, 2014.

[29] U. Ozertem and D. Erdogmus, “Locally defined principal curves and surfaces,” *J. Mach. Learn. Res.*, vol. 12, no. 4, pp. 1249–1286, 2008.

[30] Y. Yang and C. Zhu, “Extracting road centrelines from high-resolution satellite images using active window line segment matching and improved SSDA,” *Int. J. Remote Sens.*, vol. 31, no. 10, pp. 2457–2469, 2010.

[31] T. Kim *et al.*, “Tracking road centerlines from high resolution remote sensing images by least squares correlation matching,” *Photogram. Eng. Remote Sens.*, vol. 70, no. 12, pp. 1417–1422, 2004.

[32] X. Lin *et al.*, “Semi-automatic extraction of road networks by least squares interlaced template matching in urban areas,” *Int. J. Remote Sens.*, vol. 32, no. 17, pp. 4943–4959, 2011.

[33] A. Newell, K. Yang, and J. Deng. “Stacked hourglass networks for human pose estimation,” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 483–499.

[34] L. Xu *et al.*, “Structure extraction from texture via relative total variation,” *ACM Trans. Graph.*, vol. 31, no. 6, 2012, Art. no. 139.

[35] J. Canny, “A variational approach to edge detection,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-8, no. 6, pp. 679–698, 1986.

[36] C. Tomasi and R. Manduchi. “Bilateral filtering for gray and color images,” in *Proc. 6th Int. Conf. Comput. Vis.*, 1998, pp. 839–846.

[37] Massachusetts Roads Dataset. [Online]. Available: https://www.cs.toronto.edu/∼vmnih/data/. Accessed on: Oct. 13, 2019.

[38] G. Cheng *et al.*, “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.

[39] V. Mnih and G. E. Hinton, “Learning to label aerial images from noisy data,” in *Proc. 29th Int. Conf. Mach. Learn.*, 2012, pp. 567–574.

[40] T. Tieleman and G. Hinton, “Lecture 6.5 - rmsprop: Divide the gradient by a running average of its recent magnitude,” *COURSERA: Neural Netw. Mach. Learn.*, vol. 4, pp. 26–31, 2012.

Renbao Lian received the B.S. degree from University of Jinan, Jinan, China, in 2002. He is currently working toward the Ph.D. degree with the Collage of Physics and Information Engineering, Fuzhou University, Fuzhou, China. He is currently an Associate Professor with the Collage of Electronics and Information Science, Fujian Jiangxia University, Fuzhou, China. His research interests include remote sensing image processing, computer vision, and information processing.

Liqin Huang received the Ph.D. degree from Fuzhou University, Fuzhou, China, in 2009. He is currently a Full Professor with the College of Physics and Information Engineering, Fuzhou University. His research interests include image processing, computer vision, artificial intelligence, and computer network communication.