A Combination of Convolutional and Graph Neural Networks for Regularized Road Surface Extraction

Jingjing Yan, Shunping Ji, *Senior Member, IEEE*, and Yao Wei

***Abstract*— Road surface extraction from high-resolution remote sensing images has many engineering applications; however, extracting regularized and smooth road surface maps that reach the human delineation level is a very challenging task, and substantial and time-consuming manual work is usually unavoidable. In this article, to solve this problem, we propose a novel regularized road surface extraction framework by introducing a graph neural network (GNN) for processing the road graph that is preconstructed from the easily accessible road centerlines. The proposed framework formulates the road surface extraction problem as two-sided width inference of the road graph and consists of a convolutional neural network (CNN)-based feature extractor and a GNN model for vertex attribute adjustment. The CNN extracts the high-level abstract features of each vertex in the graph as the input of the GNN and also the road boundary features that allow us to distinguish roads from the background. The GNN propagates and aggregates the features of the vertices in the graph to achieve global optimization of the regression of the regularized widths of the vertices. At the same time, a biased centerline map can also be corrected based on the width prediction result. To the best of the authors' knowledge, this is the first study to have introduced a GNN to regularized human-level road surface extraction. The proposed method was evaluated on four diverse datasets, and the results show that the proposed method comprehensively outperforms the recent CNN-based segmentation methods and other regularization methods in the intersection over union (IoU) and smoothness score, and a visual check shows that a majority of the prediction results of the proposed method approach the human delineation level.**

***Index Terms*— Convolutional neural network (CNN), graph neural network (GNN), regularization, road extraction.**

I. Introduction

ROAD extraction from high-resolution satellite or aerial images has been a hot research topic in the field of remote sensing image processing over the past decades, and accurate and regularized road maps have important and broad applications in city planning, geographic information updating, vehicle navigation, and autonomous driving. However, the construction and updating of road surface maps is still a time-consuming and labor-intensive task even though the rapid development of Earth observation technology and automated machine learning methods has been witnessed. During the past few decades, deep learning has achieved excellent results in semantic segmentation [1]–[4], object detection [5]–[9], and other vision-based fields, and a variety of methods, especially deep learning-based semantic segmentation methods, have been proposed to extract road surface maps from high-resolution remote sensing images. These methods have made considerable advances, but they also suffer from occlusions, noise, and the complexity of the background in remote sensing images. At present, the fully automated delineation of accurate, regularized, and manual-level road vector maps is still difficult to achieve. The up-to-date representative road extraction methods are based on deep learning, and the first attempt at deep learning-based road extraction can be traced back to 2010 [10]. However, a decade of development has mainly focused on the development of specific convolutional neural network (CNN) structures for pixel-level road semantic segmentation [11]–[22], and how to regularize the extracted road surface maps to reach human-level and vector-based delineation has not been tackled in depth. In this article, we propose a practical regularized road surface map extraction method based on a combined CNN and graph neural network (GNN) and the aid of road centerline maps, aiming to replace the use of human labor by directly predicting regularized and smooth double-line road vector maps. In this section, we review the development of the recent deep learning-based road extraction methods and then propose our novel approach for human-level road surface extraction.

Over the last decade, road surface extraction has been widely studied, and the classic machine learning and CNN-based methods have been gradually introduced to the field. Mnih and Hinton [10] proposed a method employing restricted Boltzmann machines to segment roads from high-resolution aerial images, in which unsupervised pretraining and supervised postprocessing were introduced to improve the performance. Das *et al.* [23] exploited two salient features of roads and proposed a multistage framework to extract roads from high-resolution multispectral satellite images using four probabilistic support vector machine (SVM) models. Saito *et al.* [11] combined the textural and spectral parameters and input them into an artificial neural network (ANN) to extract road surfaces from satellite images and employed a CNN to extract buildings and roads simultaneously from

Manuscript received September 17, 2021; revised November 10, 2021 and December 30, 2021; accepted February 10, 2022. Date of publication February 15, 2022; date of current version March 24, 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 42171430 and in part by the State Key Program of the National Natural Science Foundation of China under Grant 42030102. *(Corresponding author: Shunping Ji.)*

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

Yao Wei is with the Faculty of Geo-Information Science and Earth Observation, University of Twente, 7500 AE Enschede, The Netherlands (e-mail: yao.wei@utwente.nl).

Digital Object Identifier 10.1109/TGRS.2022.3151688

remote sensing imagery. Panboonyuen *et al.* [12] presented an enhanced deep CNN framework to extract road objects from aerial and satellite images, and they applied landscape metrics and a conditional random field (CRF) model to further improve the detection results. Zhang *et al.* [13] proposed an architecture named ResUnet for road extraction, which combines the strengths of residual learning and the U-Net architecture to ease the training burden. Zhou *et al.* [14] proposed a segmentation network named D-LinkNet to extract road surfaces in high-resolution satellite imagery. This method is built on the LinkNet [24] architecture and has dilated convolutional layers in its center part. Yang *et al.* [15] designed a recurrent CNN (RCNN) unit and incorporated it into the U-Net architecture to simultaneously achieve the tasks of road detection and centerline extraction. Zhang *et al.* [16] proposed an end-to-end road extraction framework based on an improved generative adversarial network (GAN). Shamsolmoali *et al.* [17] introduced adversarial networks and domain adaptation for road segmentation. Wei *et al.* [18] proposed a multistage framework, which consists of boosting segmentation, multiple starting point tracing, and fusion of the segmentation and tracing results, to extract road surfaces and road centerlines simultaneously. Similarly, such coarse-to-fine or multitask scheme has also been used in [19] and [20]. Wang *et al.* [21] proposed the nonlocal LinkNet model with nonlocal blocks in the encoder part to segment road surfaces from VHR satellite images. Zhou *et al.* [22] proposed a network that embeds a universal iteration reinforcement module to enhance the learning ability.

These methods have boosted the performance of road surface extraction; however, in complex scenes, they usually produce poor connectivity and incomplete and irregular results that are far from the human delineation level, due to the occlusion of trees, buildings, and shadows, and the ambiguity between roads and background in remote sensing images. Although some approaches [10], [12] have attempted to alleviate these shortcomings by postprocessing after detection, they are generally ineffective as the heuristic rules used in these methods cannot adapt to the wide variety of roads. In contrast, our motivation is different from the aforementioned pixel-based image processing methods, in which we attempt to realize smooth, continuous, and regularized road surface extraction to save on human labor. Starting from this practical viewpoint, we focus on two aspects.

One aspect is to gather and utilize the other available or easily accessible road-relevant data as a supplement, in addition to the information provided by the images. Road centerline data can be obtained in a variety of ways, such as volunteered geographic information (VGI) and historical road centerline data. In particular, the emergence of VGI has allowed users to collect, edit, and share geographic information around the world, making the free and large-scale acquisition of road centerline data simple, easy, and convenient. As one of the most extensive VGI data sources, OpenStreetMap (OSM) [25] now covers a majority of the world, with the quality and quantity of the annotation growing over time as more and more contributors join in. In addition, a lot of historical centerline road data are available in the GIS maps provided by departments of surveying and mapping. However, the information in these two types of data is often of low quality and the centerline coordinates often deviate from the actual road centerlines in newly accessed remote sensing images due to a variety of factors, including outdated maps, geometric rectification errors, and different levels of details of manual delineation.

The other aspect is to develop an advanced regularization method at the theoretical level to approach the delineation of the human level. There have been a few studies of this aspect in the past. For example, Máttyus *et al.* [26] framed the road surface extraction task as width inference of each vertex in the road centerline graph, and they regularized the road extraction result by developing a Markov random field (MRF) model that encodes the different image features, such as edges, the pixel intensity, and the smoothness of the road, to postprocess the output from the local classifier. However, on the one hand, each of these features needs domain knowledge and is data-specific; on the other hand, the global postprocessing can only use the local classifier's prediction as the input, and it overlooks the spatial correlation of the road features. We are not currently aware of any deep learning-based methods for the task of road regularization. In this study, we attempted to simultaneously extract and regularize road surfaces through the use of a GNN.

Since non-Euclidean graph data, such as traffic network, social network, and biological network data, cannot be processed effectively by a traditional CNN, the GNN architecture [27] was proposed and has been gradually developed. Over the past decade, a great number of GNN variants [28]–[32] have been proposed. Due to the strong power of the modeling relationships between the vertices in a graph, GNNs have been successfully applied to many computer vision tasks, such as point cloud classification [33], [34], action recognition [35], and object boundary extraction [36]. In the field of remote sensing image processing, GNNs have been mainly applied to boundary extraction and attribute prediction. For example, Ling *et al.* [36] utilized a graph convolutional network (GCN) for object instance segmentation, where they represented an object instance as a graph and continuously adjusted the position of the vertices in the graph through the GCN to predict the boundaries of the objects. He *et al.* [37] applied a GNN to infer the number of lanes and the road type from satellite imagery. Jepsen *et al.* [38] designed a GCN architecture for road attribute inference, for attributes such as the speed limit. However, to date, we are not aware of any research that has attempted to apply a GNN to the task of regularized road surface extraction.

In this study, inspired by the recent development of GNNs and the easily accessible road centerline information, we introduced a GNN to regularized road boundary regression for the first time and developed a novel end-to-end combined CNN and GNN framework for regularized and human delineation-level road surface extraction from remote sensing images. Furthermore, the method can also correct the offsets of the biased centerlines in new remote sensing images.

The main contributions of our work are summarized as follows.

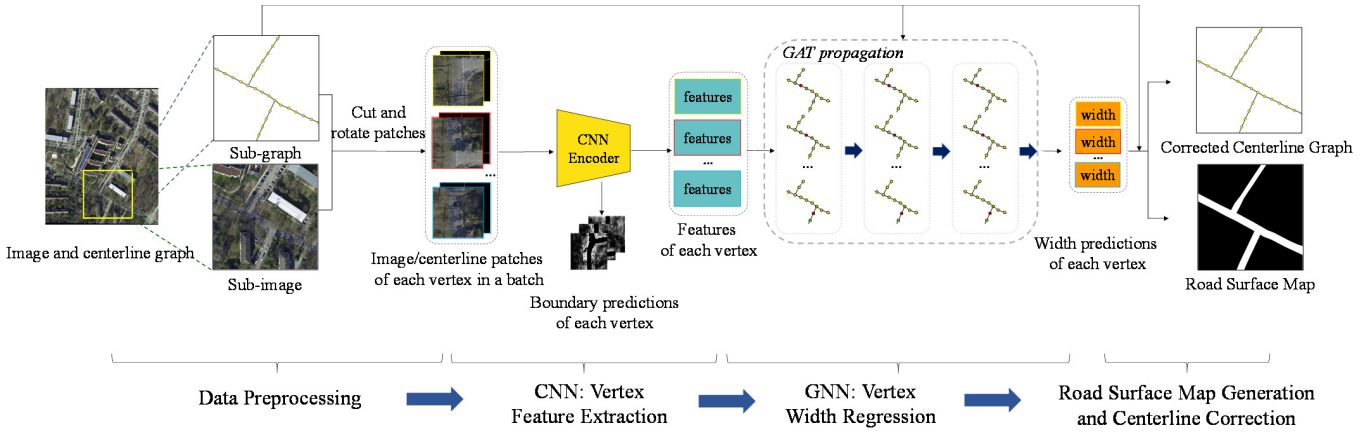


Fig. 1. Flowchart of the proposed framework for road surface extraction and centerline correction.

1) A new framework is proposed for regularized road surface extraction (and centerline correction) from remote sensing images and the available road centerline data. The proposed framework formulates the road surface extraction task as width (as an attribute of the vertices) inference in a graph that is prebuilt from the road centerline data. To the best of our knowledge, this is the first framework for regularized road surface extraction based on a GNN.
2) The framework consists of a novel combination of a CNN and a GNN. The front-end CNN extracts both the high-level abstract features for the subsequent GNN regression step and the boundary feature of each vertex for recognizing roads from backgrounds. The back-end GNN propagates and aggregates these high-level features of the vertices in the graph to achieve global optimization of the road width prediction, to achieve regularized road surface extraction.
3) Centerline offset correction is also simultaneously achieved by postprocessing of the two-sided width predictions of each road vertex. In our experiments, the average distance between the corrected centerline and the ground-truth centerline was 1.4 pixels, which is very small and reaches the human delineation level.
4) The proposed method was thoroughly evaluated on four different and versatile datasets, which confirmed that the proposed method can comprehensively outperform the existing segmentation-based methods, regularization-based methods, and the other GNN-based methods in the intersection over union (IoU), F1 score, and road smoothness indicator. In particular, the excellent performance in the road smoothness metric that we designed to measure the continuous, structured, and human-level road boundaries shows that the proposed method is the most suitable method for extracting a regularized road surface map.

The rest of this article is arranged as follows. In Section II, we present the details of the proposed method. Section III provides a detailed description of the datasets, metrics, and the experimental evaluation of the road surface extraction. We also evaluate the performance of the different GNN modules, propagation rules, different orders of neighborhood in the GNN aggregation, and the effectiveness of the offset centerline correction in this section. Finally, our conclusions are outlined in Section IV.

II. Methods

A. Framework

We propose a novel framework for regularized road surface extraction from remote sensing imagery. Taking the easily accessible road centerline data as a prerequisite, we formulate the road surface extraction problem as the regression of the piecewise and smooth widths of both sides of the roadway in a combined CNN and GNN framework. Furthermore, the output of the network can also naturally correct the possible offset of the centerlines at the same time.

The framework of the proposed method is shown in Fig. 1. The preprocessing includes three steps conducted at three levels of detail. First, we handle the original large-capacity remote sensing image to preserve the complete road topological information, instead of precropping the image into fixed small patches. The key is to generate a road graph from the road centerline map that has been geographically aligned with the remote sensing image by discretizing the centerline with a set of equidistant vertices. Each vertex has a direction property that represents the heading of the road. Second, we process n (a preset number) vertices that consist of a subgraph in a batch, according to the capacity of the GPU memory. The subgraph is dynamically generated from a current vertex. The current vertex's $n-1$ neighbor vertices are determined with depth- or breadth-first strategies in the graph. The next current vertex skips these processed vertices (including the neighboring vertices) in the training stage to rapidly cover the whole graph. Third, we handle each vertex in a subgraph. We crop a patch (e.g., 256×256 pixels) from the aligned and concatenated image and centerline map centered at a vertex position, all of which are uniformly rotated to the vertical direction (i.e., the road direction at the current vertex is upward) to form the individual inputs of the CNN encoder.

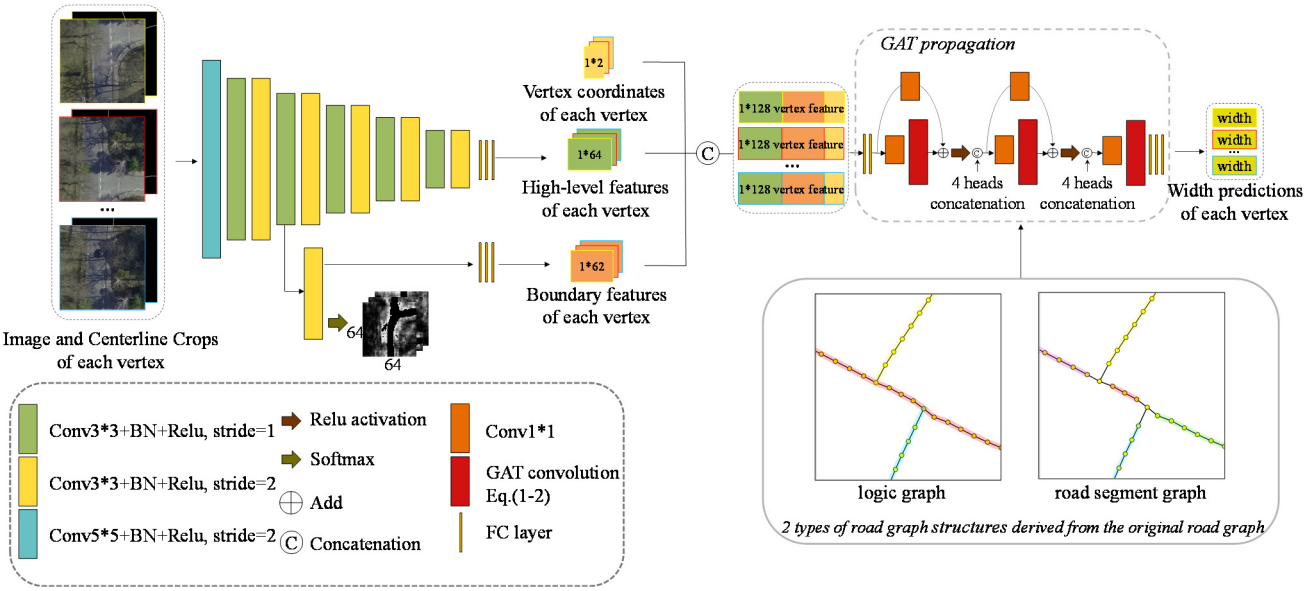


Fig. 2. Structure of the proposed network.

A lightweight CNN encoder is applied to extract the features of the vertices from each patch. We also introduce a road boundary extraction branch to help the CNN encoder concentrate on the roads rather than the background. All of the vertex features in the subgraph are input into the GNN, in which the information of the vertices is propagated within the subgraph and exchanged in the dynamically constructed subgraphs to obtain the global representation. The position attribute of the vertices is then adjusted, to finally output the two-sided widths of the vertices.

Finally, the complete road surface is directly produced by connecting the boundary points of all the vertices without any postprocessing, with the irregular vertices having been smoothed by the GNN, and the centerline is corrected through the two-sided widths of each vertex.

B. Network Structure

We propose a novel network structure combining a CNN encoder and a GNN model to extract the two-sided widths of each vertex in a road graph from the remote sensing image and the corresponding road centerlines. In the first stage, the CNN encoder extracts both the high-level features of each vertex and the road boundary information. The road boundary branch helps the model distinguish roads from the background and initially locates the boundary of the road surface so that the vertices on the boundaries can be recognized. The high-level features represent the information of a vertex in the graph as the input of the GNN. The GNN adjusts the width attribute of each vertex. It updates the state of the current vertex by aggregating its neighborhood information, propagates the information to the adjacent vertices, and integrates them within the subgraph to capture the spatial and topological correlation. The full information exchange eliminates the information communication obstacles existing in the common local extractors, reduces the irregularities of the road surface map caused by the occlusions (e.g., buildings, shadows, and trees) in the remote sensing image, and finally constructs a regularized road surface map that reaches the level of human-level delineation.

The details of the proposed network are shown in Fig. 2. The CNN extractor consists of 11 convolutional layers and three fully connected (FC) layers. The extractor layerwise abstracts the aligned and concatenated RGB image patch and rasterized centerline map to form a 64-D feature vector that represents the high-level features of each vertex. Meanwhile, a 3×3 convolutional layer is employed to process the result of the fourth convolutional layer, to output a coarse road boundary feature map, which we use to extract the boundary information. A subsequent soft-max operation obtains a road boundary segmentation probability map, which is supervised by the road boundary map. The boundary information is then compressed to a 62-D vector from the road boundary feature map through the three FC layers. The position of the vertex (x and y) in the cropped image, the 64-D vertex feature vector, and the 62-D road boundary feature vector are concatenated to make up a final and comprehensive 128-D feature, as the input of the next stage of GNN adjustment.

The GNN module is designed to have two distinct functions. First, it should adjust the position of each vertex twice, according to the 128-D feature input; in other words, it should adjust the vertex to both the left- and right-hand intersection points of the road boundaries and the perpendicular line through the current vertex, thereby creating a double-line road surface without the impact of possible centerline bias. To solve this problem, we send the vertex to the GNN module to simultaneously output the two vertex positions (four parameters). As each patch has been rotated to the upward direction of the road segment, the module can easily distinguish left and right.

Second, the GNN module should consider how to propagate the current vertex's information to the other adjacent vertices in a subgraph to achieve a smooth and structured road centerline by utilizing the complete information of a

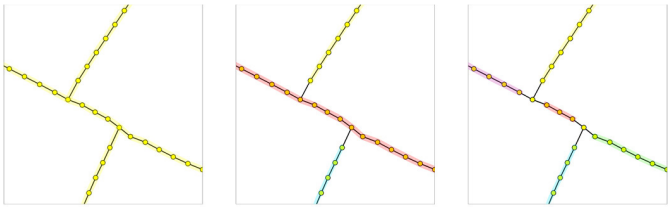


Fig. 3. Three forms of vertex propagation rules. (a) Vertex information can be propagated to all the other vertices. (b) Vertex information can only be propagated along its current road. (c) Propagation is restricted between two adjacent intersection/endpoint vertices.

batch. To solve this problem, we designed a two-step strategy. First, we define the neighborhood of the current vertex for updating its state. We use the vertices' information in the first-order neighborhood ($N1$) to infer the left- and right-hand widths of the current vertex. The number of neighborhood points can be 1 (starting vertex), 2 (vertex in a straight road), or n (intersection vertex with n branches). The second-order neighborhood ($N2$) can also be used to obtain similar results. However, $N2$ is more complex and takes more running time and memory, which is discussed in Section III-F. Second, we define the propagation rule, i.e., how the information of the current vertex is propagated within the subgraph. We assume that there are three types of possible message propagation, according to the topological property of the road network. The first (Type A) is when the information of a vertex can be unrestrictedly propagated to all of the other vertices [Fig. 3(a)]. The second (Type B) is when the current vertex (we ignore the intersection vertices) can only be propagated along its current road [Fig. 3(b)]. The current road is defined so that the turn angle in this road is smaller than a given angle (e.g., 30°). The third (Type C) is when the message propagation of the vertex is restricted between two adjacent intersection/endpoint vertices [Fig. 3(c)], which is a supplement for Type B as the road segments in a complete road may have different widths to be inferred. Type A is a default setting in most GNN-based applications, and however, it ignores that different roads or road segments probably have different widths; in contrast, the other two types adapt to practical road vertex message propagation as they enable the GNN to learn specific road attributes by guaranteeing that the different types of roads do not interfere with each other. Therefore, we apply the second and third propagation forms separately, to generate the feature vectors. These are then concatenated as the output feature of the GNN model.

The last thing is to choose a suitable GNN module that operates on the current vertex and its neighborhood. The typical problem for a segmentation network is inferred abnormal points that result in irregular boundaries. In the GNN, we need to introduce a specific mechanism that can weaken the impact of abnormal neighbors (specifically, the occlusion effects on the image) during the propagation, to achieve more regularized and smoother results. We selected a graph attention network (GAT) module [32], which introduces an attention mechanism to graph convolution for estimating the contributions of neighboring vertices and the current vertex by introducing a learnable weight parameter instead of setting a fixed equal weight. We also tried other GNN network modules, such as a GCN [28] and a gated graph neural network (GGNN) [29], and found that the GAT module is the most suitable network module (this is shown in detail in the experiments).

The GAT module calculates the hidden state of each vertex with the weighted average of the neighboring features by using the self-attention mechanism. The hidden state at propagation step t of vertex v is denoted as $h_u(t)$, where $h_u(0)$ is the 128-D features extracted by the CNN, as the original input of the GAT module. The convolutional layer of the GAT module can be denoted as

$$\alpha_{vu}^{(t)} = \mathrm{softmax}\left(g\left(a^T\left(W^{(t)}h_v^{(t-1)} \middle\| W^{(t)}h_u^{(t-1)}\right)\right)\right) \tag{1}$$

$$h_v^{(t)} = \sigma\left(\sum_{u\in N(v)\cup v} \alpha_{vu}^{(t)} W^{(t)} h_u^{(t-1)}\right) \tag{2}$$

where $\alpha_{vu}^{(t)}$ denotes the attention weight of vertex v over its neighbor u, W denotes the trainable weight matrix of a shared linear transformation, which is applied to every vertex, $N(v)$ denotes the neighborhood of vertex v in the graph (the first order of neighborhood $N1$ is applied in the experiments), $g(\cdot)$ is a LeakyReLU activation function, and a is a vector of the learnable parameters. The GAT module utilizes multihead attention, which is implemented by applying several independent attention mechanisms to compute the hidden states and then concatenating or averaging their features. We refer the reader to [32] for more details.

In the GNN model, we use three GAT layers for information exchange. The first layer has four attention heads and transforms the 128-D features into 64 dimensions in each head, and the second layer has four attention heads and transforms the concatenated 64 × 4-D features into eight dimensions each. In the last layer, there is only one attention head module, and the final 4-D outputs (which contain the coordinates of both sides of the boundary points) are extracted from the 8-D features. We can then obtain the widths of both sides of each vertex by the vertex position and the network output.

The loss function of the proposed network consists of two parts. The first part is the mean square loss of width prediction, which is implemented by calculating the difference between the ground-truth vertex (the position x and y) and the regressed vertex and is defined as

$$l_{\text{width}} = \frac{1}{4}\sum_{k=1}^{4}\left\|W_k - W_k'\right\|^2 \tag{3}$$

where W refers to the predicted 4-D output (left- and right-hand positions) and W' refers to the ground-truth vertex.

The second part is the mean square loss for the boundary branch, which calculates the pixel-level difference between the predicted coarse boundary map and the ground truth, and is defined as

$$l_{\text{boundary}} = \frac{1}{w\times h}\sum_{i=1}^{w}\sum_{j=1}^{h}\left\|Y_{ij} - Y_{ij}'\right\|^2 \tag{4}$$

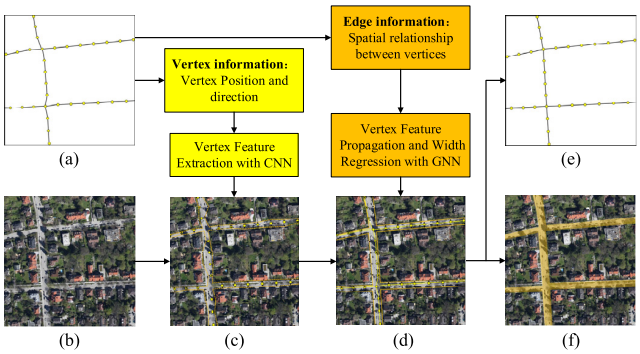


Fig. 4. Process of generating a road surface map and corrected centerline. (a) Road graph. (b) Image. (c) CNN result. (d) GNN result. (e) Corrected road centerline. (f) Road surface.

where Y is the predicted coarse boundary map, Y' is the ground truth, and w and h represent the width and height of the boundary map, respectively.

The complete loss function is then defined as

$$L = l_{\text{width}} + \lambda l_{\text{boundary}} \tag{5}$$

where λ is the weight of the boundary loss, and it was empirically set to 100 in our experiments.

In the training process, we first train only the CNN by adding an FC layer after the 128-D features of each vertex to obtain a 4-D vector as the output and then train the combination of the CNN and GNN to avoid overfitting.

C. Road Surface Map Generation

The process of generating the final road surface map is shown in Fig. 4. With the vertex information from the road graph [Fig. 4(a)] and image [Fig. 4(b)] as inputs, the CNN obtains rough width results for each vertex, and their connections form the two sides of the road surface [Fig. 4(c)]. The GNN further processes the features of vertices and adjusts the width of each vertex in the graph [Fig. 4(d)], to achieve a smooth and human delineation-level road surface map. Note that the CNN is trained first and then the whole network. Obviously, the result of our method [Fig. 4(d) and (f)] is essentially a vector map, which is different from a pixel-level segmentation map obtained from the mainstream methods. It should be noted here that it is difficult and also unnecessary to determine the ambiguous road widths of the intersection points. Instead, we do not calculate their widths during the training and testing process and instead obtain the road width and directions according to the adjacent vertices.

We can also obtain more accurate centerlines by averaging the widths of each side of the roadway. As shown in Fig. 4(e), each vertex in the corrected centerline is located at the center of the road. This function can also be applied to correct offset centerline data. Fig. 4(e) also shows that the proposed method is free from bias in the centerlines. The proposed method can obtain accurate and smooth road maps with both high-precision or biased centerlines, which is further discussed in Section III.

III. Experiments and Analysis

A. Datasets

We performed experiments on four diverse datasets: Bavaria [26], Aerial KITTI [26], Shaoxing, and Wuhan. We also simulated biased centerline data with the Aerial KITTI dataset to test the robustness of the algorithm. All of these datasets contain high-resolution aerial or satellite images and the corresponding segmentation ground truth, with variable road widths.

The open-source Bavaria dataset contains aerial images with a 13-cm ground resolution and the corresponding road segmentation annotations. This dataset was captured in the Bavaria region of Germany and covers urban, suburban, and rural areas. The total area is 4.95 km^2. There are 12 regions in this dataset. We used nine regions for the training and the other three for the testing.

The open-source Aerial KITTI dataset consists of aerial images downloaded from Google Earth Pro and the corresponding segmentation ground truth for the city of Karlsruhe, Germany. The total area is 5.96 km^2 and the ground resolution is 13 cm/pixel. There are 21 images in the Aerial KITTI dataset, and we used 16 images for the training and the rest for the testing.

To compensate for the small open-source datasets, we prepared two large datasets. The Shaoxing dataset we built contains six regions with 532 aerial images of 1024 $\times$ 1024 pixels and the corresponding ground truth, with a 0.6-m ground resolution. This dataset covers 200.82 km^2 of Shaoxing, which is a typical water city in China. We used four regions with 404 images for the training and the two other regions with 128 images for the testing.

The Wuhan dataset we built contains eight 8192 $\times$ 8192 satellite images and the corresponding segmentation annotations. The ground resolution of the images is 0.5 m/pixel and the total area is 134.22 km^2. In the experiments, six images were used for the training and the other two were used for the testing.

For the datasets not providing the centerlines, we obtained the centerline map by skeletonizing the pixel-level segmentation annotations. For each complete image in these datasets, we created the corresponding road graph by sampling vertices with equal distance in each centerline map. The sampling interval in the Aerial KITTI and Bavaria datasets was 12.5 m, and the sampling interval was 25 m in the Shaoxing and Wuhan datasets, according to their ground resolutions.

The setting of the road graph consists of two steps. First, we set the directions of all the vertices in the centerline map. We divided the vertices into three types—endpoints, midpoints, and intersection points—according to their degree of connection. For an endpoint, its direction is defined by itself and its only neighbor, for a midpoint, its direction is defined by its two neighbors, and an intersection point has no direction. Second, the vertical line passing through a vertex intersects with the boundaries of the road to create the left and right widths of the vertex. For an intersection vertex, the width is calculated by averaging the widths of its neighboring vertices.

To evaluate the method performance on an offset dataset, we also produced an analog offset dataset from the Aerial KITTI dataset. We randomly moved the vertices toward the vertical direction based on the original nonoffset centerline with random offsets (the offset was set the same within a road segment). The offset was less than one-half of the shortest width of the road section.

B. Evaluation Metrics and Settings

To evaluate the road surface extraction results at the pixel level, the precision, recall, F1 score, and IoU are introduced. The precision is the fraction of the predicted road pixels that are true road pixels, and the recall is the fraction of all the true road pixels that are correctly predicted. The F1 and IoU are overall metrics that offer a tradeoff between precision and recall. These evaluation metrics are defined as follows:

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{6}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{7}$$

$$\text{F1} = \frac{2\text{TP}}{2\text{TP} + \text{FP} + \text{FN}} \tag{8}$$

$$\text{IoU} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}} \tag{9}$$

where TP, FP, and FN refer to the true positives, false positives, and false negatives, respectively.

To quantitatively assess the effect of a regularized and smooth road surface extraction that reaches human-level delineation, we introduce a smoothness index with an empirical threshold. We regard a vertex whose two-sided width difference with its $N1$ neighboring vertices is less than the threshold (three pixels here) as a correctly predicted smooth vertex. It is acknowledged that most of the vertices in a common road or a road segment are smooth ones. The ratio of the smooth vertices and all the smooth vertices in the ground truth denotes the smoothness index, as shown in the following equation:

$$\text{road smoothness} = \frac{\text{Num}_{\text{TP}}}{\text{Num}_{\text{labeled}}} \tag{10}$$

where $\text{Num}_{\text{labeled}}$ represents the number of labeled smooth vertices and Num_{TP} represents the number of labeled smooth vertices that are correctly predicted. This index reflects the smoothness property of a common road segment, so an algorithm obtaining a higher smoothness score indicates that it generates fewer abnormal points.

All the methods were implemented based on TensorFlow, and the experiments were executed on a computer with an NVIDIA GTX1060 GPU with 6-GB memory.

We used dropout and $L2$ regularization to reduce the overfitting, and batch normalization is to speed up the training. In the loss function, the weights of the width loss and the coarse boundary map loss were set to 1 and 100, respectively. For each batch, we randomly selected a starting vertex in the road network graph and started a random deep-first search or broad-first search to find its first 32 neighborhood vertices, to form a subgraph as the input data of the batch. We set 500 batches as an epoch and saved the model with the smallest loss as the best model in the training. The learning rate was initially set to $1e^{-3}$ and was divided by 5 if the total loss stopped decreasing for eight continuous epochs. When the learning rate dropped to $2e^{-8}$, we stopped the training and set the best model as the final model.

C. Comparison of the Different Methods

We compared the proposed method with five different methods of three different types: 1) U-Net [2]; 2) D-LinkNet [14] with max width buffer postprocessing; 3) NL-LinkNet [21] with max width buffer postprocessing; 4) the MRF model [26]; 5) a CNN classifier with empirical smoothing postprocessing we designed; and 6) RoadTagger [37]. The first three methods are segmentation methods, among which U-Net is a classic fully convolutional network (FCN)-based method and D-LinkNet and NL-LinkNet are specially designed for the road extraction task. The three other methods take centerline data as a prerequisite, as with the proposed method. Methods 4 and 5 predict the vertex width and perform regularization postprocessing, which has the same goal as the proposed method, i.e., to produce a smooth road surface. Method 6 is a combined CNN and GNN method, such as the proposed method, but it attempts to count the number of lanes in a road with different CNN and GNN structures.

To incorporate the centerline information into the segmentation methods, we made some changes to the original U-Net, D-LinkNet, and NL-LinkNet. For U-Net, we trained and tested it with image patches of road vertices from the proposed method as input data, so as to more accurately locate the scope of the roads. For D-LinkNet and NL-LinkNet, we performed postprocessing by intersecting the prediction result with the maximum width buffer mask of the road centerline, to exclude the background. The maximum width in our experiments was 64 pixels.

For the MRF model, we cite the experimental results (IoU and F1) reported in [26] for the Aerial KITTI and Bavaria datasets as the method is difficult to reproduce and the source code is unavailable. For the CNN classifier with smoothing postprocessing, we used the CNN module from the proposed method to predict the widths of each vertex and averaged the widths with the neighbors to obtain a smooth width result. The quantitative results of the different methods obtained on the four datasets are listed in Table I.

As shown in Table I, the proposed method achieves the best performance and comprehensively outperforms the other methods in the IoU, F1 score, and road smoothness index on all the datasets. Specifically, it outperforms the second best method (RoadTagger) by 2.2%, 2.6%, 6.4%, and 4.2% in IoU and by 27.9%, 34.8%, 33.0%, and 27.0% in the road smoothness index on the KITTI, Bavaria, Wuhan, and Shaoxing datasets, respectively. Although the proposed method is more complex, it can run as fast as the other methods in the Aerial KITTI and Bavaria datasets, processing one 1024×1024 image with about 1 s; it runs slower in the Wuhan and Shaoxing datasets as there are many complex roads where a plenty of road vertices need to be processed.

Although we restricted the scope of the roads in the remote sensing images for helping the three segmentation-

TABLE I
ROAD SURFACE EXTRACTION RESULTS, WHERE THE VALUES IN BOLD REPRESENT THE BEST PERFORMANCE

Dataset	Method	Recall	Precision	IoU	F1	Road smoothness	Time (s)
Aerial KITTI	U-net [2]	**0.9232**	0.7964	0.7446	0.8521	0.1752	0.9683
	D-LinkNet [14]	0.6733	0.8855	0.6164	0.7544	0.4159	1.3110
	NL-LinkNet [21]	0.7205	0.8769	0.6511	0.7828	0.3439	1.3968
	MRF [26]			0.7180	0.8360		
	CNN+smoothing	0.8116	0.9302	0.7628	0.8651	0.5250	0.5425
	RoadTagger [37]	0.8921	0.9160	0.8226	0.9022	0.4276	0.7857
	Proposed	0.9063	**0.9277**	**0.8444**	**0.9154**	**0.7062**	1.5396
Bavaria	U-net [2]	**0.9069**	0.7636	0.7082	0.8277	0.0912	0.8704
	D-LinkNet [14]	0.7971	0.8734	0.7135	0.8323	0.2489	1.1378
	NL-LinkNet [21]	0.6136	0.8958	0.5702	0.7198	0.2570	1.2083
	MRF [26]			0.7350	0.8480		
	CNN+smoothing	0.7978	**0.9410**	0.7598	0.8633	0.4662	0.2930
	RoadTagger [37]	0.9025	0.8833	0.8053	0.892	0.2213	0.4120
	Proposed	0.8969	0.9200	**0.8313**	**0.9075**	**0.5697**	0.7638
Wuhan	U-net [2]	**0.8644**	0.7451	0.6670	0.7996	0.2677	0.9808
	D-LinkNet [14]	0.7753	0.8892	0.7079	0.8284	0.4858	1.3105
	NL-LinkNet [36]	0.6206	0.9252	0.5936	0.7397	0.6145	1.6797
	CNN+smoothing	0.7402	**0.9390**	0.7067	0.8278	0.6174	2.5547
	RoadTagger [37]	0.7680	0.8858	0.6996	0.8225	0.4188	3.1640
	Proposed	0.8195	0.9170	**0.7634**	**0.8655**	**0.7123**	5.8984
Shaoxing	U-net [2]	**0.9568**	0.6504	0.6318	0.7744	0.4517	0.9800
	D-LinkNet [14]	0.7737	0.8842	0.7021	0.8253	0.5892	1.3455
	NL-LinkNet [36]	0.7824	0.8445	0.6835	0.8119	0.4740	1.4141
	CNN+smoothing	0.7432	**0.9539**	0.7174	0.8354	0.8271	1.6539
	RoadTagger [37]	0.8057	0.9168	0.7508	0.8576	0.6347	2.2656
	Proposed	0.8431	0.9301	**0.7930**	**0.8845**	**0.9042**	4.0625

based methods, i.e., U-Net, D-LinkNet, and NL-LinkNet, they show the worst performance and obtain very low smoothness scores. Another option is imbedding the centerline information into a pixel-based segmentation framework, but it is outside of the scope of this work. What we have observed is in agreement with the common knowledge, i.e., the CNN-based segmentation methods struggle to achieve smooth or structured boundaries.

The conventional MRF-based method [26] considers the road centerlines as known data. It performs better than U-Net, D-LinkNet, and NL-LinkNet; however, it is outperformed by 12.57% and 9.63% in the IoU score by the proposed method on the KITTI and Bavaria datasets, respectively. Moreover, the MRF-based method requires the empirical setting and computing of various types of features, including edge features, road and car detector results, domain knowledge of road smoothness, and overlapping constraints. In contrast, the proposed deep learning-based method gets rid of the need for handcrafted feature design and is easier to use.

RoadTagger, which is the second best performing method, was designed to count the number of lanes in a road by the combination of a CNN and a GNN, as in the proposed network structure. This implies that the cutting-edge approach of GNN-based fine adjustment with CNN-based feature extraction is the best baseline model for structured road surface extraction. However, the proposed method outperforms RoadTagger significantly, due to the specific CNN encoder with road boundary branch and the GNN model with a proper propagation rule. In particular, the road smoothness scores of the proposed method are extremely high (27.86%, 34.84%, 32.92%, and 26.95% higher than RoadTagger on the Aerial KITTI, Bavaria, Wuhan, and Shaoxing datasets, respectively). The improvement in road smoothness is critical to practical and automatic road extraction and is equivalent to manual smoothing and structured double-line road delineation.

The CNN classifier with smoothing postprocessing we designed can significantly improve the road smoothness when compared with U-Net, D-LinkNet, and NL-LinkNet; however, this empirical global postprocessing operation is isolated from the learnable width regression, and it cannot tell whether the road width really changes or if this is just a prediction error made by the CNN classifier. In contrast, the GNN in the proposed method can iteratively learn how to adjust the road width of each vertex (road segment) and can improve the road smoothness score by about 10%.

Fig. 5 shows a qualitative comparison of the different road surface extraction methods on images from the different datasets. From top to bottom, each pair of rows represents the results for images from the Aerial KITTI, Bavaria, Wuhan, and Shaoxing datasets. From left to right, the images denote the ground truth and the results of U-Net, D-LinkNet, NL-LinkNet, CNN + smoothing postprocessing, RoadTagger, and the proposed method. In the fifth to the seventh columns, we use green circles and yellow lines to, respectively, represent the vertices and centerlines. Fig. 5 also shows different road conditions, including straight roads, T-junctions, and other complicated roads.

It can be observed that U-Net, D-LinkNet, and NL-LinkNet show the worst performance, in both the segmentation and regularization effects. The segmentation-based methods are

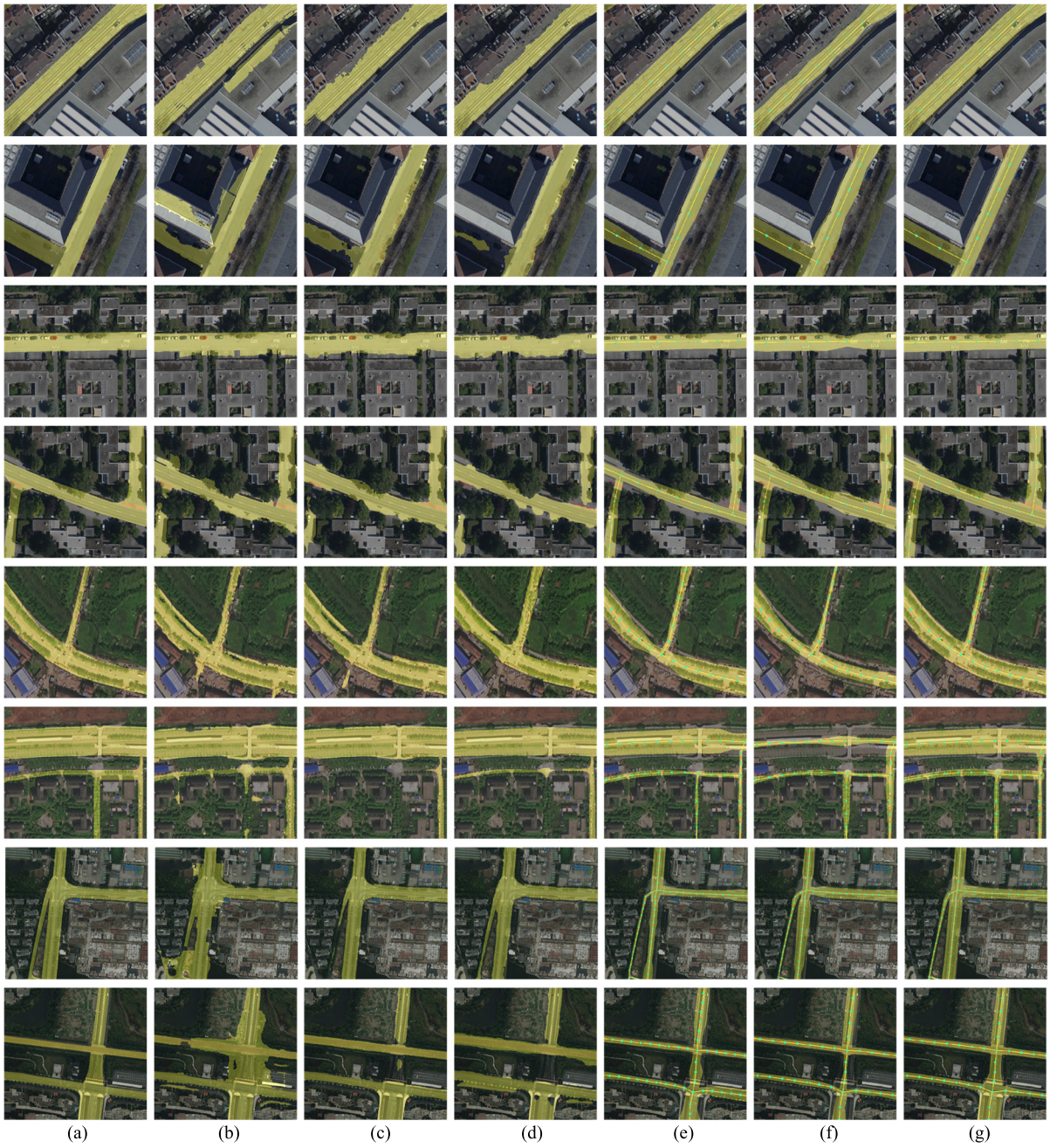


Fig. 5. Qualitative results of the different road surface extraction methods. From top to bottom, each pair of rows represents the prediction results for images from the Aerial KITTI, Bavaria, Wuhan, and Shaoxing datasets. (a) Ground truth. (b) Result of U-Net. (c) Result of D-LinkNet. (d) Result of NL-LinkNet. (e) Result of CNN + smoothing postprocessing. (f) Result of RoadTagger. (g) Result of the proposed method.

sensitive to shadows, trees, and other occlusions. As a result, the predicted road boundaries are irregular and some of them are broken. In addition, the segmentation-based methods also misjudge backgrounds with similar textures to roads. Clearly, this kind of method cannot be directly applied in the production of road surface maps.

Due to the constraint of the centerlines, the extraction results of the CNN + smoothing postprocessing and RoadTagger methods are much smoother than those of the segmentation-based methods; however, there are still many obvious flaws. In some straight road sections, the smoothing postprocessing can work well, but it performs worse in complicated situations, such as at intersections and occluded parts of the road in particular, as shown in the fifth and sixth columns of Fig. 5.

In contrast, the proposed method can not only handle the diverse backgrounds in the images, including shadows, trees, and other occlusions, but it can also handle the diverse foregrounds, including T-shaped intersections, X-shaped intersections, and other complex road sections, to obtain regularized, smooth, and complete road surface maps, as shown in the last column of Fig. 5. The well-designed GNN model allows us to suppress the influence of abnormal road vertices through the

Fig. 6. Offset centerlines (yellow) and the corrected centerlines (blue) by our method.

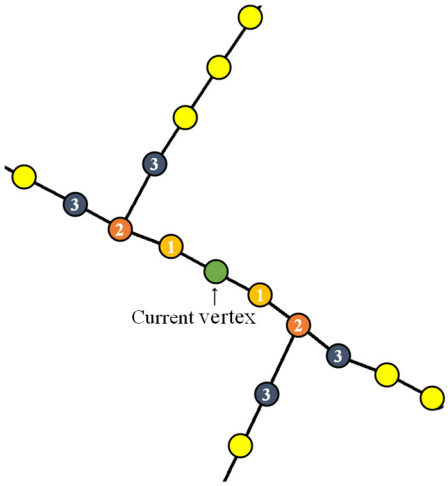


Fig. 7. Examples of different orders of neighborhood. The green circle is the current vertex, and the numbers in the vertices refer to the corresponding nth order neighborhood of the current vertex.

weighted information aggregation in the neighborhood and the proper message propagation in the graph, to obtain regularized and accurate road extraction results that approach the human delineation level.

D. Offset Centerline Correction

A side product of the proposed method is the corrected centerline map, which is very useful in practical situations as many newly accessed remote sensing images often cannot exactly fit the existing centerline vector maps, due to the geometric or geographical bias. We verified the performance of the proposed method on the Aerial KITTI dataset with simulated centerline bias through a comparison with the RoadTagger and CNN + smoothing postprocessing methods.

In Table II, the centerline offset is the average distance between the corrected centerline and the ground-truth centerline. The proposed method comprehensively outperforms the other two methods by at least 4% in IoU, 17% in smoothness, and 0.25 pixels in centerline bias. The centerline offset of the proposed method is only 1.4 pixels, which can be ignored in practice, and reaches the human delineation level. Fig. 6 shows a comparison of before (yellow lines) and after (blue lines) correction of the offset centerlines, where the centerlines have been adjusted to the correct positions.

It can also be observed that the road surface extraction results of the proposed method with biased centerlines as the input data are comparable to those obtained using accurate centerlines as the input data (by comparing Tables I and II), which proves that the proposed method is robust to the differing qualities of the existing centerline maps.

E. Comparison of Different Propagation Rules in the Road Graph

We designed the specific propagation rule in the road graph for propagating and aggregating the vertex information. We also introduced three different forms of propagation rules for the vertices in Section II-A: Type A (all vertices are connected), Type B (the vertices from different straight roads are blocked), and Type C (vertices from different road segments are blocked). In this section, we analyze the performances of each of these rules and their combination on the Aerial KITTI dataset. Table III shows that the combination of Types B and C outperforms the others. Compared with Type A, which is the commonly used and default propagation rule in GNN-based vision tasks, the combination of Types B and C obtains an improvement of 1.07% in IoU and 0.64% in F1 score. This indicates that the constraints of Types B and C are more adaptable to the practical road topological properties, and they reduce the unnecessary and even adverse mutual influence between roads (or road segments) of different types. As a result, we chose the combination of Types B and C as the final form of the propagation rule in the proposed GNN model.

F. Comparison of Different Orders of Neighborhood During the GNN Aggregation

In addition to the propagation rule, the definition of the neighborhood in the road graph also impacts the performance of GNN-based regression. In this experiment, we evaluated the influence of different orders of neighborhood of a vertex in the GNN aggregation on the Aerial KITTI dataset.

The order of neighborhood determines how many neighbors participate in the vertex information exchange of the current vertex, as shown in Fig. 7, where $N1$ represents the neighbors of the first order, $N2$ represents the neighbors of the second order (including the first order), and $N3$ represents the neighbors of the third order.

Table IV lists the road surface extraction results obtained using different orders of neighborhood during the GNN aggregation. It can be seen that $N1$ achieves the best performance in IoU and F1 score, and the result of using $N2$ achieves the highest road smoothness. The result of using $N3$ is the worst. These results can be explained by the fact that using more neighborhood vertices ($N2$), which aggregate more information, does improve the road smoothness when compared to $N1$, but also introduces bias through the averaging operation and difficulty in the proper message propagation at the same time. Further increase of the neighborhood vertices

TABLE II

ROAD SURFACE EXTRACTION RESULTS FOR THE AERIAL KITTI ANALOG OFFSET DATASET, WHERE THE VALUES IN BOLD REPRESENT THE BEST PERFORMANCE

Method	Recall	Precision	IoU	F1	Road smoothness	Centerline Offset (pixel)
RoadTagger [37]	0.8880	0.8785	0.7891	0.8816	0.2089	4.3745
CNN+smoothing	0.8263	**0.9228**	0.7699	0.8696	0.5460	1.6883
Proposed	**0.9018**	0.9143	**0.8294**	**0.9063**	**0.7173**	**1.4358**

TABLE III

ROAD SURFACE EXTRACTION RESULTS OBTAINED USING DIFFERENT PROPAGATION RULES ON THE AERIAL KITTI DATASET

Propagation rule			IoU	F1	Road smoothness
A	B	C			
√			0.8337	0.9090	0.7035
	√		0.8317	0.9077	0.5728
		√	0.8373	0.9112	0.6792
√	√		0.8348	0.9095	0.6765
√		√	0.8378	0.9115	0.6617
	√	√	**0.8444**	**0.9154**	**0.7062**
√	√	√	0.8344	0.9094	0.6671

TABLE IV

ROAD SURFACE EXTRACTION RESULTS OBTAINED USING DIFFERENT ORDERS OF NEIGHBORHOOD ON THE AERIAL KITTI DATASET

Neighborhood	Recall	Precision	IoU	F1	Road smoothness
N1	**0.9063**	**0.9277**	**0.8444**	**0.9154**	0.7062
N2	0.8956	0.9272	0.8351	0.9097	**0.7466**
N3	0.8962	0.9248	0.8335	0.9088	0.6833

TABLE V

ROAD SURFACE EXTRACTION RESULTS OBTAINED USING DIFFERENT GNN MODULES ON THE WUHAN DATASET

GNN module	Recall	Precision	IoU	F1	Road smoothness
GGNN [29]	0.8042	0.9130	0.7475	0.8550	0.4643
GCN [28]	0.8030	0.9161	0.7484	0.8557	0.6652
GAT [32]	**0.8196**	**0.9171**	**0.7636**	**0.8655**	**0.7481**

($N3$) instead results in the burden of information propagation and aggregation.

G. Comparison of Different GNN Modules

The GNN module is the third factor that impacts the performance of GNN-based regression. In this section, we compare three recent GNN modules—GCN [28], GGNN [29], and GAT [32]—on the Wuhan dataset. We used the same CNN encoder and set the number of stacked GNN modules to three for fairness. Table V and Fig. 8 show the quantitative and qualitative comparison results of using different GNN modules in the proposed framework.

It can be observed from Table V that using the GAT module results in a clearly better performance in both IoU and smoothness compared to the other two modules. This is because the other two modules apply equal weights for all the neighboring vertices when aggregating the vertex information, while the GAT module assumes that the contribution of each neighboring vertex is different and introduces an attention mechanism to learn the optimal weights instead of fixing them.

In Fig. 8, from top to bottom, we list three typical types of road surface: straight, T-junction, and intersection. The GGNN and GCN modules are more sensitive to occlusions on the road, resulting in jagged and irregular road boundaries, while the GAT module has a better aggregation strategy to resist occlusions in the images, achieving more regularized and smooth boundaries.

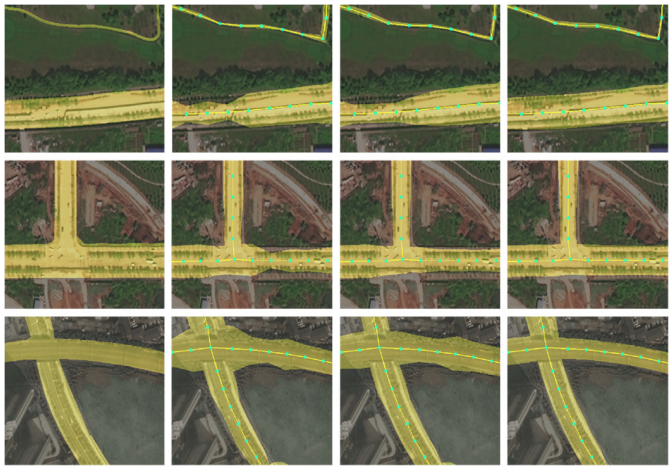


Fig. 8. Qualitative results of using different GNN modules. (a) Ground truth. (b) GGNN. (c) GCN. (d) GAT.

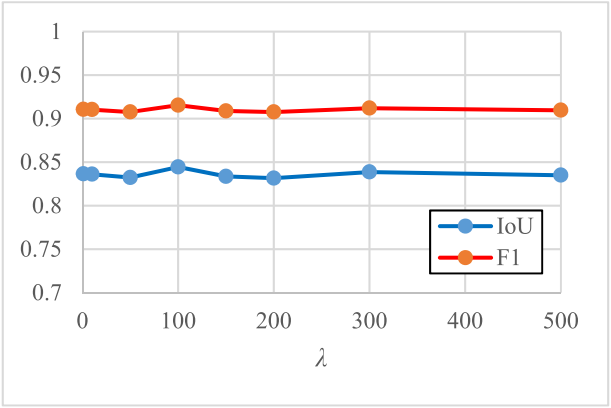


Fig. 9. IoU (blue line) and F1 (red line) curves with different λ values on the Aerial KITTI dataset.

H. Setting of Parameter λ

The tunable parameter λ balances the losses of road boundary and width. By changing the values of λ, we obtained IoU and F1 curves on the Aerial KITTI dataset, as shown in Fig. 9. The very little fluctuations in the curves indicate that the proposed method has good stability against the setting of λ. We set $\lambda = 100$ where a small peak is reached. Please note that we have trained the CNN part before training the whole network.

I. Prerequisite

The aim of this article is different from the mainstream methods that formulate road extraction as a pixel-level segmentation problem and inevitably suffer from shadows, trees, and other occlusions. In contrast, we start from a more practical view, i.e., how to reach human-level (essentially vector-level) delineation and reduce manual works. To achieve this goal, our method requires rough road centerline maps as prerequisite. Fortunately, OSM and other VGI sources have provided road centerlines around the world. Moreover, the historical centerline data can be accessed from the survey and mapping, or GIS department in an engineering task. If there is really no centerline data available, our method is also beneficial: one can easily draw a rough centerline map, instead of drawing the fine-grain road surface map with much heavier manual work, and input it into our network to produce the regularized road surface map.

IV. Conclusion

In this article, we have proposed a novel framework to extract regularized road surfaces from remote sensing imagery and easily accessible road centerline data. The offset of the road centerlines can also be corrected at the same time. In this framework, we formulate the road surface extraction problem as piecewise two-sided width regression of the road centerlines and introduce a novel combined CNN and GNN framework for the width regression. The CNN extracts both the high-level features and the boundary information from each patch of the vertex, and the GNN propagates the vertex features within the road graph with the constraint of two propagation rules to output the smooth two-sided width of each vertex. The novel design of the GNN-based adjustment results in regularized and smooth road surface extraction in most of the testing areas that approach the human delineation level, showing great potential for reducing the manual work of road map production.

References

[1] 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.

[2] 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.*, 2015, pp. 234–241.

[3] 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.

[4] L. C. Chen, G. Papandreou, and I. Kokkinos, "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.

[5] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 580–587.

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

[7] S. Ren *et al.*, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 91–99.

[8] X. Zhang, G. Wang, P. Zhu, T. Zhang, C. Li, and L. Jiao, "GRS-Det: An anchor-free rotation ship detector based on Gaussian-mask in remote sensing images," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 4, pp. 3518–3531, Apr. 2021.

[9] T. Zhang *et al.*, "Semantic attention and scale complementary network for instance segmentation in remote sensing images," *IEEE Trans. Cybern.*, early access, Aug. 26, 2021, doi: 10.1109/TCYB.2021.3096185.

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

[11] S. Saito, T. Yamashita, and Y. Aoki, "Multiple object extraction from aerial imagery with convolutional neural networks," *Electron. Imag.*, vol. 2016, no. 10, pp. 1–9, 2016.

[12] 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.

[13] 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.

[14] 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. 182–186.

[15] 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.

[16] X. Zhang, X. Han, C. Li, X. Tang, H. Zhou, and L. Jiao, "Aerial image road extraction based on an improved generative adversarial network," *Remote Sens.*, vol. 11, no. 8, p. 930, Apr. 2019.

[17] P. Shamsolmoali, M. Zareapoor, H. Zhou, R. Wang, and J. Yang, "Road segmentation for remote sensing images using adversarial spatial pyramid networks," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 6, pp. 4673–4688, Jun. 2021.

[18] Y. Wei, K. Zhang, and S. Ji, "Simultaneous road surface and centerline extraction from large-scale remote sensing images using CNN-based segmentation and tracing," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 12, pp. 8919–8931, Dec. 2020.

[19] M. Zhou, H. Sui, S. Chen, J. Wang, and X. Chen, "BT-RoadNet: A boundary and topologically-aware neural network for road extraction from high-resolution remote sensing imagery," *ISPRS J. Photogramm. Remote Sens.*, vol. 168, pp. 288–306, Oct. 2020.

[20] T. Alshaikhli, W. Liu, and Y. Maruyama, "Simultaneous extraction of road and centerline from aerial images using a deep convolutional neural network," *ISPRS Int. J. Geo-Inf.*, vol. 10, no. 3, p. 147, Mar. 2021.

[21] Y. Wang, J. Seo, and T. Jeon, "NL-LinkNet: Toward lighter but more accurate road extraction with nonlocal operations," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[22] K. Zhou, Y. Xie, Z. Gao, F. Miao, and L. Zhang, "FuNet: A novel road extraction network with fusion of location data and remote sensing imagery," *ISPRS Int. J. Geo-Inf.*, vol. 10, no. 1, p. 39, Jan. 2021.

[23] 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.

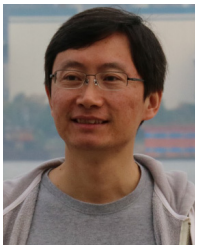
[24] A. Chaurasia and E. Culurciello, "LinkNet: Exploiting encoder representations for efficient semantic segmentation," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, Dec. 2017, pp. 1–4.

[25] M. Haklay and P. Weber, "OpenStreetMap: User-generated street maps," *IEEE Pervasive Comput.*, vol. 7, no. 4, pp. 12–18, Oct. 2008.

[26] G. Mattyus, S. Wang, S. Fidler, and R. Urtasun, "Enhancing road maps by parsing aerial images around the world," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1689–1697.
[27] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, and G. Monfardini, "The graph neural network model," *IEEE Trans. Neural Netw.*, vol. 20, no. 1, pp. 61–80, Jan. 2016.
[28] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," 2016, *arXiv:1609.02907*.
[29] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," 2015, *arXiv:1511.05493*.
[30] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, pp. 1263–1272.
[31] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1025–1035.
[32] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," 2017, *arXiv:1710.10903*.
[33] L. Landrieu and M. Simonovsky, "Large-scale point cloud semantic segmentation with superpoint graphs," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4558–4567.
[34] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 38, no. 5, pp. 1–12, 2019.
[35] S. Yan, Y. Xiong, and D. Lin, "Spatial temporal graph convolutional networks for skeleton-based action recognition," 2018, *arXiv:1801.07455*.
[36] H. Ling, J. Gao, A. Kar, W. Chen, and S. Fidler, "Fast interactive object annotation with curve-GCN," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 5257–5266.
[37] S. He *et al.*, "RoadTagger: Robust road attribute inference with graph neural networks," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 10965–10972.
[38] T. S. Jepsen, C. S. Jensen, and T. D. Nielsen, "Relational fusion networks: Graph convolutional networks for road networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 1, pp. 418–429, Jan. 2022.

Jingjing Yan received the B.Sc. degree from the School of Remote Sensing and Information Engineering, Wuhan University, Wuhan, China, in 2020, where she is currently pursuing the M.Sc. degree.

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



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

He is currently a Professor with the School of Remote Sensing and Information Engineering, Wuhan University. He has coauthored more than 70 articles. His research interests include photogrammetry, remote sensing image processing, mobile mapping system, and machine learning.

Yao Wei received the B.S. degree in geographic information science from the China University of Petroleum, Qingdao, China, in 2018, and the M.S. degree in photogrammetry and remote sensing from Wuhan University, Wuhan, China, in 2021. She is currently pursuing the Ph.D. degree with the Faculty of Geo-Information Science and Earth Observation (ITC), University of Twente, Enschede, The Netherlands.

Her research interests include computer vision and machine learning.