ROAD NETWORK EXTRACTION FROM SATELLITE IMAGES USING CNN BASED SEGMENTATION AND TRACING

Yao Wei [1], *Kai Zhang* [1], *Shunping Ji* [1]*

[1] School of Remote Sensing and Information Engineering, Wuhan University, 430079, China
*Corresponding author: jishunping@whu.edu.cn

ABSTRACT

Drawing and updating road networks are both time-consuming and labor-intensive. Deep learning technology and high-resolution remote sensing images have provided opportunities for automatic road extraction. However, recent convolutional neural network (CNN) based segmentation methods have shown serious problems on connectivity; road tracing methods with single starting point perform well in connectivity but often result in part areas unreached. We propose a multiple starting points tracer which benefits from both segmentation and tracing methods. We compare our approach with most recent tracing methods on satellite images of global cities and find that our method achieves 8% improvement on IoU.

Index Terms— Road network extraction, segmentation, tracing, convolutional neural network, corner detection.

1. INTRODUCTION

Road extraction from high-resolution satellite images, a fundamental research in earth observation and remote sensing, plays an important role in the applications of geo-information updating, urban transportation planning and autonomous vehicle driving. Moreover, accurate road maps can assist in scene understanding by providing prior knowledge for identifying buildings, crops and many other surface objects. Although it has received considerable attentions in the past decades, road extraction is still a challenging task because of complex ground information, such as shadow of buildings, shade of trees, vehicles, and road-like constructions. Therefore, extracted roads may suffer from poor connectivity and false recognition.

Traditional road extraction algorithms perform well in simple scenes but fail to handle complex scenes. In recent years, researchers have put forward several effective models and algorithms inspired by deep learning. In general, most works on this topic can be divided into two categories: segmentation-based approach, and tracing-based approach. The former aims at generating binary pixel-wise mask of roads, and the latter aims at detecting road central lines.

With respect to segmentation-based approach, one of the early attempts was made by Guo et al. [1], who utilized the gradient information to segment road areas assisted by digital line graph (DLG) data. Watershed transformation [2] and morphology method [3] were used to extract roads from aerial images. As the development of deep learning, many CNN based methods proposed for semantic segmentation have been introduced to detect roads from aerial or satellite images. Zhang et al. [4] combined residual learning and U-net model to achieve better results. Hong et al. [5] utilized a pyramid like architecture to enhance road connectivity. In addition, complex post processing like conditional random field (CRF) are usually required to refine road segmentation.

In the case of tracing-based approach, researchers applied the GPS data [6] to improve road topology. Road centerlines are usually obtained by eroding prior segmentation result. Some researchers have attempted to search road centerlines from satellite images directly. Gellert et al. [7] estimated road topology on segmentation masks, reasoned and repaired the brokenness by shortest path search. Favyen et al. [8] proposed a state-of-the-art iterative search algorithm to estimate road centerlines directly from satellite images.

Up to now, remarkable improvements have been made by segmentation-based approach and tracing-based approach, while the problem is far from being solved. Roads extracted by segmentation-based methods often miss links in crossroad, and road network graphs inferred by single-starting-point searching methods are usually blocked by objects like rivers, viaducts. In order to solve this problem, we combine the latest segmentation and tracing strategies to utilize their complementary advantages. Specifically, we supply multiple starting points for tracing-based approach by detecting corners from the output of an improved segmentation-based approach. Besides, we propose two algorithms to refine the final graph. By contrast, our approach obtains more road details while maintains road connectivity.

2. METHODOLOGY

This study attempts to construct topological road network graphs from remote sensing images. A two-stage method for road centerline extraction is proposed and shown in **Fig. 1**. Stage A includes an initial pixel-wise semantic segmentation, followed by a starting points generation algorithm described in Section 2.1. Stage B constructs road network maps by road centerline tracing and post refinements, which is described in Section 2.2.

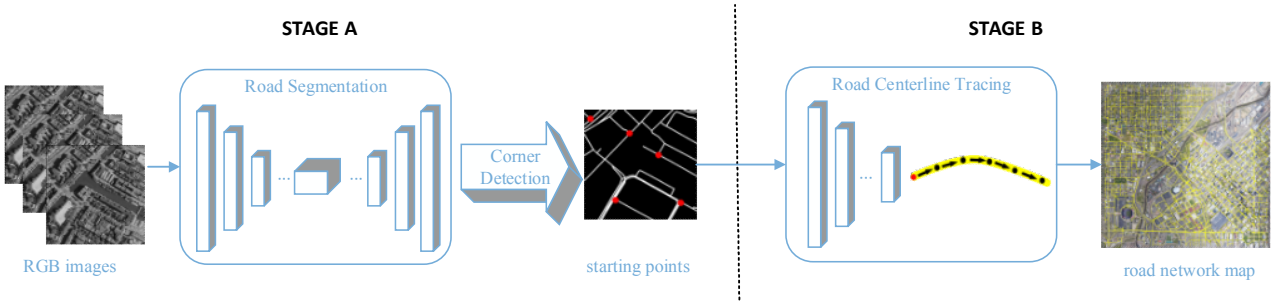


Fig. 1. Overview of our method. In STAGE A, a segmentation network is trained to produce binary masks for generating starting points by utilizing a corner detection algorithm. In STAGE B, road centerline tracer constructs road network maps from multiple starting points.

2.1 Generating starting points

Starting point generation is divided into two steps, road segmentation and starting points detection. We utilize a fully convolutional network (FCN) similar to D-Linknet [9], which won the DeepGlobe 2018 Road Extraction Challenge with the best IoU scores. The network includes three parts: encoder, decoder and lateral connections between them. The encoder utilizes ResNet-34 [10] model which was pre-trained on the ImageNet dataset [11] to accelerate training procedure. Besides, dilated convolutions with various dilation rates are introduced into the lateral connections to capture multi-scale information as well as increase the receptive field of feature points. Transposed convolution is involved in decoder to restore the resolution of feature map. Apart from binary cross entropy (BCE) loss and dice coefficient loss, we design a new loss named link loss for constraining road connectivity:

$$\text{link loss} = 1 - \frac{\sum_{i=1}^{N} |P_i \cap GT_i|}{\sum_{i=1}^{N} |GT_i|} \tag{1}$$

where P is predicted mask, GT is single-pixel width ground truth mask after morphology thinning, and N is batch size.

Then, we use corner detector to generate starting points from segmentation masks. In order to produce high-quality road network maps, the starting points should be uniformly distributed in images. For each point, it is regarded as efficient if it has more potential directions. For example, if the starting point locates at the crossroad, it has four potential search directions. We locate those efficient starting points by the Good Feature To Track operator [12], which is an improved Harris corner detector. In Harris corner operator, the scoring function is:

$$R = \lambda_1 \lambda_2 - k(\lambda_1 + \lambda_2)^2 \tag{2}$$

While in the Good Feature To Track operator, it is:

$$R = \min(\lambda_1, \lambda_2) \tag{3}$$

where λ_1 and λ_2 are eigen values of matrix M which is a weighted covariance matrix as in (4), I means gradient image.

$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix} \tag{4}$$

Instead of applying the detector directly on segmentation map, we also skeletonize the road mask to obtain thin road lines for better searching accurate starting points.

2.2 RoadTracer with multi-starting points

We utilize a road tracing model similar to RoadTracer [8], to construct accurate road maps from satellite images. An iterative search algorithm based on CNN is utilized to derive road network maps. The network structure is demonstrated as **Fig. 2**. The input layer consists of a 256 × 256 window centered on the current point S_{top} in a stack. This window has five channels: the RGB values of the image patch around S_{top}, the ground truth road graph G*, which is only used in training and replaced with a blank graph in prediction, the currently constructing graph, G. The output layer consists of two components: an action component that decides either walk or stop, O_{action} = <o_{walk}, o_{stop}>, and an angle component that decides which angle to walk towards, O_{angle}.

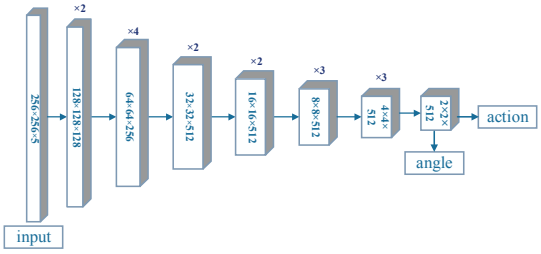
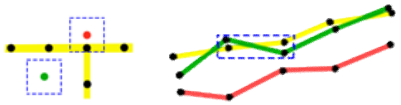


Fig. 2. Architecture of the CNN for road tracing. The ×N means a stack of N same layers, all the convolution kernel size is 3 × 3, and the down-sampling is achieved with a stride of 2.

The searching starts from a known point on the road network. Vertices and edges are added to the path as search goes. The CNN is invoked at each step to determine either to walk a fixed distance at an angle or stop and step back to the previous vertex in the search tree. However, in the original RoadTracer [8], the searching starts from a given point on the road network, which lowered the automation of the algorithm.

We use multiple corner points automatically derived from Stage A as starting points for road centerline tracer, rather than choose single starting point from ground truth as RoadTracer does. Multi-point tracing means reasoning road graph from each point in a points list. Considering the computing expense, we propose an Adaptive Starting Point Decision (ASPD) algorithm shown in **Fig. 3 (a)**, which dynamically picks out next starting point according to earlier explored graph. Specifically, the starting point of following

tracing is determined by whether earlier explored graph is outside a bounding box centered at a current point. Multiple starting point road tracing generates road network graphs with overlaps which requires post processing. Accordingly, we propose a Graph Merging (GM) algorithm shown in **Fig. 3 (b)**, which first randomly takes a graph as the base graph, then judges whether the angle between edges contained in the same bounding box exceeds the threshold. For example, if the two edges are parallel, the angle is 0. We merge edges if the angle is lower than the threshold, or add edges into base graph on the contrary. Algorithm 1 shows the pseudocode for road centerline tracer with multiple starting points.



(a) (b)

Fig. 3. The yellow line represents the base graph, black points are vertices on the road. (a) shows the ASPD algorithm. The blue window centered at a starting point (red) intersecting with base graph will be removed. On the contrary, the starting point (green) outside the graph is retained. (b) shows the GM algorithm. The green segment in the blue bounding box will be merged to base graph (yellow). On the other hand, the red one will be retained.

Algorithm 1 Road Centerline Tracer with multiple starting points

Input: starting points list C, an initial graph array G_{array}, window W_i centered at C_i, threshold for GM algorithm T,
while C is not empty **do**
  random pick C_i from C
  initialize W_i centered at C_i
  if G_{array} intersect with W_i; **break**
  else
    G_i =centerline_tracing (C_i, Image)
    add G_i to G_{array}
  end if
  remove C_i from C
end while
random pick G_{base} from G_{array}
for G_i in G_{array}
  for edge in G_i:
    if angle_difference(edge, G_{base}) > T **then**
      add edge to G_{base}
    end if
return G_{base}

3. EXPERIMENTAL RESULTS AND DISCUSSIONS

3.1 Datasets and metrics

To thoroughly evaluate the performance of our algorithm, we assemble a large collection of high-resolution satellite images and corresponding ground truth road network graphs, which is similar to the dataset of RoadTracer [8]. We obtain satellite images from Google Earth at 60 cm/pixel resolution, and the road network graphs from OpenStreetMap (OSM) [13] covering the urban core of 37 cities across 6 countries. We convert the coordinate system of the road network so that the annotations and satellite images correctly correspond. For each city, the graph covers a region of approximately 24 square kilometers around the city center. The dataset is divided into a training set with 25 cities and a test set with 12 other cities. Due to the limitation of GPU capacity, those images are cropped into 1024×1024 tiles for training. In prediction, we choose an 8192 × 8192 pixels region of each city for accuracy evaluation as area of different cities varies.

The road extraction task can be considered as a binary classification problem, where road pixels are positives and non-road pixels are negatives. In [8], the road centerline graph is evaluated on junction metric, however, this indicator focuses only on connectivity and performs poor on completeness. In contrast, we evaluate the road centerline graph on F1-Score and Intersection-over-Union (IoU). IoU refers to the ratio of the intersection between predicted pixels and actual pixels to their union, which is commonly used as the evaluation indicator of semantic segmentation and target detection. Considering that IoU cannot properly evaluate road topology on single-pixel width centerlines, we have expanded the single-pixel width ground truth and prediction to 8 pixels wide. Although they may show relative low score in assessing road centerline graph, we find the IoU and F1-score could well discriminate the performances of different methods, and reflect both connectivity and completeness of a graph.

3.2 Implementation details

For our segmentation model, we implement data augmentation including image horizontal flip, vertical flip, diagonal flip, color jittering, shifting and scaling. We add BCE loss, dice coefficient loss and our link loss with equal weight as loss function and choose Adam [14] as our optimizer. The learning rate was originally set to 2e-4, and divided by 5 while observing the training loss decreasing slowly for 3 times. The batch size during training phase was fixed as 4 on 1024×1024 tiles. It took about 108 epochs for our network to converge. We did test time augmentation (TTA) in prediction, including image horizontal flip, vertical flip, diagonal flip (predicting each image $2 \times 2 \times 2 = 8$ times) also on 1024×1024 tiles, and then stitch the outputs to produce the final segmentation maps of original 8192×8192 image size. Then, we averaged the probability of each prediction, using 0.5 as our prediction threshold to generate binary outputs.

For corner detection from binary outputs of segmentation, we generate maximum 100 points and minimum distance of 400 pixels for adjacent corners. For road centerline tracer, we set the search window size as 256 × 256 pixels. The batch size is 4 and the loss function includes three parts, detection loss, action loss and angle loss. We use Adam optimizer and train about 400 epochs. In the ASPD algorithm, we set the radius of search bounding box as 60 pixels which is 3 times of each road segment length. Each edge's bounding box is expanded by 40 pixels and the angle threshold is 30° in the GM algorithm.

3.3 Results

We implement RoadTracer [8], our RoadTracer with single starting point (RoadTracer-S), RoadTracer with multiple starting points (RoadTracer-M) and evaluate these three models on 12 cities in the test set. The performance of different models is shown in Table 1.

Table 1. Comparison of different methods on 12 cities

Method	F1-Score	IoU
RoadTracer [8]	0.2692	0.1700
RoadTracer-S (ours)	0.2717	0.1725
RoadTracer-M (ours)	**0.3733**	**0.2575**

From the experiments, we find that the starting point locating in crossroad could make slight improvement on RoadTracer, nearly 0.2% on IoU. This proves that the starting point with more potential directions could produce little better results. Compared with RoadTracer, our RoadTracer-M makes great improvement in road topology search: 10.4% and 8.7% improvement (38.6% and 51.1% relative improvement) on the F1-Score and IOU, respectively. We also present some detailed results of cities in our test set in Table 2.

Table 2. Quantitative Evaluation results (IoU) on four test cities

	Chicago	Paris	Pittsburgh	Toronto
RoadTracer	0.17	0.15	0.05	0.26
RoadTracer-S	0.15	0.15	0.07	0.26
RoadTracer-M	**0.31**	**0.24**	**0.28**	**0.48**

In **Fig. 4**, we illustrate qualitative results in crops from four cities: Chicago, Pittsburgh, Paris and Toronto. Compared with RoadTracer, RoadTracer-S could find more roads in some regions. RoadTracer-M performs much better in searching more areas and extracting more roads in cities while RoadTracer is always blocked by bridge and viaduct.

(a) (b) (c)

Fig. 4. Comparison of results obtained by roadtracer from three types of starting points in four cities, Chicago (top), Paris, Pittsburgh, Toronto (bottom). (a) RoadTracer. (b) RoadTracer-S. (c) RoadTracer-M. We overlay the predicted graph (yellow) over ground truth from OSM (light blue).

4. CONCLUSION

In this paper we presented an approach which integrates CNN based road segmentation and road centerline tracing. Segmentation result is utilized to assist tracing by generating many uniformly distributed starting points. Searching from multiple starting points can efficiently eliminate viaduct or river blocking problems which exist in single starting point road tracing. The experiment results show that our multiple starting points road tracing method improves road extraction result significantly. Furthermore, two optimizing methods are proposed to reduce computing cost and improve final results. We plan to keep exploring the combination of segmentation-based approach and tracing-based approach to provide high-quality road centerline networks and pixel-based road masks simultaneously.

5. REFERENCES

[1] D. Guo, A. Weeks, H. Klee. "Segmentations of road area in high resolution images." In *Geoscience and Remote Sensing Symposium, Proceedings. IEEE International*, vol. 6, pp. 3810-3813, 2004.

[2] S. Beucher, M. Bilodeau. "Road segmentation and obstacle detection by a fast watershed transformation." In *Intelligent Vehicles' 94 Symposium, Proceedings.* pp. 296-301, 1994.

[3] S. Letitia, E.C. Monie. "Road segmentation from satellite aerial images by means of adaptive neighborhood mathematical morphology." In *Computer and Communication Engineering. International Conference on*, pp. 427-432, 2008.

[4] Z. Zhang, Q. Liu, Y. Wang. "Road Extraction by Deep Residual U-Net." *IEEE Geoscience and Remote Sensing Letters.* pp.749-753, 2018.

[5] Z. Hong, D. Ming, K. Zhou, Y. Guo, T. Lu. "Road Extraction From a High Spatial Resolution Remote Sensing Image Based on Richer Convolutional Features." *IEEE Access*, vol 6, pp. 46988-47000. 2018.

[6] J. Biagioni, J. Eriksson. "Map inference in the face of noise and disparity." In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, pp. 79-88, 2012.

[7] G. Máttyus, W. Luo, R. Urtasun. "DeepRoadMapper: Extracting Road Topology From Aerial Images." In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3438-3446. 2017.

[8] F. Bastani, S. He, S. Abbar, M. Alizadeh. "RoadTracer: Automatic Extraction of Road Networks from Aerial Images." In *Computer Vision and Pattern Recognition (CVPR)*. 2018.

[9] L. Zhou, C. Zhang, M. Wu. "D-LinkNet: LinkNet with Pretrained Encoder and Dilated Convolution for High Resolution Satellite Imagery Road Extraction." In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 182-186, 2018.

[10] K. He, X. Zhang, S. Ren, J. Sun. "Deep residual learning for image recognition." In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770-778. 2016.

[11] J. Deng, W. Dong, R. Socher, L.J. Li, K. Li. "Imagenet: A large-scale hierarchical image database." In *Computer Vision and Pattern Recognition. IEEE Conference on*, pp. 248-255, 2009.

[12] J. Shi, C. Tomasi. *Good features to track*. Cornell University, 1993.

[13] M. Haklay, P. Weber. "Openstreetmap: User-generated street maps." IEEE *Pervas Comput*, vol 7, no. 4, pp.12-18, 2008.

[14] D.P. Kingma, J. Ba. "Adam: A method for stochastic optimization." *arXiv preprint arXiv:1412.6980*, 2014.