RNGDet++: Road Network Graph Detection by Transformer with Instance Segmentation and Multi-scale Features Enhancement

Zhenhua Xu, *Graduate Student Member, IEEE*, Yuxuan Liu, *Graduate Student Member, IEEE*, Yuxiang Sun, *Member, IEEE*, Ming Liu, *Senior Member, IEEE*, and Lujia Wang, *Member, IEEE*

***Abstract*—The graph structure of road networks is critical for downstream tasks of autonomous driving systems, such as global planning and navigation. In the past, the road network graph is usually manually annotated by human experts, which is time-consuming and labor-intensive. To obtain the road network graph with better effectiveness and efficiency, automatic approaches for road network graph detection are required. Previous works either post-process semantic segmentation maps or propose graph-based algorithms to directly predict the road network graph. However, previous works suffer from hard-coded processing algorithms and inferior final performance. To enhance the previous SOTA (State-of-the-Art) approach RNGDet, we add an instance segmentation head to better supervise the model training, and enable the model to leverage multi-scale features of the backbone network. Since the new proposed approach is improved from RNGDet, it is named RNGDet++. All approaches are evaluated on two large publicly available datasets. RNGDet++ outperforms baseline models on almost all metrics scores. It improves the topology correctness APLS (Average Path Length Similarity) by around 3%. The demo video and supplementary materials are available on our project page https://tonyxuqaq.github.io/projects/RNGDetPlusPlus/.**

***Index Terms*—Road Network Detection, Imitation Learning, GIS, Autonomous Driving, Robotics.**

I. Introduction

THE vector map of road elements, including HD maps (High-definition maps) and SD maps (Standard-definition maps), is critical for nowadays autonomous vehicles. The road network graph is one kind of SD map and it records the reliable position and topology information of drivable roads, which is fundamental for downstream tasks of autonomous vehicles, such as global route planning and navigation [1], [2]. The autonomous vehicle must query the prior road network graph to find the optimal road path to reach the destination, especially in complicated urban areas. In addition, the road network graph can be applied for navigation tasks of human users, such as the Google map we use in our daily life. Usually, the road network graph consists of vertices and edges, where vertices represent key points of the road network (e.g., road ends and road intersections) while each edge demonstrates one segment of road. Since road network graph tends to cover a large area, such as a city or even a country, manually annotating it by human experts is time-consuming and labor-intensive, which severely raises the cost and slows down the wide deployment of autonomous vehicles. Therefore, the approach that could automatically detect the road network in vector graph format is of great interest to the community.

Since the road network only contains road-level information of roads, the detection task of the road network does not demands very high-resolution input data. Thus, aerial images obtained by UAV (Unmanned Aerial Vehicle) or satellites [3] are already sufficient for our task, which is much cheaper and easier to access than data collected by vehicle mounted sensors. In this paper, the road network is detected from large aerial image tiles with 1m/pixel resolution ratio.

There are a lot of past works that have similar tasks with ours, which could be classed into three categories. The first category of works are based on semantic segmentation [3]–[13]. These works first predict the semantic segmentation mask of the road network and then extract the graph structure by post-processing algorithms. The second category of works is two-stage and can directly obtain the road network graph without complicated post-processing [14]–[16]. In these works, the authors first calculate graph vertices by predicting the vertex heatmap, and then predict graph edges by connecting obtained graph vertices. The last category of works treats the graph detection task as an MDP (Markov Decision Process) problem and proposes an iterative decision-making algorithm to detect the road network graph [17]–[20]. Starting from predicted initial candidates, these works train an agent network that can detect the road network graph by iterations, which behaves in a similar way to human experts. Among them, RNGDet [19] (Road Network Graph Detector) proposes and trains a DETR-like (Detection by Transformer) [21] transformer network to track and detect the road network graph, which presents the SOTA (State-of-the-Art) performance so far. However, RNGDet does not fully make use of multi-scale features extracted by the CNN backbone network, which restricts the further improvement of this model.

In this paper, we propose RNGDet++, a more powerful and novel approach that directly detects the road network in graph format. Compared with RNGDet, RNGDet++ can make use of multi-scale features of the backbone network in a similar way to FPN (Feature Pyramid Network) [22], and presents more powerful effects in our task. Besides, we add an instance segmentation head into the model to better supervise the training phase of

Zhenhua Xu and Yuxuan Liu are with The Hong Kong University of Science and Technology (email: {zxubg,yliuhb}@connect.ust.hk).

Yuxiang Sun is with the Department of Mechanical Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong (e-mail: yx.sun@polyu.edu.hk, sun.yuxiang@outlook.com).

Ming Liu is with The Hong Kong University of Science and Technology (Guangzhou), Nansha, Guangzhou, 511400, Guangdong, China, and also with The Hong Kong University of Science and Technology, Hong Kong SAR, China, and also with HKUST Shenzhen-Hong Kong Collaborative Innovation Research Institute, Futian, Shenzhen. (email: eelium@ust.hk)

Lujia Wang is with The Hong Kong University of Science and Technology, and also with Clear Water Bay Insitute of Autonomous Driving (Shenzhen) (email: eewang@ust.hk).

Corresponding author: Lujia Wang.

the network, which enhances the robustness and performance of the approach. RNGDet++ is trained by imitation learning. We conduct comparison experiments and ablation studies on the city-scale dataset released by Sat2Graph [15] and the SpaceNet dataset [23]. The contributions of this paper are as follows:

- We propose RNGDet++, a novel approach that can make full use of multi-scale features to effectively detect road networks in the graph format.
- We add an instance segmentation head to the model, which better supervises the training of the network and improves the final performance of RNGDet++.
- RNGDet++ and all baseline models are evaluated on two large publicly available datasets. RNGDet++ presents superior results than baseline approaches in the comparison experiments.
- We open source the code and data which are available on our project webpage https://tonyxuqaq.github.io/projects/RNGDetPlusPlus/.

II. Related Works

A. Naive Road Element Graph Detection from Bird's-Eye View

Naive road element graph detection refers to the detection task of road elements with relatively simple topology, such as road boundaries [14], [24], [25], road curbs [26], [27] and road lanelines [28]–[31]. Under common circumstances, these road elements do not have complicated merge, split or intersection, thus the detection of the graph structure of these road elements are relatively with fewer difficulties. The input data of past works is BEV (Bird's-Eye View) images which are either aerial images [14], [25], [26] or projected images of the pre-built point-cloud map [24], [28], [29]. Most previous works utilize a decision-making network to iteratively detect the graph structure of target naive road elements. Liang *et al.* [24] proposed a CNN-based decision network to detect the road boundary in BEV images obtained from the pre-built point cloud map. Li *et al.* [29] further modified the algorithm to handle simple topology changes of lane lines (e.g., split and merge) on high ways. Although the aforementioned approaches could achieve satisfactory results in their specific detection tasks of target naive road elements, they cannot be adapted to the detection task of the road network graph, since road network tends to have more complicated topology structure, such as road intersections and road overlapping (e.g., overpasses). Therefore, more powerful algorithms are demanded for our task.

B. Road Network Detection from Aerial Images

With the fast development of aerial imaging techniques, high-resolution aerial images from all over the globe can be easily accessed nowadays. Thus, most past works on road network detection take aerial images as input [3]–[13], [16]–[20], and they could be classified into three categories: (1) Segmentation-based approaches [3]–[13]. This category of approaches first predict the semantic segmentation map of road networks, and then conduct post-processing algorithms (e.g., skeletonization and binarization) to extract the graph structure. However, they usually have inferior topology correctness, especially when road intersections or road overlappings are encountered. (2) Two-stage-graph-based approaches [15], [16]. He *et al.* [15] proposed a two-stage algorithm Sat2Graph to directly predict the graph of road networks without complicated hard-code post-processing. The authors first predicted the heatmap of road network graph vertices and extracted vertex coordinates by processing algorithms. Then, based on predicted graph vertices, they designed an encoding scheme to demonstrate graph edges and the input aerial image was projected to an 18-D tensor by the encoding scheme. A deep neural network was trained to predict the 18-D encoding tensor of the input image, and the graph edges could be calculated by decoding the predicted encoding tensor. Sat2Graph presents quite promising results, but it is not end-to-end trainable, which degrades its final performance. Moreover, the isomorphic encoding issue [15] also restricts Sat2Graph from having better evaluation scores. (3) Iterative-graph-based approaches [17]–[20]. These approaches convert the road network detection task to an MDP problem, in which an agent is trained to detect the road network graph vertex by vertex iteratively. It is believed that RoadTracer proposed by Bastani *et al.* [17] is the first work belonging to this category of approaches. RoadTracer trained a CNN-based decision network to control an agent to explore the road network by iterations. At each step, the network predicted the moving direction of the next step, and the agent moved in the predicted direction by a fixed distance. Inspired by RoadTracer, Xu *et al.* [19] proposed a DETR-like network RNGDet to detect the road network graph. RNGDet achieved the SOTA performance. At each step, RNGDet directly predicted coordinates of vertices in the next step, so that RNGDet can have flexible step length and handle road intersections with arbitrary numbers of incident roads. However, RNGDet only utilized the feature of the deepest layer of the backbone network, leaving multi-scale features not fully used, which prevents further improvement.

C. Detection by Transformer

Compared with CNN (Convolutional Neural Network), transformer [32] can better handle variant-length input, and capture global relationships between patches of the input image. Transformer-based detection framework DETR (Detection by Transformer) was first proposed by Carion *et al.* [21]. Compared with previous detection works, DETR is more simple, effective and end-to-end trainable. Taken as input images, DETR directly outputs a fixed-length vector encoding certain information about each candidate object. By modifying the output vector as well as the transformer network, DETR is adopted to handle various different kinds of detection tasks, such as line segment detection [33] (output vector is the coordinate of line segment endpoints), road centerline detection [34] (output vector is the coordinate of B-spline control points), general graph detection [35] (output vector is the coordinate of vertices and relation of vertices) and road network graph detection [19] (output vector is the coordinate of vertices in the next step). Even if most DETR-like approaches present satisfactory results for a specific task, they only utilize a single feature layer obtained by the backbone network, while the multi-scale feature tensors are not fully made use of. In this paper, RNGDet++ is proposed mainly to conquer this problem by using multi-scale features for both training and inference to further enhance the final performance.

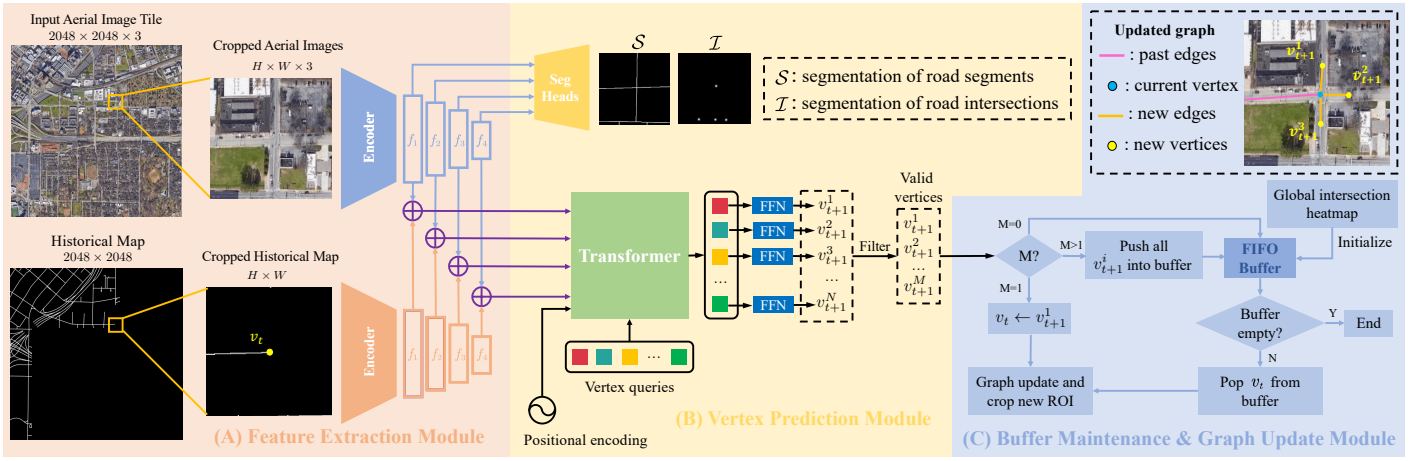

Fig. 1: System diagram of RNGDet++. In this diagram, RNGDet++ conducts a single-step processing at time t. RNGDet++ mainly consists of three modules: (A) Feature extraction module. With the RGB aerial image and the binary historical map as input, this module first crops ROIs centering at v_t, and then extracts multi-scale deep features of ROIs by two backbone networks. (B) Vertex prediction module. Based on extracted multi-scale features, this module predicts two semantic segmentation maps $\mathcal{S}$, $\mathcal{I}$, vertices in the next step $\{v_{t+1}^i\}_{i=1}^M$, as well as instance segmentation maps of all valid vertices $\{(\mathcal{S}_\mathcal{I})_{t+1}^i\}_{i=1}^M$. (C) Buffer maintenance & graph update module. After obtaining vertices in the next step, RNGDet++ updates the graph G, and controls the agent to make corresponding actions. The agent keeps repeating the above steps until the buffer is empty. When the buffer is empty, RNGDet++ stops and outputs the predicted road network graph G. This figure is best viewed in color. Please zoom in for details.

III. Methodology

A. Overview

In this paper, we propose RNGDet++ to detect the graph structure of road networks for downstream autonomous driving applications. Compared with RNGDet, RNGDet++ makes full use of multi-scale backbone features and adds an instance segmentation head to better supervise the training phase. Suppose the road network graph is $G = (V, E)$, where V is a set of key points of the road network as vertices and E is a set of road segments as edges. Taken as input large aerial image tiles, the task of RNGDet++ is predicting the road network graph G. The system diagram of RNGDet++ is visualized in Fig. 1.

RNGDet++ controls an agent to iteratively detect the road network graph, whose current coordinates are denoted by v_t, where t is the current time stamp. Since the input aerial image tile (i.e., I) is usually very large, such as 2048×2048 or 4096×4096, considering limited computation resources, RNGDet++ processes an 128×128 ROI (Region of Interest) at one time. To provide the agent with historical information, the rasterized graph detected by RNGDet++ so far is recorded as the historical map H. H is represented by a binary image whose size is the same as that of I. H is obtained by rasterizing the vector format historical graph into an image so that it can be processed by the CNN backbone network together with I. Centering at v_t, the image ROI (i.e., I_R) and historical map ROI (i.e., H_R) are cropped on I and H, respectively. Taken as input I_R and H_R, RNGDet++ extracts the multi-scale deep features by two ResNet [36] backbone networks. The i-th feature layer is denoted by f_i, and larger i indicates a deeper feature layer.

With a FPN (Feature Pyramid Network) [22] , RNGDet++ predicts the segmentation of road segments (i.e., $\mathcal{S}$) and the segmentation of road intersection points (i.e., $\mathcal{I}$). The DETR-like transformer network makes fully use of multi-scale backbone features, and predicts coordinates (i.e., $\{v_{t+1}^i\}_{i=1}^N$) and valid probability (i.e., $\{p_{t+1}^i\}_{i=1}^N$) of N vertices in the next step. These predicted vertices are then filtered by removing those with low valid probability p_{t+1}^i and RNGDet++ finally obtains M valid vertices in the next step.

RNGDet++ maintains a FIFO (First-in, First-out) buffer saving initial candidates, which are initial vertices to initialize the iteration of the agent. These initial vertices may come from local peaks of the global segmentation heatmap of road intersections, or from breakpoints of the agent iteration. Based on the number of valid vertices M in the next step, the agent takes different actions to update the graph. If $M = 1$, there is only one vertex v_{t+1}^1 in the next step, and the agent directly moves to v_{t+1}^1; if $M = 0$, the agent pops a new initial candidate from the buffer; if $M > 1$, it indicates that road intersections are met, so the agent pushes all $\{v_{t+1}^i\}_{i=1}^M$ into the buffer and pops a new initial candidate from it. After this, the agent crops new ROIs, predicts new vertices and repeats the aforementioned process until the buffer is empty.

B. Feature Extraction

Centering at the current coordinate of the agent (i.e., v_t), RNGDet++ crops I_R and H_R on the input aerial image and the historical map. I_R and H_R provide the agent with visual information and historical information, respectively. RNGDet++ extracts multi-scale deep features of these two ROIs by two ResNet backbone networks. Each ResNet backbone can obtain four layers of features denoted as $f_i, i \in (1, 2, 3, 4)$. A larger i indicates a deeper feature layer. The extracted features obtained by two backbones are added for feature fusion.

C. Multi-scale Feature Fusion

The main difference between RNGDet++ and the previous RNGDet is that RNGDet++ can make use of multi-scale backbone features while RNGDet only utilizes a single feature

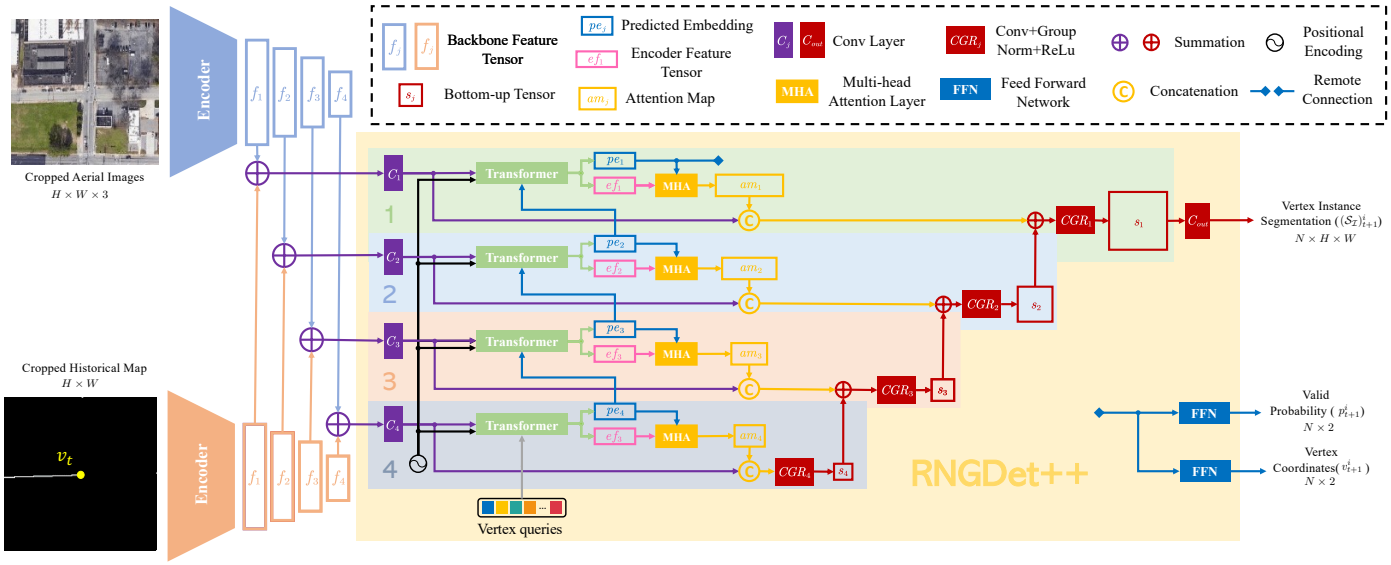

Fig. 2: Network structure of RNGDet++. RNGDet++ can make use of all four levels of features while RNGDet only utilizes the 4-th feature layer. At each layer, the transformer predicts an embedding tensor pe_i (blue box) of each input vertex query, and outputs the feature tensor extracted by the transformer encoder ef_i (pink box). $\{pe_i\}_{i=1}^4$ is summed up and then fed to FFNs (Feed Forward Networks) to predict the coordinate and valid probability. A multi-head attention layer (orange rectangle) is used to predict the attention map am_i (orange box) based on pe_i and ef_i. The predicted attention map is then sent to an FPN (red elements) to predict the instance segmentation map. This figure is best viewed in color. Please zoom in for details.

layer. Suppose the ResNet backbone obtains four levels of features denoted by $\{f_i\}_{i=1}^4$, and larger i indicates a deeper level feature tensor. RNGDet only uses the deepest layer f_4 for vertex prediction while tensors at other levels are ignored, which causes information loss. Inspired by UNet [37] and FPN [22], the proposed RNGDet++ makes predictions based on features extracted at all four levels, which can better capture the deep feature of the input images to predict the vertex in the next step. Please refer to the supplementary document for detailed network comparison of RNGDet and RNGDet++.

Transformers are trained to process multi-scale features. Taken as input f_i and a vertex query, a shared transformer predicts an embedding tensor pe_i. $\{pe_i\}_{i=1}^4$ is summed and sent to an FFN (Feed Forward Network) for the prediction of one vertex in the next step. pe_i is also used to calculate the attention map am_i together with the output of transformer encoder ef_i. Both $\{am_i\}_{i=1}^4$ and $\{f_i\}_{i=1}^4$ are fed to an FPN segmentation head for instance segmentation of road segments ahead. The network structure of RNGDet++ is visualized in Fig. 2.

D. Vertex Prediction

With the extracted fused multi-scale deep features as input, RNGDet++ outputs several predictions by different heads.

1) Semantic Segmentation: RNGDet++ predicts the semantic segmentation maps of road segments (i.e., $\mathcal{S}$) and road intersection points (i.e., $\mathcal{I}$) with a FPN segmentation head. $\mathcal{S}$ helps the network to learn the feature of the road network, while $\mathcal{I}$ enables the network to be better aware of road intersections, which improves the performance of RNGDet++ to detect road networks with complicated intersections. The semantic segmentation head only takes deep features of the input aerial image as input and ignores that of the historical map.

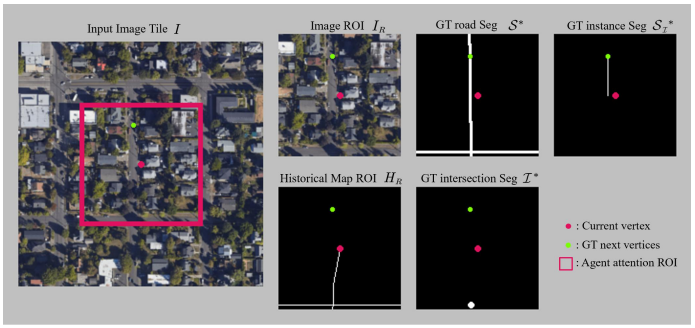

Fig. 3: Visualization of the ground truth instance segmentation mask label (top right mask). When the agent v_t (pink point) is away from the right track, the instance segmentation head still supervises the agent to capture the correct road information, which improves the final performance.

2) Next Step Vertices: Besides fused multi-scale deep features, the DETR-like transformer also takes N vertex queries as input. The vertex query is a fixed-length trainable tensor, which could be treated as a slot that the transformer can utilize to make predictions. For each vertex query, the transformer predicts the x-y coordinate of vertex in the next step v_{t+1}^i, a valid probability p_{t+1}^i, and an instance segmentation map of the ahead road $(\mathcal{S}_\mathcal{I})_{t+1}^i$. If p_{t+1}^i is larger than a threshold, the predicted vertex is classified as valid and will be used to update the graph. Suppose v^* is the vertex projected from a vertex v onto the ground-truth road network. $(\mathcal{S}_\mathcal{I})_{t+1}^i$ is the road segment aheading v_t that connects v_t^* and $(v_{t+1}^i)^*$, which could better supervise the training phase of the network and improve the reasoning ability of the agent. An example is visualized in Fig. 3 to show how we define instance segmentation labels.

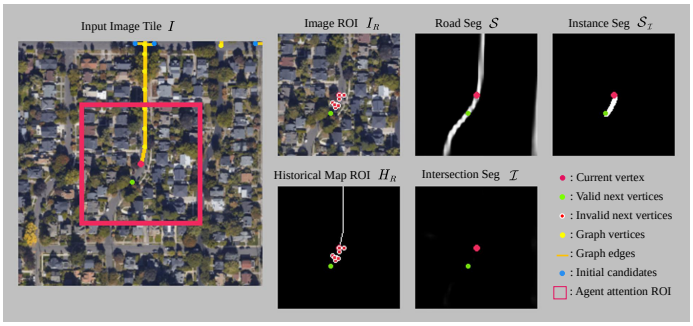

Fig. 4: Demonstrative visualization of the output of RNGDet++. With ROIs I_R and H_R as input, RNGDet++ predicts two semantic segmentation maps $\mathcal{S}$ and $\mathcal{I}$, as well as N vertices in the next step. The prediction of each vertex contains its coordinates, valid probability, and instance segmentation of the road ahead. Vertex with high valid probability is classified as valid vertex (green node), otherwise it is filtered out as invalid (red node with white margin). This figure is best viewed in color. Please zoom in for details.

Since there are N input vertex queries, the transformer predicts N vertices in the next step. After filtering those vertices with low valid probability, we finally have M valid vertices in the next step. The visualization of RNGDet++ outputs is shown in Fig. 4.

E. Buffer Maintenance & graph update

During the iteration of RNGDet++, we maintain a FIFO buffer to save initial candidates. An initial candidate is one initial vertex that the agent starts the iteration. At the very beginning, we predict the global segmentation heatmap of road intersections by merging $\mathcal{I}$ of ROIs that cover the whole I, and push all local peak points of the segmentation map into the buffer as initial candidates. Then, we pop one initial candidate from the buffer, crop ROIs centering at the newly popped initial candidate and predict vertices in the next step. Based on the number of valid predicted vertices M, we take different actions to maintain the buffer and update the graph:

- $M = 0$. No road is ahead. Thus the agent stops processing the current road instance, and pops a new initial candidate from the buffer to work on.
- $M = 1$. A single road is ahead. The agent directly moves to $v^i_{t+1}(i = 1)$ and continues the iteration without operating on the buffer.
- $M > 1$. Multiple roads are met (e.g., road split or road intersections). The agent pushes all v^i_{t+1} into the buffer as new initial candidates, and pops a new vertex from the buffer to work on.

RNGDet++ keeps repeating the aforementioned steps until the buffer is empty. If the buffer is empty, RNGDet++ completes the detection task of the current input aerial image and outputs the predicted G. The working pipeline of RNGDet++ is visualized in module C of Fig. 1.

F. Training Data Sampling

The training of RNGDet++ relies on imitation learning. We create an expert by using the BFS (Breadth First Search) graph traversal algorithm proposed in [19]. The expert can annotate the road network graph vertex by vertex in the same way as human experts. For better robustness, we add even-distributed noise to expert trajectories.

Then the task of RNGDet++ is mimicking the behavior of the expert and trying to learn its policy. In our experiment, considering the training efficiency, we use the behavior-cloning imitation learning algorithm [38] to collect training samples and train RNGDet++.

G. Loss Functions

For semantic segmentation maps $\mathcal{S}$ and $\mathcal{I}$, binary cross entropy loss $\mathcal{L}_{seg}$ is utilized. To conquer the imbalance of foreground pixels and background pixels, a larger training weight is applied for foreground pixels.

Similar to DETR [21], for vertices in the next step, RNGDet++ first matches predictions with the ground truth by conducting the Hungarian algorithm. After matching, vertex coordinate is trained by L1 loss (i.e., $\mathcal{L}_{coord}$), and valid probability is trained by cross-entropy loss (i.e., $\mathcal{L}_{prob}$).

For each valid predicted vertex, we calculate the binary cross entropy segmentation loss (i.e., $\mathcal{L}_{ins}$) to train the instance segmentation head.

Finally, we have the training loss $\mathcal{L} = \mathcal{L}_{seg} + \alpha\mathcal{L}_{coord} + \beta\mathcal{L}_{prob} + \gamma\mathcal{L}_{ins}$.

IV. Experiment

A. Dataset

In this paper, all experiments are conducted on the city-scale dataset released in [15] and the SpaceNet dataset [23]. The city-scale dataset contains 180 2048 pixel×2048 pixel RGB aerial images captured from different cities all over the globe. This dataset provides the ground truth road network in the format of vector graphs. Following Sat2Graph [15], we split the dataset into train/valid/test with 144/9/27 tiles, respectively. The SpaceNet dataset has 2551 400 pixel×400 pixel RGB aerial images. Compared with the city-scale dataset, SpaceNet dataset focuses on smaller regions. The dataset is split into train/valid/test with 2042/127/382 images, respectively. Images of both datasets have 1m/pixel resolution ratio. These two datasets are large and have aerial images collected from various scenarios, thus making them sufficient to train road network graph detection approaches and comprehensively evaluate them.

B. Baselines and Evaluation Metrics

We compare RNGDet++ with previous SOTA approaches, including four segmentation-based approaches and three graph-based approaches.

- Segmentation-based baselines [8], [9], [15], [37]. Unet [37] is a widely used classic semantic segmentation network, we adopt it to our task. DRM (Deep Road Mapper) [9] and ImprovedRoad [8] conduct post-processing steps to refine the road network segmentation results, which achieves better performance. DLA (Deep Layer Aggregation) [39] has more powerful backbone network, which is also used by Sat2Graph [15].

(a) Ground truth (b) Sat2Graph [15] (c) RNGDet [19] (d) RNGDet++

Fig. 5: Qualitative visualization on the city-scale dataset. (a) Ground truth road networks (Cyan lines). (b) Road network graph detected by Sat2Graph. (c) Road network graph detected by RNGDet. (d) Road network graph detected by RNGDet++. For (b)-(d), yellow points represent graph vertices and orange lines represent graph edges. For the visualization, it is found that RNGDet++ can output road network graphs with more accurate structure and correctness compared with previous works. This figure is best viewed in color. Please zoom in for details.

- Graph-base baselines [15], [17], [19]. RoadTracer [17] is believed to be the first graph-based approach for the road network detection task. Sat2Graph and RNGDet are two SOTA approaches that can directly output the graph structure of road networks.

All approaches are evaluated based on metrics used in [15], including TOPO [40] and APLS (Average Path Length Similarity) [23]. Within the input aerial image, TOPO first randomly samples seed vertices on the ground truth graph and the predicted one, then compares the similarity of sub-graphs that seed vertices can reach. This metric uses precision, recall and f1 to measure the average sub-graph similarity. APLS randomly samples a vertex pair (v_1, v_2) on the ground truth graph and projects them to the predicted graph as $(\hat{v}_1, \hat{v}_2)$. Then APLS compares the shortest distance between (v_1, v_2) and $(\hat{v}_1, \hat{v}_2)$. Smaller distance difference means better graph similarity. For both metrics, larger scores indicate better performance.

C. Implementation Details

In our experiments, all ROIs are 128×128-sized. We run the sampling algorithm to collect the training data. We finally obtain around 400K samples to train RNGDet++. During the training phase, the loss weights α, β and γ are set to 5, 1 and 1, respectively. Considering the road topology, we set the number of vertex queries as 10, which is sufficient to handle any common road networks. RNGDet++ is trained for 50 epochs, with 10^{-4} initial learning rate. All experiments are conducted on 4 RTX-3090 GPUs.

In our experiments, all approaches are tuned on the validation set of the corresponding dataset. We aim to maximize the TOPO-F1 score of the validation set by trying different parameter settings of the evaluated approach and then use the tuned model to infer the test set. APLS is not considered during the parameter tuning process.

TABLE I: The quantitative comparison results. The best results are highlighted in bold font. For all the metrics, larger values indicate better performance.

Methods	City-scale Dataset				SpaceNet Dataset			
	Prec. ↑	Rec. ↑	F1↑	APLS ↑	Prec. ↑	Rec. ↑	F1↑	APLS ↑
Seg-UNet [37]	75.34	65.99	70.36	52.50	68.96	66.32	67.61	53.77
Seg-DRM [9]	76.54	71.25	73.80	54.32	82.79	72.56	77.34	62.26
Seg-Improved [8]	75.83	68.90	72.20	55.34	81.56	71.38	76.13	58.82
Seg-DLA [39]	75.59	72.26	73.89	57.22	78.99	69.80	74.11	56.36
RoadTracer [17]	78.00	57.44	66.16	57.29	78.61	62.45	69.90	56.03
Sat2Graph [15]	80.70	72.28	76.26	63.14	85.93	**76.55**	80.97	64.43
RNGDet [19]	**85.97**	69.78	76.87	65.75	90.91	73.25	81.13	65.61
RNGDet++	85.65	**72.58**	**78.44**	**67.76**	**91.34**	75.24	**82.51**	**67.73**

TABLE II: The quantitative results for the ablation study. The best results are highlighted in bold font. For all the metrics, larger values indicate better performance. We assess the instance segmentation head (I) and the multi-scale features (M).

I	M	City-scale Dataset				SpaceNet Dataset			
		Prec. ↑	Rec. ↑	F1↑	APLS ↑	Prec. ↑	Rec. ↑	F1↑	APLS ↑
	✓	**86.04**	70.65	77.94	66.36	90.74	75.10	82.18	67.21
✓		85.62	71.88	78.01	66.94	**91.46**	75.11	82.48	67.01
✓	✓	85.65	**72.58**	**78.44**	**67.76**	91.34	**75.24**	**82.51**	**67.73**

TABLE III: The time usage for model inference. We report the time used (hours) to infer all testing images.

	Sat2Graph	RNGDet	RNGDet++
City-scale Dataset	2.51h	2.68h	3.85h
SpaceNet Dataset	1.15h	1.22h	1.88h

D. Comparison Experiments

In evaluation, RNGDet++ is compared with seven baseline approaches, including four segmentation-based approaches and three graph-based baselines. The quantitative results of the comparison experiment are shown in Tab. I. Qualitative demonstrations on the city-scale dataset are visualized in Fig. 5.

Segmentation-based approaches first predict the pixel-level semantic segmentation map of road networks, and then conduct post-processing algorithms to extract and refine the graph structure of road networks. From Tab. I,We can see that they tend to have relatively good TOPO scores since TOPO mainly measures the performance of the sub-graph detection, which focuses on the locality. Since segmentation-based approaches directly optimize pixels, they can achieve satisfactory results on pixel-level or local topology-level metrics. However, they have a degraded APLS score mainly because their global topology is not good enough, which could be caused by incorrect detection of road intersections or overlapped overpasses. Therefore, we claim that segmentation-based approaches can detect the road network graph with satisfactory local performance, but cannot present good results on the global scale.

Different from the aforementioned segmentation-based approaches, graph-based approaches directly output and optimize the graph structure of road networks. Thus they usually have better topology-level performance. Sat2Graph and RNGDet present the SOTA performance among baselines, and present superior results not only in TOPO metrics but also in the APLS metric. RNGDet++ is enhanced from RNGDet by utilizing multi-scale deep features, and it has the best evaluation scores on both APLS and TOPO metrics. Therefore, the superiority and effectiveness of RNGDet++ are well demonstrated and verified.

E. Ablation Studies

We conduct ablation studies to verify the rationality of the design of RNGDet++, including the instance segmentation head and the multi-layer features. The quantitative results of ablation studies are shown in Tab. II.

First, the instance segmentation head is removed from RNGDet++. The instance segmentation head is used to predict road segments ahead of v_t, which can better supervise the training of RNGDet++, making it better capture the spatial and topology information of the road network graph. Based on the evaluation scores, RNGDet++ without the instance segmentation head presents inferior results. Therefore, the importance of the instance segmentation head is well demonstrated.

Then, to learn how multi-layer features affect the performance of RNGDet++, we train RNGDet++ by only utilizing the deepest feature layer (i.e., f_4). From the evaluation results, we find that RNGDet++ without utilizing multi-layer features cannot reach as good results as the original RNGDet++. Thus, the necessity of using multi-scale features is verified.

F. Limitations

1) Lower efficiency: Since at each layer, the transformer makes one prediction based on fused features, the time used for the inference of RNGDet++ is relatively longer than RNGDet. The inference time usage is reported in Tab. III. However, it should be noted that our task (i.e., road network graph detection) is an offline task, which is not sensitive to efficiency. Thus, considering the superior effectiveness performance of RNGDet++, we think the relatively lower efficiency is acceptable at this stage. We plan to further simplify the network and replace the model modules with lighter structures.

2) Too complicated intersection and overpass: Even though RNGDet++ outperforms previous works and presents the best final results, it still cannot handle too complicated road intersections or overpasses very well. And since RNGDet++ is trained by imitation learning, incorrect predictions in these scenes may affect the afterward behaviors of the agent. We plan to further optimize the training strategy, and use more powerful backbone networks to improve the reasoning ability of RNGDet++.

V. Conclusion

In this paper, we enhanced the previous state-of-the-art road network graph detection approach RNGDet by adding an additional instance segmentation head and enabling it to leverage multi-scale features of the backbone network. The new novel approach was named RNGDet++. The instance segmentation head could better supervise the network training and improve the reasoning ability of RNGDet++. Besides, RNGDet++ could utilize all layers of features obtained by the backbone network, which enabled the network to capture multi-scale information

of the road network so that RNGDet++ presented superior results. RNGDet++ was trained by the behavior-cloning imitation learning algorithm. RNGDet++ and all baselines were fairly evaluated on two large publicly available datasets. Compared with all baselines, RNGDet++ achieved better evaluation scores, not only at pixel level but also at topology level. In the future, we plan to further simplify the network structure for efficiency, and apply more powerful backbone networks to better capture the feature of road networks.

REFERENCES

[1] T. Liu, Q. hai Liao, L. Gan, F. Ma, J. Cheng, X. Xie, Z. Wang, Y. Chen, Y. Zhu, S. Zhang *et al.*, "The role of the hercules autonomous vehicle during the covid-19 pandemic: An autonomous logistic vehicle for contactless goods transportation," *IEEE Robotics & Automation Magazine*, vol. 28, no. 1, pp. 48–58, 2021.

[2] H. Christensen, D. Paz, H. Zhang, D. Meyer, H. Xiang, Y. Han, Y. Liu, A. Liang, Z. Zhong, and S. Tang, "Autonomous vehicles for micro-mobility," *Autonomous Intelligent Systems*, vol. 1, no. 1, pp. 1–35, 2021.

[3] V. Mnih and G. E. Hinton, "Learning to detect roads in high-resolution aerial images," in *European Conference on Computer Vision*. Springer, 2010, pp. 210–223.

[4] X. Hu, Y. Li, J. Shan, J. Zhang, and Y. Zhang, "Road centerline extraction in complex urban scenes from lidar data based on multiple features," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 11, pp. 7448–7456, 2014.

[5] W. Shi, Z. Miao, Q. Wang, and H. Zhang, "Spectral–spatial classification and shape features for urban road centerline extraction," *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 4, pp. 788–792, 2013.

[6] C. Unsalan and B. Sirmacek, "Road network detection using probabilistic and graph theoretical methods," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 50, no. 11, pp. 4441–4453, 2012.

[7] G. Cheng, F. Zhu, S. Xiang, and C. Pan, "Road centerline extraction via semisupervised segmentation and multidirection nonmaximum suppression," *IEEE Geoscience and Remote Sensing Letters*, vol. 13, no. 4, pp. 545–549, 2016.

[8] A. Batra, S. Singh, G. Pang, S. Basu, C. Jawahar, and M. Paluri, "Improved road connectivity by joint learning of orientation and segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 385–10 393.

[9] G. Máttyus, W. Luo, and R. Urtasun, "Deeproadmapper: Extracting road topology from aerial images," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3438–3446.

[10] A. V. Etten, "City-scale road extraction from satellite imagery v2: Road speeds and travel times," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2020, pp. 1786–1795.

[11] W. Gedara Chaminda Bandara, J. M. J. Valanarasu, and V. M. Patel, "Spin road mapper: Extracting roads from aerial images via spatial and interaction space graph reasoning for autonomous driving," *arXiv e-prints*, pp. arXiv–2109, 2021.

[12] G. Cheng, Y. Wang, S. Xu, H. Wang, S. Xiang, and C. Pan, "Automatic road detection and centerline extraction via cascaded end-to-end convolutional neural network," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 6, pp. 3322–3337, 2017.

[13] G. Zhou, W. Chen, Q. Gui, X. Li, and L. Wang, "Split depth-wise separable graph-convolution network for road extraction in complex environments from high-resolution remote-sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, 2021.

[14] Z. Xu, Y. Liu, L. Gan, X. Hu, Y. Sun, L. Wang, and M. Liu, "csboundary: City-scale road-boundary detection in aerial images for high-definition maps," *arXiv preprint arXiv:2111.06020*, 2021.

[15] S. He, F. Bastani, S. Jagwani, M. Alizadeh, H. Balakrishnan, S. Chawla, M. M. Elshrif, S. Madden, and M. A. Sadeghi, "Sat2graph: road graph extraction through graph-tensor encoding," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXIV 16*. Springer, 2020, pp. 51–67.

[16] G. Bahl, M. Bahri, and F. Lafarge, "Single-shot end-to-end road graph extraction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1403–1412.

[17] F. Bastani, S. He, S. Abbar, M. Alizadeh, H. Balakrishnan, S. Chawla, S. Madden, and D. DeWitt, "Roadtracer: Automatic extraction of road networks from aerial images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4720–4728.

[18] Y.-Q. Tan, S.-H. Gao, X.-Y. Li, M.-M. Cheng, and B. Ren, "Vecroad: Point-based iterative graph exploration for road graphs extraction," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8910–8918.

[19] Z. Xu, Y. Liu, L. Gan, Y. Sun, X. Wu, M. Liu, and L. Wang, "Rngdet: Road network graph detection by transformer in aerial images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–12, 2022.

[20] Z. Li, J. D. Wegner, and A. Lucchi, "Topological map extraction from overhead images," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 1715–1724.

[21] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European Conference on Computer Vision*. Springer, 2020, pp. 213–229.

[22] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[23] A. Van Etten, D. Lindenbaum, and T. M. Bacastow, "Spacenet: A remote sensing dataset and challenge series," *arXiv preprint arXiv:1807.01232*, 2018.

[24] J. Liang, N. Homayounfar, W.-C. Ma, S. Wang, and R. Urtasun, "Convolutional recurrent network for road boundary extraction," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9512–9521.

[25] Z. Xu, Y. Sun, and M. Liu, "Topo-boundary: A benchmark dataset on topological road-boundary detection using aerial images for autonomous driving," *IEEE Robotics and Automation Letters*, vol. 6, no. 4, pp. 7248–7255, 2021.

[26] Z. Xu, Y. Sun, and M. Liu, "icurb: Imitation learning-based detection of road curbs using aerial images for autonomous driving," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 1097–1104, 2021.

[27] Z. Xu, Y. Sun, L. Wang, and M. Liu, "Cp-loss: Connectivity-preserving loss for road curb detection in autonomous driving with aerial images," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2021, pp. 1117–1123.

[28] N. Homayounfar, W.-C. Ma, S. Kowshika Lakshmikanth, and R. Urtasun, "Hierarchical recurrent attention networks for structured online maps," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3417–3426.

[29] N. Homayounfar, W.-C. Ma, J. Liang, X. Wu, J. Fan, and R. Urtasun, "Dagmapper: Learning to map by discovering lane topology," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 2911–2920.

[30] Q. Li, Y. Wang, Y. Wang, and H. Zhao, "Hdmapnet: A local semantic map learning and evaluation framework," 2021.

[31] Z. Xu, Y. Liu, Y. Sun, M. Liu, and L. Wang, "Centerlinedet: Road lane centerline graph detection with vehicle-mounted sensors by transformer for high-definition map creation," *arXiv preprint arXiv:2209.07734*, 2022.

[32] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[33] Y. Xu, W. Xu, D. Cheung, and Z. Tu, "Line segment detection using transformers without edges," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4257–4266.

[34] Y. B. Can, A. Liniger, D. P. Paudel, and L. Van Gool, "Structured bird's-eye-view traffic scene understanding from onboard images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 15 661–15 670.

[35] S. Shit, R. Koner, B. Wittmann, J. Paetzold, I. Ezhov, H. Li, J. Pan, S. Sharifzadeh, G. Kaissis, V. Tresp *et al.*, "Relationformer: A unified framework for image-to-graph generation," *arXiv preprint arXiv:2203.10202*, 2022.

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

[37] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[38] T. Osa, J. Pajarinen, G. Neumann, J. A. Bagnell, P. Abbeel, and J. Peters, "An algorithmic perspective on imitation learning," *arXiv preprint arXiv:1811.06711*, 2018.

[39] F. Yu, D. Wang, E. Shelhamer, and T. Darrell, "Deep layer aggregation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 2403–2412.

[40] J. Biagioni and J. Eriksson, "Inferring road maps from global positioning system traces: Survey and comparative evaluation," *Transportation research record*, vol. 2291, no. 1, pp. 61–71, 2012.