TopDiG: Class-agnostic Topological Directional Graph Extraction from Remote Sensing Images

Bingnan Yang[1] , Mi Zhang[1†] , Zhan Zhang[2] , Zhili Zhang[1] , Xiangyun Hu[1]

[1] School of Remote Sensing and Information Engineering, Wuhan University, China
[2] State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, China

[†] Corresponding author { mizhang@whu.edu.cn }

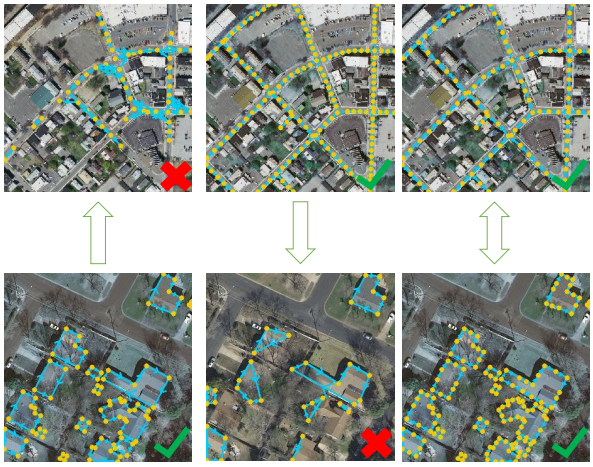
Abstract

Rapid development in automatic vector extraction from remote sensing images has been witnessed in recent years. However, the vast majority of existing works concentrate on a specific target, fragile to category variety, and hardly achieve stable performance crossing different categories. In this work, we propose an innovative class-agnostic model, namely TopDiG, to directly extract topological directional graphs from remote sensing images and solve these issues. Firstly, TopDiG employs a topology-concentrated node detector (TCND) to detect nodes and obtain compact perception of topological components. Secondly, we propose a dynamic graph supervision (DGS) strategy to dynamically generate adjacency graph labels from unordered nodes. Finally, the directional graph (DiG) generator module is designed to construct topological directional graphs from predicted nodes. Experiments on the **Inria**, **CrowdAI**, **GID**, **GF2** *and* **Massachusetts** *datasets empirically demonstrate that TopDiG is class-agnostic and achieves competitive performance on all datasets.*

1. Introduction

Vector maps that are represented as topological directional graphs act as the foundation to various remote sensing applications, such as property mapping, cartographic generalization and disaster assessment [21, 25]. Traditional manual or semi-automatic vector map generation from remote sensing images is extremely time-consuming and expensive. In contrast, state-of-the-art approaches, including segmentation-based [4, 9, 13, 30], contour-based [1, 16, 22, 29, 35, 39] and graph generation [3, 26, 31–34, 41] methods have typically developed to achieve automation. However, these works are concentrated on a specific category and can hardly achieve satisfactory performance when applied to other classes.

Among aforementioned approaches, a dominant paradigm is the *segmentation-based* method. It follows the segmentation-vectorization pipeline and requires sophisticated vectorization procedures to binary masks. Typical examples that fall in this scope include PolyMapper [14], Frame Field [9] and ASIP [13]. These methods retrieve coarse raster maps with missing details and unavoidably demand elaborate post-processing. Another paradigm mainly adopts *contour-based* instance segmentation approaches, which usually refine initial contours to obtain vector maps. For instance, Polygon-RNN [6], Polygon-RNN++ [1], Deep Snake [22], SharpContour [39] and E2EC [35] can delineate the outlines of polygon-shape targets, such as buildings and water bodies. However, it depends on the quality of initial contours and can barely be reliable on line-shape targets, such as roads. The flourishing *graph generation* methods construct the topological graph based on nodes and their connectivity. A few of such approaches,



(a) PolyWorld (b) Enhanced iCurb (c) Ours

Figure 1. **Visual illustrations of current and our approaches on different targets**. In contrast with PolyWorld and Enhanced iCurb which only serve a specific category, TopDiG is class-agnostic and can tackle both polygon-shape and line-shape targets. Yellow dots refer to detected nodes while blue arrowed lines indicate the directional topological connection between node pairs (best viewed by zooming in).

including RoadTracer [3], VecRoad [26], iCurb [33] and RNGDet [32] focus on line-shape targets by iteratively predicting nodes in a one-by-one manner. Nevertheless, these methods suffer from the low efficiency, the accumulated node connectivity error, and the poor reliability to polygon-shape targets. Alternatively, the connectivity of the nodes also can be recovered from adjacency matrix as introduced in PolyWorld [41] and csBoundary [31]. These workflows successfully produce visually pleasing vector topology without irregular edges and overly smoothed corners. Unfortunately, for intricate or sinuous structures, such methods lead to severe topological errors.

Given the class-dependent characteristics, existing works can hardly apply to other classes as illustrated in Figure 1. For example, PolyWolrd [41] (Figure 1(a)) is able to extract well-vectorized buildings but fails to delineate road networks. By contrast, Enhanced iCurb [34], originally concerning line-shape road curbs, is challenged by polygon-shape buildings (Figure 1(b)). They adopt the similar scheme where topological graphs are constructed by connecting detected nodes and aim at either polygon-shape or line-shape targets, respectively. Nevertheless, neither of them can achieve stable and reliable topological directional graphs regardless of varying categories. In this work, we propose a class-agnostic approach named TopDiG which can robustly obtain precise topological directional graphs both for polygon-shape and line-shape targets (Figure 1(c)). The underlying innovation is that the TopDiG formulates diverse topological structures as directional graphs and narrows the gap among categorical varieties. Besides, we further develop a dynamic graph supervision strategy that enables flexible arrangement of the predicted nodes and stabilizes performance crossing different categories. Our contributions are summarized as follows:

A *Dynamic Graph Supervision* (DGS) strategy is designed to generate the ground truth of adjacency matrix in an on-the-fly manner during training. Instead of utilizing the adjacency matrix labels established from ordered ground truth nodes [31, 36, 41], we dynamically generate such labels according to real-time unordered predict nodes in each training epoch. Our strategy alleviates the compulsory assumption that the sequence of predict nodes must be in consistent with real ones as in PolyWorld [41]. Consequently, DGS can facilitate the connectivity of unordered nodes and ease the demand for the accurate positions of nodes. We further propose a novel topology-concentrated node detector (TCND) to guarantee an appropriate density of predicted nodes. Unlike PolyWorld [41] and csBoundary [31] that mainly employ semantic contexts, TCND concentrates on compact geometric textures via the meticulous perception of topological components, which boosts the topological **APLS** score by approximately 8.06%.

A *Class-agnostic Topological Directional Graph Extraction* (TopDiG) approach is proposed to extract polygon-shape and line-shape targets, i.e., buildings, water bodies and roads, from remote sensing images. In contrast with existing approaches that can only serve a specific category, TopDiG directly performs class-independent vector map generation from diverse targets. We introduce TCND and directional graph (DiG) generator module to retain the geometrical shapes, i.e., polygon-shape and line-shape targets. Our method is performed in an end-to-end manner and does not require initial contours or additional post-processing. The TopDiG outperforms the segmentation-based, contour-based and previous graph generation approaches, achieving a competitive performance with boundary mIoU scores of 68.39%, 72.51%, 74.51% and 75.28% on **Inria**, **CrowdAI**, **GID** and **GF2** datasets, respectively. Moreover, TopDiG can construct reliable topological directional graphs, with the application to **Massachusetts** dataset, achieving an average path length score (**APLS**) of 64.60%.

2. Related Work

Segmentation-based methods. Conventional pipeline for segmentation-based methods, such as TDAC [10], ASIP [13], Frame Field [9] and BT-RoadNet [38], utilizes a two-step strategy to vectorize the segmentation masks. They first obtain the irregular binary mask and subsequently achieve the topological graph by simplifying the raster map. Wei *et al.* [30] obtained polygon-shape buildings from binary masks by the Douglas-Peucker algorithm [8] and refine these polygons with handcraft post-processing. ASIP [13] method designs sophisticated post-processing to polygonize low-complexity buildings at a cost of low efficiency. TDAC [10] incorporates the active contour model (ACM) to refine initial segmentation outlines of buildings in an end-to-end fashion and reports superior accuracy over past approaches. Girard *et al.* [9] employed a learnable frame field and proposed active skeleton model (ASM) to regularize coarse raster buildings, leading to more regular outlines. Zhou *et al.* [38] followed a coarse-to-fine framework to improve contiguousness of extracted road network masks. Some other studies dedicate to vectorization techniques, such as GGT model [5] which employs self-attention mechanism to generate vectorized roads from binary masks. However, these methods usually overly depend on correctness of predicted segmentation maps, require dedicated post-processing steps and suffer from serious topological errors when applied to complicated boundaries [31].

Contour-based methods. Unlike the previous segmentation-based approaches, contour-based methods directly extract vector topology of targets from input images. In general, initial contours are first obtained by object detectors or segmentation methods. Subsequently, final topological graphs are refined from initial contours. Ear-

ly works usually design a fixed template of the initial contour for each instance. For example, focusing on natural images, semi-automatic method Curve-GCN [16] uses manually drawn circles surrounding objects as initial contours and then trains a graph convolution network (GCN) to refine them. Wei *et al.* [29] extended CurveGCN to the vector extraction of building boundaries from aerial images by replacing manually drawn circles with predicted object bounding boxes. DeepSnake [22] transforms object bounding boxes to octagons and achieves better accuracy. Recent works replace constant handcraft templates of initial contours with coarse contours to enhance the performance of refinement. For instance, Zhang *et al.* [35] proposed E2EC workflow that gradually refines segmentation coarse boundaries and outperforms past approaches. SharpContour [39] iteratively deforms categorical nodes sampled from raster contours to obtain sharper object outlines. Suffering from unavoidable dependence in initial contours, these contour-based methods can scarcely apply to the line-shape targets, such as roads. Besides, the deformation paths of the contours seriously pare the initialization strategy, which reduces the upper bound of the performance.

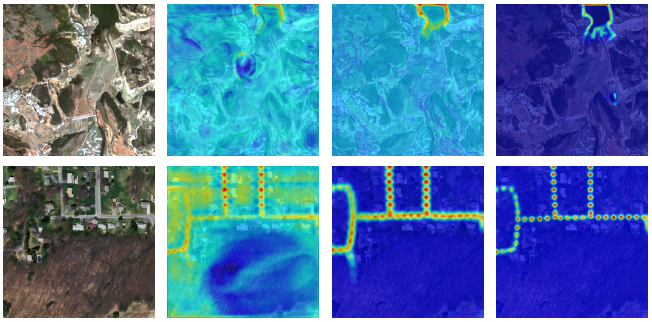
Graph generation methods. Approaches of this category extract vector topology by predicting nodes and their connections. A portion of them concentrate on delineating road graphs by exploring sequential nodes. RoadTracer [3] iteratively moved a fixed distance from current nodes based on predicted directions and action decisions started by a manually selected node. VecRoad [26] employs auto-selected starting nodes and flexible step distance to reduce topology errors and human laboring. RNGDet [32] employs Transformer [28] and can predicts arbitrary number of adjacencies of the current nodes to increase the efficiency of training and inference. Another portion of graph generation methods firstly extracts nodes of targets and then connects possible node pairs. APGA [40], aiming to extract building boundaries, successfully learns a direction map to construct the relationships among building corners. PPGNet [36] learns an adjacency matrix of predicted junctions by a convolution neural network (CNN) to infer topological graphs of line segments and is capable of handling multiple instances simultaneously. Instead of CNN, Transformer [28] is introduced to predict the adjacency matrix in csBoundary [31] and achieves better performance in road boundary detection from city-scale aerial images. PolyWorld [41] concatenates predicted clockwise and counter-clockwise adjacency matrices of building corners and employs Sinkhorn algorithm [7, 23, 24] to produce final graphs. Nevertheless, existing graph generation approaches mainly consider a specific class and few works can resist categorical varieties. Instead, we propose TopDiG, a class-agnostic framework that achieves reliable topological directional graphs crossing different categories.

3. Our Approach

As illustrated in Figure 2, the TopDiG can directly extract topological directional graphs from remote sensing images. TopDiG is trained in an end-to-end manner, which consists of TCND, DGS and DiG generator modules. In Section 3.1, we introduce TCND that focuses on concrete geometric textures to facilitate the detection of nodes and meticulous feature perception of topological components. Section 3.2 introduces DGS strategy that dynamically generates graph labels and is conducted in an on-the-fly manner. We further introduce the DiG generator (Section 3.3) which constructs the directional topological graph with self-attention network.

3.1. Topology-Concentrated Node Detector

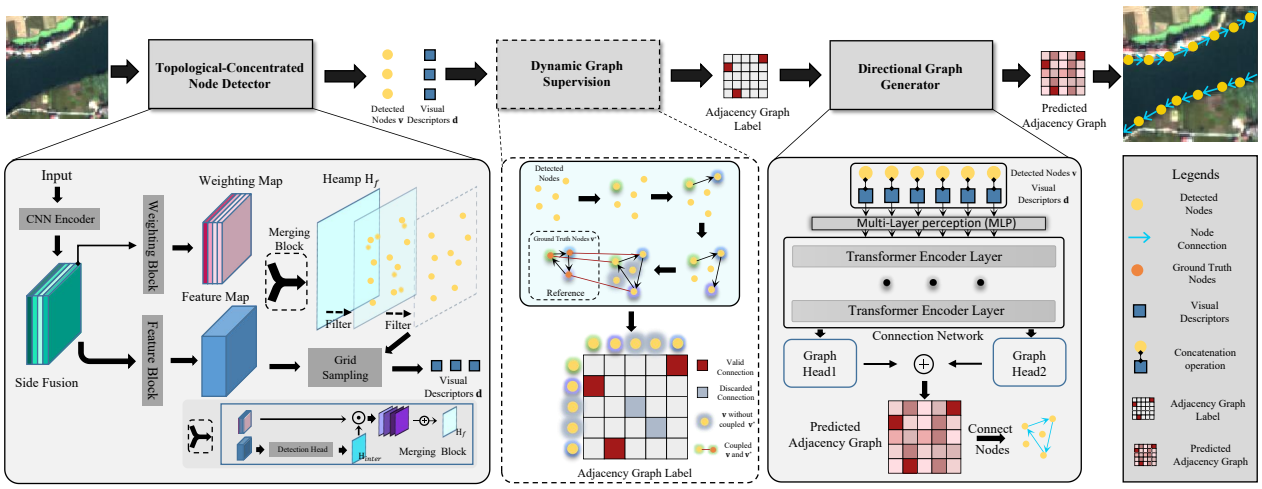
The detection of nodes requires a compact representation of topological components to resist semantic turbulence from complicated contexts in remote sensing images. Traditionally, the previous works, such as PolyWolrd [41] and csBoundary [31], utilize R2UNet [2] and FPN [15] to guarantee the compactness of the nodes. These methods are originate from semantic segmentation networks which overly emphasize semantic contexts instead of topological components. Figure 3 presents the attentive maps on different categories including water bodies and roads. R2UNet and FPN can perceive semantic contexts but neglect topological components, such as water boundaries and road centerlines. Inspired by DFF [12], we design the TCND that can concentrate on topological components and provide compact responses on the potential node positions. The TCND guarantees the density and topology of these nodes and thereby it can tackle varying classes of targets.



(a) Image (b) FPN (c) R2UNet (d) TCND

Figure 3. **Visual comparison of attentive maps** for FPN, R2UNet and TCND on different categories. The topological components can be well perceived by TCND.

Figure 2(a) depicts the architecture of TCND. TCND receives an input image $\mathbf{I}^{C\times H\times W}$, where H and W are the height and width of the image and C is the number of channels. Firstly, the image is processed by a CNN encoder which contains four stages and each stage yields a sideout heatmap with the size of $H \times W$. All sideouts are



(a) TCND (b) DGS (c) DiG generator

Figure 2. **The pipeline of TopDiG**. Given an input image, TCND extracts nodes and visual descriptors. DGS dynamically generates adjacency graph labels in an on-the-fly manner. DiG generator predicts adjacency graphs recording connectivity of nodes. The top row presents the entire pipeline of TopDiG; The bottom row illustrates details of TCND, DGS and DiG generator.

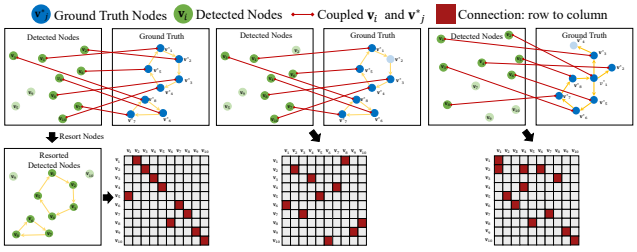
then concatenated as the side fusion $\mathbf{S}^{4\times H\times W}$ which is fed into a feature block. This block enhances low-level topological features and produces a D-dimensional feature map $\mathbf{F}^{D\times H\times W}$. Besides, a weighting block receives the last sideout and yields the weighting map $\mathbf{W}^{4\times H\times W}$ to balance the semantic contexts. After that a merging block is conducted to fuse the weighting map and feature map. The feature map is decoded by a detection head and obtains the intermediate heatmap $\mathbf{H}_{inter}{}^{1\times H\times W}$. Next, the weighting map $\mathbf{W}^{4\times H\times W}$ is multiplied by $\mathbf{H}_{inter}{}^{1\times H\times W}$ and added up along the channel dimension to yield the heatmap $\mathbf{H}_f{}^{1\times H\times W}$.

Subsequently, the heatmap $\mathbf{H}_f{}^{1\times H\times W}$ is filtered by a non-maximum suppression (NMS) algorithm [41] to extract N nodes donated as $\mathbf{v} = \{\mathbf{v}_i \mid i = 1, 2, ..., N\}$, $\mathbf{v} \in \mathbb{R}^{N\times 2}$ where each node $\mathbf{v}_i = \{(x_i, y_i)\}$. Unlike previous approaches that simply extract N most relevant peaks from heatmap, we adopt an additional distance tolerance Φ to force the N nodes in an appropriate density. This strategy eliminates the overly clustered nodes and thereby topological components can be well preserved. Based on obtained nodes, visual descriptors $\mathbf{d} = \{\mathbf{d}_i \mid i = 1, 2, ..., N\}$, $\mathbf{d} \in \mathbb{R}^{N\times D}$ (each $\mathbf{d_i} \in \mathbb{R}^D$) that capture local features are extracted from $\mathbf{F}^{D\times H\times W}$ by grid sampling method [22, 31, 35, 41]. The $\mathbf{v}$ is employed to generate adjacency graph labels in DGS module (see Section 3.2) while the $\mathbf{d}$, together with $\mathbf{v}$ are fed into DiG generator (see Section 3.3) to predict node connections. The predicted heatmap is regressed with the mean square error (MSE) loss:

$$\mathcal{L}_{\mathbf{node}} = \mathcal{M}(\mathbf{h} - \bar{\mathbf{h}})^{\mathbf{2}}, \tag{1}$$

where $\mathcal{M}(\bullet)$ donates absolute mean, $\bar{\mathbf{h}}$ represents ground truth heatmap while the $\mathbf{h}$ refers to the predicted one.

3.2. Dynamic Graph Supervision



(a) Previous works (b) DGS for polygons (c) DGS for lines

Figure 4. **Visual comparison of the adjacency graph label generated by the previous works and DGS**. (a) Previous works generate the adjacency graph labels from ordered ground truth nodes and cluster the topological connections near diagonal, degenerating their resistance to categorical varieties. By contrast, the DGS illustrated in (b) and (c) makes full use of the unordered detected nodes to dynamically establish the adjacency graph labels, which augments the class-agnostic ability on both polygon-shape and line-shape targets.

Conventional approaches [31, 41] generate the adjacency graph label from ordered ground truth nodes. They resort the order of predicted nodes to be consistent with the ground truth, which cripples the ability to tackle nodes with an unpredictable order in practice. Alternatively, we propose to utilize an order-independent adjacency graph that is dynamically generated from detected nodes in an on-the-fly manner. As shown in Figure 2(b), given the unordered extracted nodes $\mathbf{v}$ and ordered ground truth nodes $\mathbf{v}^* = \{\mathbf{v}_j{}^* \mid j = 1, 2, ..., N\}$ where each node $\mathbf{v}_j{}^* = \{(x_j{}^*, y_j{}^*)\}$, we adopt $\mathbf{v}_j{}^*$ as the reference to iteratively find the nearest $\mathbf{v}_i$ and the matching relationship is exclusive. Based on the matched pairs, we construct the adjacency graph label from extracted nodes $\mathbf{v}$ without rearranging their order. Those uncoupled nodes are recorded in the diagonal of adjacency

graph. This adjacency graph is served as supervision of the DiG generator. As illustrated in Figure 4, our DGS retains the order of predicted nodes and dynamically produces adjacency graph labels. Figure 4(a) and Figure 4(b) both aim at polygon-shape targets, previous works [31, 41] generate labels with overly clustered distribution while DGS produces randomly scattered ones. Moreover, DGS can apply the same scheme to different categories (Figure 4(b) and Figure 4(c)), which permits class-agnostic applications on topological directional graph extraction.

3.3. Directional Graph Generator

DiG generator predicts a directional adjacency graph to connect extracted nodes. We design a transformer-based DiG generator that can capture long-term dependencies among numerous nodes. This enables each node to search potential adjacencies in the entire node set. Consequently, DiG generator can seek out sufficient node connections to splice the topological graphs.

The pipeline of the DiG generator is illustrated in Figure 2(c). Receiving visual descriptors $\mathbf{d}$ and detected nodes $\mathbf{v}$, the DiG generator firstly concatenates each coupled $\mathbf{d}_i$ and $\mathbf{v}_i$ to embedded descriptors $\mathbf{d}_{emb} \in \mathbb{R}^{N\times(D+2)}$. A multi-layer perception (MLP) is then utilized to encode the $\mathbf{d}_{emb}$ and produce the D'-dimensional initial descriptors $\mathbf{d}_{init} \in \mathbb{R}^{N\times D'}$. Afterwards, the $\mathbf{d}_{init}$ is fed into a connection network that consists of M transformer encoder layers to yield final descriptors $\mathbf{d}_{final} \in \mathbb{R}^{N\times D'}$. Following the common practice [31, 32, 41], we adopt the self-attention transformer as below:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V, \quad (2)$$

where Q, K and V are query, key and value vectors which are three $\mathbf{d}_{final}$ encoded by individual linear projections. d_k refers to the dimension of $\mathbf{d}_{final}$.

Two parallel graph heads receive the $\mathbf{d}_{final}$ and predict two directional adjacency graphs $\mathbf{A} \in \mathbb{R}^{N\times N}$ as well as $\mathbf{B} \in \mathbb{R}^{N\times N}$. These two graphs are added up to export the final adjacency graph $\mathbf{A}_{final} \in \mathbb{R}^{N\times N}$, which indicates the directional connections of nodes. The predicted adjacency graph is supervised by a binary cross-entropy loss :

$$\mathcal{L}_{\mathbf{graph}} = -(\bar{\mathbf{p}}\log(\mathbf{p}) + (\mathbf{1}-\bar{\mathbf{p}})\log(\mathbf{1}-\mathbf{p})), \quad (3)$$

where $\mathbf{p}$ represents the predicted adjacency graph and $\bar{\mathbf{p}}$ is the adjacency graph label.

4. Implementation Details

Network architecture. TopDiG adopts ResNet50 [11] as the CNN encoder. The dimension of feature map $\mathbf{F}^{D\times H\times W}$ and visual descriptors $\mathbf{d}$ is set as D = 64 while that of initial descriptors $\mathbf{d}_{init}$ and final descriptors $\mathbf{d}_{final}$ is D' = 768. The MLP in DiG generator module consists of two fully-connected layers. In terms of the connection network, we sequentially stack M = 2 transformer encoder layers which contain h = 12 parallel heads. When tackling the polygon-shape targets, we apply Sinkhorn [7] algorithm to optimize the final adjacency graph $\mathbf{A}_{final}$ as suggested in [23, 41].

Training. TopDiG is trained in an end-to-end manner by adding up the losses of TCND and DiG generator: $\mathcal{L}_{\mathbf{total}} = \mathcal{L}_{\mathbf{node}} + \mathcal{L}_{\mathbf{graph}}$. Instead of training from the scratch, we suggest first pretraining TCND with separate $\mathcal{L}_{\mathbf{node}}$ and then training the full model when the TCND detects sufficient nodes. The parameter N is set to 320 and Φ is set as 2 (see Section 5.2). In addition, augmentations *w.r.t* rotation, flipping, Gaussian blur and changes in HSV (Hue, Saturation and Value) space are randomly applied to training images. Other hyperparameters for each specific dataset can be found in Section 5.3. We train TopDiG with the automatic mixed precision (AMP) strategy provided by PyTorch framework. The computing platform is powered by Ubuntu 18.04 and equipped with NVIDIA Tesla V100 GPU as well as Intel Xeon Gold 5218 CPU @ 2.3GHz.

Inference. We recover the topological directional graph by connecting the nodes from predicted adjacency graph $\mathbf{A}_{final}$. The connections of nodes that are represented in the diagonal of $\mathbf{A}_{final}$ are discarded.

5. Experiments

5.1. Datasets and metrics

Datasets. Buildings, water bodies and roads are chosen as representative targets in remote sensing images. We evaluate TopDiG on five datasets, namely Inria Aerial Image Labeling dataset (**Inria**), CrowdAI Mapping Challenge dataset (**CrowdAI**), five-classes Gaofen Image Dataset (**GID**), a Gaofen-2 satellite water bodies dataset (**GF2**) and Massachusetts Roads Dataset (**Massachusetts**).

- **Inria** [18] dataset is designed for building extraction and provides 170 / 10 aerial images for `train` / `val`. Their size is 5000×5000 pixels and the spatial resolution is $0.3m$. We crop raw images to the size of 300×300 pixels and remove tiles without buildings.
- **CrowdAI** [20] is an urban landscapes dataset that consists of 280741 and 60317 images for `train` and `val`. The size of all images is 300×300 pixels. In this work, we validate models on its small `val` set that contains 1820 images.
- **GID** [27] dataset is designed for multi-class semantic segmentation tasks and consists of 150 images with the spatial resolution of $4m$. We crop raw images to 31500 tiles with the size of 512×512 pixels. These images are randomly split to 25500, 6000 for `train` and `val`, respectively.

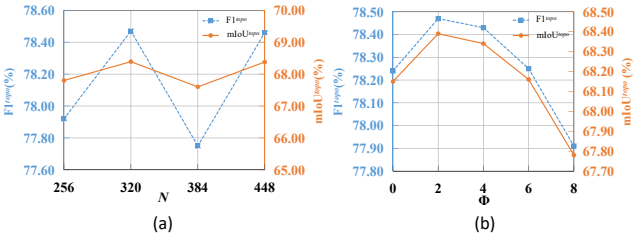


Figure 5. **Ablation studies on the effects of N and Φ.** (a) $\mathbf{F1}^{topo}$ and $\mathbf{mIoU}^{topo}$ of different detected nodes number N; (b) $\mathbf{F1}^{topo}$ and $\mathbf{mIoU}^{topo}$ of tolerance distance Φ between two detected nodes. Notably, $\mathbf{F1}^{topo}$ and $\mathbf{mIoU}^{topo}$ achieve the highest scores when N and Φ are set to 320 and 2, respectively.

- **GF2** [37] dataset serves the semantic segmentation of water bodies and is composed of 13787 images for training and 5949 images for validation. The size of all `train` and `val` images is 512×512 pixels with the spatial resolution is $0.5m$.
- **Massachusetts** [19] is a publicly accessible roads dataset which is composed of 1108 and 14 aerial images for `train` and `val`. The raw images are 1500×1500 pixels and have the resolution of $1m$. We crop all images to the size of 300×300 pixels following the `train` and `val` partition and filter out samples without roads.

Metrics. To evaluate the performances of extracted buildings and water bodies, standard pixel-wise metrics, namely pixel accuracy ($\mathbf{PA}^{mask}$), F1 score ($\mathbf{F1}^{mask}$) and mean intersection over union ($\mathbf{mIoU}^{mask}$) are measured using ground truth segmentation labels and predicted topological directional graphs. For buildings and water bodies, those masks are the interiors of their contours. Roads predictions of **Massachusetts** are not evaluated by the aforementioned metrics since our method directly obtains road centerlines.

The quality of topological directional graphs is evaluated for buildings, water bodies and roads by comparing the predicted graphs to ground truth topological graphs. We evaluate the topological quality by dilating δ pixels (see Section 5.2) around boundary or centerlines for polygon-shape and line-shape targets, respectively. We employ topology-wise metrics of $\mathbf{PA}^{topo}$, $\mathbf{F1}^{topo}$, $\mathbf{mIoU}^{topo}$ and average path length similarity (**APLS**).

5.2. Diagnostic Experiment

To quantitatively analyse and verify the design of each module in the TopDiG, ablation studies are conducted on the **Inria** dataset.

Effects of N and Φ. We investigate influences of parameters N and Φ (Section 3.1) with $\mathbf{F1}^{topo}$ and $\mathbf{mIoU}^{topo}$. $\mathbf{F1}^{topo}$ balances the precision and recall to examine the quality of boundary topology. $\mathbf{mIoU}^{topo}$ is a metric that evaluates the correctness of predicted boundaries within a buffer area. Quantitative comparisons are illustrated in Figure 5. **First**, increasing of N is not prone to boost the accuracy of predictions (Figure 5(a)). When N is set as 320, $\mathbf{F1}^{topo}$ and $\mathbf{mIoU}^{topo}$ reach the peak with 78.47 % and 68.39 %, respectively. Otherwise leading to decreases in all metrics. **Second**, enlarging Φ from 0 to 2 gains improvement in the predicted topological directional graphs (Figure 5(b)). Both $\mathbf{F1}^{topo}$ and $\mathbf{mIoU}^{topo}$ increase from 78.24 %, 68.15 % to 78.47 %, 68.39 %, respectively. This demonstrates that node density controlled by Φ = 2 is reasonable to recover topological graphs from detected nodes.

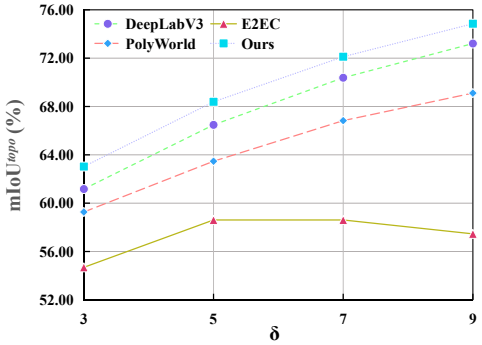


Figure 6. $\mathbf{mIoU}^{topo}$ **with respect to different** δ**.** For fair comparison, we adopt the dilating factor $\delta = 5$ for all experiments.

Method	Pixel-wise Metrics			Topology-wise Metrics			
	$PA^{mask}\uparrow$	$F1^{mask}\uparrow$	$mIoU^{mask}\uparrow$	$PA^{topo}\uparrow$	$F1^{topo}\uparrow$	$mIoU^{topo}\uparrow$	APLS↑
FPN	94.26	90.70	83.56	93.59	77.43	67.26	40.03
TCND w/o	81.75	44.98	40.88	87.48	46.66	43.74	45.79
TCND w/	94.70	91.32	84.56	93.88	78.47	68.39	48.09

Table 1. **Analysis of different node detectors** on **Inria** dataset. The three rows show scores of TopDiG when adopting FPN, TCND w/o or w/ weighting block.

Effects of different node detectors. We also evaluate the design of TCND (Section 3.1) by comparing with FPN. As shown in Table 1, TCND outperforms FPN on all metrics. By concentrating on compact geometric textures via the meticulous perception on topological components, it leads to 1 %, 1.13 % and 8.06% increases in $\mathbf{mIoU}^{mask}$, $\mathbf{mIoU}^{topo}$ and **APLS**, respectively. Moreover, we observe that topology-wise metrics $\mathbf{mIoU}^{topo}$ gains more improvement than pixel-wise $\mathbf{mIoU}^{mask}$. This indicates that TCND facilitates the preservation of topological components and reduces incorrect connections among extracted nodes.

Impact of weighting block. We further estimate the effects of weighting block designed in TCND (Section 3.1). Table 1 shows that the weighting block is crucial for balance of semantic contexts. Excluding weighting block significantly deteriorates the performance of TopDiG. Consequently, the $\mathbf{mIoU}^{mask}$ and $\mathbf{mIoU}^{topo}$ scores decline from 84.56% and 68.39% to 40.88% and 43.74%, respectively. This demonstrates that semantic attention provided by the weighting block is necessary to assist the compact percep-

Type	Method	Memory & Speed			Accuracy	
		#Params (Mb)	MACs (Gb)	FPS	mIoU^{mask}	mIoU^{topo}
2 Layers	+1 Heads	41.04	144.75	11.49	39.70	45.42
	+4 Heads			12.20	40.10	45.81
	+8 Heads			12.98	40.84	46.40
12 Heads	+2 Layers	41.04	144.75	14.49	40.66	46.81
	+4 Layers	55.21	149.28	12.35	38.93	45.36
	+6 Layers	69.39	153.82	11.76	39.76	45.25

Table 2. **Comparisons of different M and h on Inria**. Obviously, TopDiG achieves the best performance and efficiency when M and h are set as 2 and 12, respectively.

tion of target topology.

Impact of dilating factor. We exterminate the dilating factor δ to search the most reasonable dilating pixels for topological evaluation. As shown in Figure 6, DeepLabV3, PolyWorld and TopDiG depict the same tendency that $\mathbf{mIoU}^{topo}$ progressively increases with δ. By contrast, score of E2EC reaches the inflection point at $\delta = 5$ and tends to drop afterwards. For fair comparison, we uniformly set dilating factor δ to 5 for all experiments.

Influences to memory consumption and speed. We evaluate the total number of parameters, multiply–accumulate operations (MAC) and frame per second (FPS) with respect to M and h. For efficiently evaluating these metrics, models are assessed on randomly selected 100 / 40 images for `train` / `val`. Table 2 indicates that fewer transformer encoder layers in DiG generator decreases steadily in parameters and MAC, while more heads bring faster inference speed and higher $\mathbf{mIoU}^{topo}$. Consequently, we choose $M = 2$ and $h = 12$ as the default value.

5.3. Comparison with state-of-the-art methods

Extensive experiments are conducted to compare TopDiG with segmentation-based, contour-based and graph generation methods. Quantitative comparisons on **Inria**, **CrowdAI**, **GID**, **GF2** and **Massachusetts** datasets are reported in Table 3. TopDiG achieves the best performance in most cases and displays the reliability on both polygon-shape and line-shape targets. Evaluations on different datasets are as follows:

Inria. When training TopDiG on the **Inria** dataset, we set the learning rate as 1e-4 and adopt an Adam optimizer without learning rate scheduler. Early stop strategy is utilized as long as no improvements in $\mathcal{L}_{\mathbf{node}}$, $\mathcal{L}_{\mathbf{graph}}$, $\mathbf{mIoU}^{mask}$ and $\mathbf{mIoU}^{topo}$. As shown in Table 3, TopDiG outperforms all other models on topological quality metrics by at least 1.91% $\mathbf{mIoU}^{topo}$ and 7.61% **APLS**. In addition, it also reports competitive scores on pixel-wise metrics such as the 84.56% $\mathbf{mIoU}^{mask}$. These assessments imply that TopDiG can precisely extract topology of polygon-shape targets from aerial images. Figure 7 illustrates some visualized predictions of TopDiG.

CrowdAI. On this dataset, early stop strategy is also adopted and all other settings were the same as the **Inria** dataset. TopDiG is solidly superior over all three competitors on the correctness of predicted topological graphs correctness with highest 59.61% **APLS**. Moreover, compared with DeepLabV3, TopDiG also gains completive performance on pixel-wise metrics with 90.23% $\mathbf{mIoU}^{mask}$. Though DeepLabV3 reports decent scores on segmentation, it unavoidably suffers from jagged contours and requires post-processing for polygon simplification. By contrast, TopDiG can directly extract compact topological directional graphs from images and perverse their completeness.

GID. Experiment settings and strategy on **Inria** are also adopted on **GID**. TopDiG surpasses E2EC and PolyWorld by 24.26% and 8.28% $\mathbf{mIoU}^{mask}$, respectively. Furthermore, it outperforms all three other methods in terms of topology metrics by at least 3.96% $\mathbf{mIoU}^{topo}$ and 5.10% **APLS**. Figure 7 visually presents a few of examples for the water bodies, which clearly demonstrate the topological preservation ability of TopDiG in delineating details of polygon-shape targets.

GF2. We set the same hyperparameters as those in **Inria**. Images in the **GF2** dataset have higher spatial resolution and are better annotated than **GID** dataset. Obviously, TopDiG presents superior scores over other approaches on extracting topological directional graphs. According to quantitative evaluation for **GF2**, TopDiG suppresses other methods by at least 0.16% $\mathbf{mIoU}^{mask}$. In terms of topology quality, TopDiG notably outperforms DeepLabV3, E2EC and PolyWorld by approximately 4.58%, 7.45%, 8.08% $\mathbf{mIoU}^{topo}$ and at least 3.41% **APLS**.

Massachusetts. We adopt AdamW [17] optimizer and set initial learning rate as 2e-3. We also utilize the early stop strategy and all other settings are the same as **Inria**. Table 3 reveals the performance of TopDiG with 70.66% $\mathbf{mIoU}^{topo}$, which is better than that of DeepLabV3 and PolyWorld. TopDiG further achieves much better **APLS** than other methods with the score of 64.60%. This indicates that TopDiG can tackle line-shape tagetes and construct precise topological graphs. Figure 7 shows a few of representative examples and illustrates the ability of TopDiG on both polygon-shape and line-shape targets. This demonstrates that TopDiG is class-agnotic and can extract topological directional graphs regardless of categories.

6. Conclusion

In this work, we introduce a class-agnostic framework called TopDiG to extract topological directional graphs from remote sensing images. TopDiG formulates both polygon-shape and line-shape targets as directional graphs and can tackle targets regardless of their classes. We design TCND to perceive compact topological components and extract nodes with the appropriate density. In addition, the DGS strategy is proposed to dynamically generate adjacency graph labels in an on-the-fly manner and stimulates the prediction of nodes connections. Finally, the DiG genera-

Figure 7. **Visual examples of TopDiG** on the **Inria**, **GID** and **Massachusetts** datasets.

Dataset	Method	Pixel-wise Metrics			Topology-wise Metrics			
		PA^{mask} ↑	$F1^{mask}$ ↑	$mIoU^{mask}$ ↑	PA^{topo} ↑	$F1^{topo}$ ↑	$mIoU^{topo}$ ↑	APLS ↑
Inria [20]	DeepLabV3	94.94	91.93	85.45	93.20	76.77	66.48	40.48
	E2EC	88.46	70.85	63.64	92.69	65.83	58.61	39.46
	PolyWorld	90.82	83.54	73.41	92.92	73.60	63.47	40.03
	Ours	94.70	91.32	84.56	93.88	78.47	68.39	48.09
CrowdAI [20]	DeepLabV3	97.08	95.74	91.92	94.82	83.49	74.06	48.07
	E2EC	95.62	92.11	86.72	93.70	78.67	69.13	36.05
	PolyWorld	93.67	90.29	82.89	93.21	77.71	67.43	51.73
	Ours	96.45	94.77	90.23	94.51	82.20	72.51	59.61
GID [27]	DeepLabV3	99.05	97.34	94.92	99.23	79.27	70.55	80.14
	E2EC	98.85	76.03	69.68	99.16	73.90	66.32	76.17
	PolyWorld	98.07	90.05	85.66	99.17	73.65	66.04	75.84
	Ours	99.17	96.52	93.94	99.39	82.56	74.51	85.24
GF2 [37]	DeepLabV3	99.24	98.14	96.40	99.11	79.46	70.70	77.90
	E2EC	99.25	81.29	75.37	99.01	75.96	67.83	74.34
	PolyWorld	98.55	94.68	91.10	98.98	75.51	67.20	73.63
	Ours	99.40	98.20	96.56	99.28	83.57	75.28	81.31
Massachusetts [19]	DeepLabV3	-	-	-	95.75	77.94	68.30	51.40
	PolyWorld	-	-	-	94.28	76.56	66.59	47.74
	Ours	-	-	-	95.16	80.33	70.66	64.60

Table 3. **Quantitative comparisons for TopDiG and segmentation-based, contours-based, graph generation approaches**. We evaluate the pixel-wise and topology-wise metrics on **Inria**, **CrowdAI**, **GID**, **GF2** and **Massachusetts** datasets. Red and blue represent the top-2 scores. We use ↑ and ↑ to indicate the increases crossing all datasets.

tor is introduced to recover precise topological graphs from nodes. Extensive experiments demonstrate that TopDiG achieves competitive performance on the prediction of topological directional graphs. Our future work will concentrate on multi-class extraction and the lightweight design. We wish this work provides meritorious insights for the vector topology extraction field.

7. Acknowledgment

This work was supported by the Chinese National Natural Science Foundation Projects (No. 41901265), funded by State Key Laboratory of Geo-Information Engineering, NO. SKLGIE2021-M-3-1, a Major Program of the National Natural Science Foundation of China (No. 92038301), and was supported in part by the Special Fund of Hubei Luojia Laboratory (No. 220100028).

References

[1] David Acuna, Huan Ling, Amlan Kar, and Sanja Fidler. Efficient interactive annotation of segmentation datasets with polygon-rnn++. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 859–868, 2018. 1

[2] Md Zahangir Alom, Mahmudul Hasan, Chris Yakopcic, Tarek M Taha, and Vijayan K Asari. Recurrent residual convolutional neural network based on u-net (r2u-net) for medical image segmentation. *arXiv preprint arXiv:1802.06955*, 2018. 3

[3] Favyen Bastani, Songtao He, Sofiane Abbar, Mohammad Alizadeh, Hari Balakrishnan, Sanjay Chawla, Sam Madden, and David DeWitt. Roadtracer: Automatic extraction of road networks from aerial images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4720–4728, 2018. 1, 2, 3

[4] Anil Batra, Suriya Singh, Guan Pang, Saikat Basu, CV Jawahar, and Manohar Paluri. Improved road connectivity by joint learning of orientation and segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10385–10393, 2019. 1

[5] Davide Belli and Thomas Kipf. Image-conditioned graph generation for road network extraction. *arXiv preprint arXiv:1910.14388*, 2019. 2

[6] Lluis Castrejon, Kaustav Kundu, Raquel Urtasun, and Sanja Fidler. Annotating object instances with a polygon-rnn. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5230–5238, 2017. 1

[7] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013. 3, 5

[8] David H Douglas and Thomas K Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: the international journal for geographic information and geovisualization*, 10(2):112–122, 1973. 2

[9] Nicolas Girard, Dmitriy Smirnov, Justin Solomon, and Yuliya Tarabalka. Polygonal building extraction by frame field learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5891–5900, 2021. 1, 2

[10] Ali Hatamizadeh, Debleena Sengupta, and Demetri Terzopoulos. End-to-end trainable deep active contour models for automated image segmentation: Delineating buildings in aerial imagery. In *European Conference on Computer Vision*, pages 730–746. Springer, 2020. 2

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[12] Yuan Hu, Yunpeng Chen, Xiang Li, and Jiashi Feng. Dynamic feature fusion for semantic edge detection. *arXiv preprint arXiv:1902.09104*, 2019. 3

[13] Muxingzi Li, Florent Lafarge, and Renaud Marlet. Approximating shapes in images with low-complexity polygons. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8633–8641, 2020. 1, 2

[14] Zuoyue Li, Jan Dirk Wegner, and Aurélien Lucchi. Topological map extraction from overhead images. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1715–1724, 2019. 1

[15] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 3

[16] Huan Ling, Jun Gao, Amlan Kar, Wenzheng Chen, and Sanja Fidler. Fast interactive object annotation with curve-gcn. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5257–5266, 2019. 1, 3

[17] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 7

[18] Emmanuel Maggiori, Yuliya Tarabalka, Guillaume Charpiat, and Pierre Alliez. Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark. In *IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, 2017. 5

[19] Volodymyr Mnih. *Machine learning for aerial image labeling*. University of Toronto (Canada), 2013. 6, 8

[20] SP Mohanty. Crowdai mapping challenge 2018 dataset, 2019. 5, 8

[21] Thomas Panagopoulos and Maria Dulce Carlos Antunes. Integrating geostatistics and gis for assessment of erosion risk on low density quercus suber woodlands of south portugal. *Arid Land Research and Management*, 22(2):159–177, 2008. 1

[22] Sida Peng, Wen Jiang, Huaijin Pi, Xiuli Li, Hujun Bao, and Xiaowei Zhou. Deep snake for real-time instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8533–8542, 2020. 1, 3, 4

[23] Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, and Andrew Rabinovich. Superglue: Learning feature matching with graph neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4938–4947, 2020. 3, 5

[24] Richard Sinkhorn and Paul Knopp. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343–348, 1967. 3

[25] Andrew K Skidmore, Wietske Bijker, Karin Schmidt, and Lalit Kumar. Use of remote sensing and gis for sustainable land management. *ITC journal*, 3(4):302–315, 1997. 1

[26] Yong-Qiang Tan, Shang-Hua Gao, Xuan-Yi Li, Ming-Ming Cheng, and Bo Ren. Vecroad: Point-based iterative graph exploration for road graphs extraction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8910–8918, 2020. 1, 2, 3

[27] Xin-Yi Tong, Gui-Song Xia, Qikai Lu, Huanfeng Shen, Shengyang Li, Shucheng You, and Liangpei Zhang. Land-cover classification with high-resolution remote sensing images using transferable deep models. *Remote Sensing of Environment*, 237:111322, 2020. 5, 8

[28] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 3

[29] Shiqing Wei and Shunping Ji. Graph convolutional networks for the automated production of building vector maps from aerial images. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–11, 2021. 1, 3

[30] Shiqing Wei, Shunping Ji, and Meng Lu. Toward automatic building footprint delineation from aerial images using cnn and regularization. *IEEE Transactions on Geoscience and Remote Sensing*, 58(3):2178–2189, 2019. 1, 2

[31] Zhenhua Xu, Yuxuan Liu, Lu Gan, Xiangcheng Hu, Yuxiang Sun, Ming Liu, and Lujia Wang. csboundary: City-scale road-boundary detection in aerial images for high-definition maps. *IEEE Robotics and Automation Letters*, 7(2):5063–5070, 2022. 1, 2, 3, 4, 5

[32] Zhenhua Xu, Yuxuan Liu, Lu Gan, Yuxiang Sun, Xinyu Wu, Ming Liu, and Lujia Wang. Rngdet: Road network graph detection by transformer in aerial images. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–12, 2022. 1, 2, 3, 5

[33] Zhenhua Xu, Yuxiang Sun, and Ming Liu. icurb: Imitation learning-based detection of road curbs using aerial images for autonomous driving. *IEEE Robotics and Automation Letters*, 6(2):1097–1104, 2021. 1, 2

[34] Zhenhua Xu, Yuxiang Sun, and Ming Liu. Topo-boundary: A benchmark dataset on topological road-boundary detection using aerial images for autonomous driving. *IEEE Robotics and Automation Letters*, 6(4):7248–7255, 2021. 1, 2

[35] Tao Zhang, Shiqing Wei, and Shunping Ji. E2ec: An end-to-end contour-based method for high-quality high-speed instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4443–4452, 2022. 1, 3, 4

[36] Ziheng Zhang, Zhengxin Li, Ning Bi, Jia Zheng, Jinlei Wang, Kun Huang, Weixin Luo, Yanyu Xu, and Shenghua Gao. Ppgnet: Learning point-pair graph for line segment detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7105–7114, 2019. 2, 3

[37] Zhili Zhang, Meng Lu, Shunping Ji, Huafen Yu, and Chenhui Nie. Rich cnn features for water-body segmentation from very high resolution aerial and satellite imagery. *Remote Sensing*, 13(10):1912, 2021. 6, 8

[38] Mingting Zhou, Haigang Sui, Shanxiong Chen, Jindi Wang, and Xu Chen. Bt-roadnet: A boundary and topologically-aware neural network for road extraction from high-resolution remote sensing imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 168:288–306, 2020. 2

[39] Chenming Zhu, Xuanye Zhang, Yanran Li, Liangdong Qiu, Kai Han, and Xiaoguang Han. Sharpcontour: A contour-based boundary refinement approach for efficient and accurate instance segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4392–4401, 2022. 1, 3

[40] Yunhui Zhu, Buliao Huang, Jian Gao, Enxing Huang, and Huanhuan Chen. Adaptive polygon generation algorithm for automatic building extraction. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–14, 2021. 3

[41] Stefano Zorzi, Shabab Bazrafkan, Stefan Habenschuss, and Friedrich Fraundorfer. Polyworld: Polygonal building extraction with graph neural networks in satellite images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1848–1857, 2022. 1, 2, 3, 4, 5