Direction-guided Segmentation and Vectorisation of curbstones from high-resolution ortho-images

Mariya Jose[1] Stefan Auer[2] Jiaojiao Tian[2]
[1]Institute for Photogrammetry and Geoinformation, Leibniz University, Hannover, Germany
mariya.jose@stud.uni-hannover.de
[2]Remote Sensing Technology Institute, German Aerospace Center, Wessling, Germany
{stefan.auer, jiaojiao.tian}@dlr.de

Abstract

Generating road networks manually has always been an ineffective and labour-intensive task. Accurate representation of road networks is crucial for various applications, including urban planning, infrastructure management, navigation systems, and especially autonomous vehicle development. In the realm of autonomous driving, the accurate detection of curbstones holds particular significance, as they serve as critical boundaries for vehicle navigation and safety. In this paper, we address the problem of curbstone detection as an iterative graph generation task, wherein curbstone edges are detected vertex by vertex from initial curbstone candidates identified through segmentation. Leveraging techniques from imitation learning, we take a high-resolution ortho-image as input and output a graph representing the detected curbstones. We introduce a direction-guided approach with a novel loss function, termed Slope Penalty loss, aimed at refining the model training process by addressing the slight variations in gradients of the predicted vertices. Our experimental evaluations underscore the effectiveness of these enhancements, as demonstrated through comparisons with the already existing curbstone detection algorithms. The proposed approach is tested over the city area of Munich, Bavaria, Germany.

1. Introduction

The development of autonomous driving technology has increasingly emphasized the critical role of automated detection of curbstones to ensure the safe and efficient movement of vehicles, especially in urban traffic environments where structured road networks are common. The detection of curbs remains a key factor in the ability of autonomous vehicles to distinguish between driving zones on roads, thereby significantly improving both road safety and mobility [2]. Traditionally, curbstone detection has relied on sensors such as LiDARs and cameras mounted on vehicles [25] to acquire images needed for detection. However, data obtained from these methods face major challenges, like differences in lighting conditions, obstacles from vehicles, and pedestrians, environmental factors such as vegetation [2], and the presence of shadows and sparse point cloud data, which can degrade detection accuracy, in particular areas away from the vehicles, which undermines the accuracy and reliability of the detection. There are also hardware requirements for this data acquisition essentially contributing to the overall cost of the system, and establishing new barriers to widespread adaptation and implementation [25].

To successfully address these problems and complement terrestrial mapping efforts, there has been a growing interest in using remote sensing data to map distinguishing elements such as curbstones over large areas. By integrating remote sensing information into maps used in automotive applications, such as Opendrive, there is a chance to boost the accuracy of map data, hence significantly enhancing the performance of autonomous driving systems [25]. One especially intriguing approach to curbstone detection is the use of high-resolution aerial photos.

Curbstone detection from high-resolution ortho-images has been investigated primarily based on two principal methodologies: segmentation followed by heuristic post-processing and end-to-end iterative graph generation. The former comprises segmenting the curbstones from the ortho-images and further heuristic post-processing to refine the results. While this method typically provides excellent pixel-level results, it grapples with issues of poor topology correctness and the existence of noise in segmentation results [2], particularly in complex topological scenarios. In contrast, the iterative graph generation method describes the problem as an iterative process in which curbstones are predicted vertex by vertex from initial candidates found through segmentation. This technique presents the ability for stronger topology correctness and decreased noise in

the final results, thereby constituting an appealing venue for curbstone detection in diverse and complicated environments [2], [24], [25]. In short, the combination of aerial imagery with advanced deep-learning strategies suggests extraordinary promise for curbstone detection and road network extraction in urban environments.

This paper builds upon [25] by adapting the iCurb framework for curbstone vectorization in high-resolution ortho-images, tailored to Bavaria, Germany. Key contributions include the exploration of a novel Slope Penalty Loss to improve training by addressing gradient variations in generated edges and implementing graph growth restrictions to confine expansion within segmentation boundaries. These enhancements reduce spurious connections, ensuring greater accuracy and topological correctness in the final vectorized output.

Section 2 provides a review of existing literature on segmentation and deep learning methodologies. The framework for segmentation and vectorization is detailed in Section 3. Section 4 describes the experimental setup, including the dataset and training procedure. The results along with their evaluation using relevant metrics are presented in Section 5, demonstrating the framework's effectiveness. Finally, Section 6 concludes with a summary of the findings, contributions, and potential directions for future research.

2. Related Work

2.1. Segmentation based approaches

Segmentation-based approaches that have been used to detect linear structures like curbstones or roads mainly have two stages: predict the segmentation map and then process this segmentation map to derive vectorized outputs. Since both roads and curbstones are linear structures and are affected by complex topology, the methods for road detection can also be adapted to detect curbstones.

Mnih et al. [15] is credited with pioneering the utilization of neural networks for the segmentation of road networks in aerial images. Subsequent works in segmentation follow a comparable methodology employing more advanced segmentation networks like UNet [19], FPN [12], etc. Usually, these networks' outputs map in raster format, so further post-processing is required to vectorise these maps, filter out outlier detections and correct discontinuities in the connections of the network. To extract especially the lines in digital images, a mathematical method is presented by Hough [9]. Alternatively, the centreline of the segmentation masks could be extracted as in Zhang et al. [26] and then the number of points reduced using the method of Douglas et al. [6], which is combined and used to detect curbstones from ortho-images in Tian et al. [22].

As semantic segmentation solely operates on pixel-level predictions, it often overlooks topology information. Consequently, road network graphs derived from segmentation-based methods exhibit inadequate topology accuracy. Furthermore, heuristic post-processing techniques struggle to rectify errors within the resulting road network graph.

2.2. End-to-end approaches

Different from segmentation-based approaches, end-to-end approaches for the generation of curbstone or road network detection directly output the vectorised results. Barzohar et al. [1] are the first to introduce an automatic approach for finding main roads in aerial images, then paper [5], and paper [23] use machine learning-based methods for road detection. DeepRoadMapper [14] is the first deep learning technique to extract road topology from aerial images, then Hinz et al. [7] used multi-view images and Hu et al. [10] leveraged the high-order topological features such as intersections for road network extraction.

In RoadTracer [2] an iterative search process guided by a CNN-based decision function to predict the graph growth is used to generate the road network vertex by vertex. Belli et al. [3] and Li et al. [11] also demonstrate the potential of graph-based methods in road detection. Similarly, Vecroad [21], employs point-based iterative graph refinement based on local features and connectivity constraints. DAGMapper [8] incorporates learning-based approaches into graph-based road detection methodologies. Finally, the approach iCurb, introduced by Xu et al. [25], utilizes imitation learning for iterative graph generation. This method predicts points sequentially, similar to visual navigation, from a set of candidate points generated by a segmentation model like FPN.

In conclusion, end-to-end graph-based methods have significantly advanced the field of detection from aerial imagery, offering robustness, accuracy, and scalability in delineating complex road networks.

3. Methodology

3.1. General Overview

iCurb encompasses two main tasks: the extraction of initial vertex candidates and the subsequent generation of graphs from these points [25].

Initially, [25] uses a deep learning model like UNet to segment the RGB ortho-image I, to generate the binary segmentation mask, S (also a feature map F for the iterative graph generation). This choice is motivated by UNet's superior performance in segmenting linear structures like curbstones compared to other architectures such as FPN, which is used in iCurb [25]. Then to extract potential candidates for initial vertices, we apply thresholding and skeletonisation [27] to S and then extract the first point of each line element, which yields a list of potential initial vertex candidates, Q.

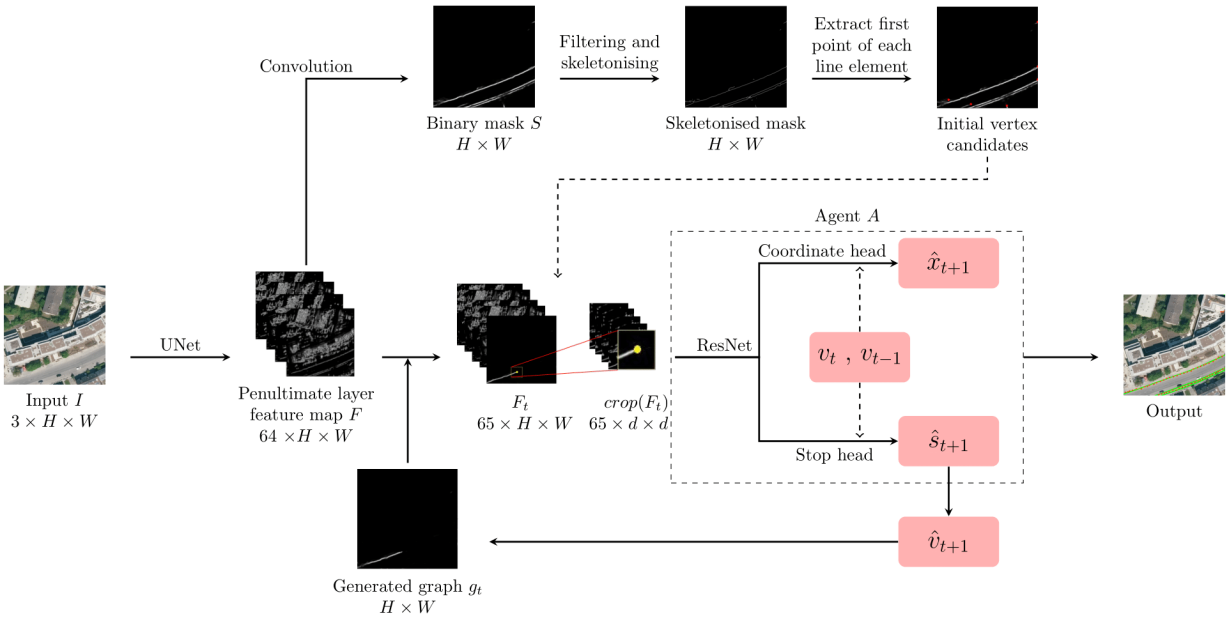


Figure 1. The overview diagram of the iCurb pipeline. A feature extraction backbone like UNet extracts the feature map F which is further processed to yield initial vertex candidates Q, denoted by the red nodes. Q is used only at the beginning of each iteration to grow a graph instance as indicated by the dashed line. The binary map, g_t, tracks the predicted vertices up to time t. g_t is concatenated with F into F_t. The current vertex, v_t (denoted by the yellow node) and a $d \times d$ local feature crop from F_t, $crop(F_t)$ to predict the next vertex $\hat{v}_{t+1}$ by the agent, A. Then g_t is updated. This iterative process continues until all vertices in Q have been processed. The final output consists of predicted vertices denoted by red nodes and edges denoted by green lines. [25].

Subsequently, utilising an imitation learning algorithm for the iterative graph generation approach, we predict the curbstone graphs $G = V, E$. Starting from an initial vertex from Q, the algorithm predicts subsequent vertices sequentially. Each vertex in the generated vertices list V comprises 2D coordinates along with a variable indicating whether the graph generation process should cease or continue, $\{v_t = x_t, s_t\}$. When the value of s_t equals 1, the algorithm halts further point generation for that instance and proceeds to initiate a new graph generation process using the unused vertices from Q. The neighbouring vertices are connected to generate the edges E, during the graph generation.

The iterative graph generation process utilising an imitation learning algorithm involves three main steps:

1. Extract the feature map F using the UNet from the input I as illustrated in Figure 1. To save the previously predicted vertices, set all the pixels covered by the predicted vertices or edges until time t as 1 in g_t (initially g_t will be initialised as a zero matrix).

2. Concatenate F and g_t into a new multi-channel feature map F_t. Here, F_t can be regarded as the environment in imitation learning. Assuming that the current vertex is $v_t \in Q$, we crop a $d \times d$ square block $crop(F_t)$ with the centre at v_t to represent the local environment of v_t. With $crop(F_t)$, v_t and v_{t-1} as input, the agent A predicts the next vertex $v_{t+1} = \{x_{t+1}, s_{t+1}\}$:

$$v_{t+1} = \underset{v_{t+1} \in \text{crop}(F_t)}{\arg\max}\ \pi[v_{t+1} | \text{crop}(F_t), v_t, v_{t-1}] \quad (1)$$

where π is the policy of agent A, $crop(F_t)$ is the current environment, v_t is the current vertex and v_{t-1} is the previous vertex. x_{t+1} and s_{t+1} are predicted by the coordinate head and stop head respectively. v_t is then updated to g_t.

3. During training, the graph is grown with more complex strategies like restricted exploration and free explorations based on DAgger [20], to generate training samples D to teach the agent A. The agent A is trained on D after all the initial vertex candidates in Q are processed and the policy π is updated based on the equation:

$$\hat{\pi}_{update} = \arg\min_{\pi} \sum^{t} |\pi[\text{crop}(F_t), v_t, v_{t-1}] - v^*_{t+1}| \quad (2)$$

where v^*_{t+1} is the ground truth, which will be selected based on dynamic labelling discussed in [25].

Figure 2. Visual representation of how Slope Penalty loss can be calculated. The ground truths are denoted by cyan lines and predicted edges by orange lines. Yellow nodes represent predicted vertices and blue nodes represent their corresponding ground truth label.

3.2. Slope Penalty loss

To improve the training process for our curbstone detection model and encourage smoother predictions, we propose a direction-guided approach by adding a novel loss function: the Slope Penalty Loss. This loss component explicitly penalizes discrepancies between the slopes of predicted and ground truth curbstone coordinates. As illustrated in the flow diagram 3, the iCurb agent takes generated samples as input and predicts vertices iteratively. These predictions, $\hat{x}_{t+1} = (x_p, y_p)$ are compared with the ground truth labels, $x^*_{t+1} = (x_t, y_t)$. The Slope Penalty Loss, L_{slope} can be calculated as:

$$L_{slope} = \text{Smooth L1 Loss}(\min(\Delta\theta, \pi - \Delta\theta), 0) \quad (3)$$

where $\Delta\theta = \left|\arctan\left(\frac{y_p}{x_p}\right) - \arctan\left(\frac{y_t}{x_t}\right)\right|$ as illustrated in the example shown in Figure 2. [1]

Integrating the Slope Penalty Loss into the overall training process is achieved by combining it with the existing coordinate loss (L_{l_1}) which is a regression loss calculated using smooth L1 loss [18] between the predicted and true vertices using a weighted sum. The slope penalty encourages smoother transitions between predicted vertices, penalizing abrupt directional changes. The weighting factor, w determines the influence of the slope penalty on the overall loss calculation. By adjusting this coefficient, we can effectively control the impact of the slope deviation on the training dynamics.

$$L = L_{l_1} + w \cdot L_{slope} \quad (4)$$

[1] Please note that this figure does not depict the actual performance of the tracker but serves as an illustrative example to explain the concept.

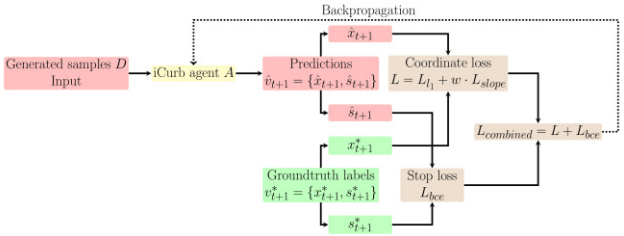


Figure 3. Illustration of the training process for the iCurb agent, integrating the Slope Penalty Loss.

This total loss is backpropagated to refine the agent's predictions iteratively as shown in Figure 3. The slope penalty encourages smoother transitions between predicted vertices, penalizing abrupt directional changes, as depicted by the orange and blue lines in Figure 2.

4. Experimental Setup

4.1. Dataset

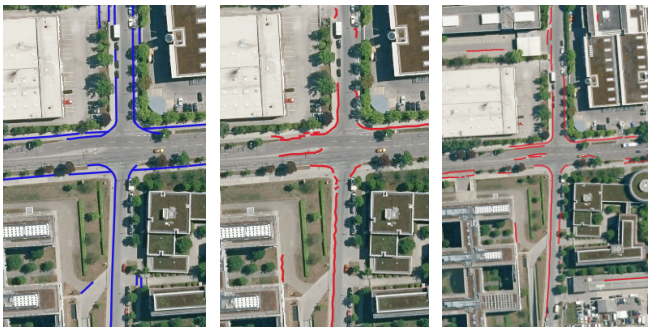
The data used is available from the Agency for Digitisation, High-Speed Internet and Surveying (LDBV), Bavarian Germany. This dataset comprises true orthophotos (TrueDOP-20) with 20cm ground resolution and three channels (RGB = red, green, blue) capturing Munich, Germany. The training data comprised 49, 3500×3500 sized ortho-images, which were split into 512 × 512 sized patches (each image is split into 144 patches with 50% overlap), 6048 RGB patches for training and 150 RGB patches out of 1008 available for validation of the model. 12 ortho-images of sizes 3500 × 3500 pixels are used directly (without splitting) to test the trained model.

For each ortho-image, curbstone vector ground truth data is manually annotated in a vector format as 2D Line Strings with attributes. To facilitate the training and testing, this data is converted into a rasterised form for segmentation and also a dense annotation sequence for the iterative graph generation using imitation learning.

4.2. Training process

The training process consists of two phases. Initially, the focus is on pre-training the segmentation model. Subsequently, train the iCurb pipeline, utilising the pre-trained weights obtained from the segmentation model.

The segmentation model was trained on 512×512 image patches in batches of 8 for 30 epochs using the Adam optimizer (initial learning rate 0.0001). A learning rate scheduler reduced the rate by 50% if the F1-score did not improve for three consecutive epochs. Binary Cross Entropy Loss with a Sigmoid Function [17] was employed. Validation was conducted after each epoch, and the model with the highest F1 score was retained.



(a) Ground truth (b) iCurb (c) iCurb (+ Slope Penalty Loss)

Figure 4. An instance where iCurb trained on Slope Penalty loss performs better than iCurb algorithm. Blue lines indicate the ground truth, and red lines indicate the predictions.

Training of the imitation learning agent followed the sample generation procedure in [20]. Image patches (63×63) were processed in batches of 128. The agent was trained for three epochs with the Adam optimizer (initial learning rate 0.0001), using a scheduler to halve the rate upon stagnation in validation performance. Validation occurred after every 500 images, and the best-performing model was saved. The agent network includes two heads for predicting the coordinate and stopping criteria. The coordinate head employed a Slope Penalty Loss (w = 0.01, which is determined empirically using trial and error. It is observed that a smaller value of w provides better results thus enhancing the model's ability to learn without over-penalizing slope deviations.) as in Equation (4), while the stop head utilized Binary Cross Entropy Loss with Sigmoid Function [17]. These losses collectively optimized model performance during training.

All training procedures were conducted using an NVIDIA GeForce GTX TITAN X GPU.

5. Results and Discussion

For the method comparison, we generated baseline results including two vectorisations of the segmented results using heuristic processes (Hough transform, Douglas Peucker) and iterative graph generation (iCurb and iCurb (+ Slope Penalty loss)).

5.1. Vectorisation with traditional methods

Initially, we explored segmentation-based approaches, we employed the Hough transform leveraging the understanding that most curbs are linear. While effective for identifying straight segments of curbstones, this methodology faltered when faced with complex scenarios such as curves and intersections. Its shortcomings were further emphasised in scenes with overpasses, where it frequently misinterpreted structures, yielding erroneous linear detections.

An alternative method was explored using the Douglas-Peucker algorithm, which improved upon the Hough transform's limitations by better accommodating curves and complex intersections. However, this approach introduced its own challenges, particularly because it failed to accurately constrain the shape of the vectors produced. Numerous spurious vectors were found, reducing the algorithm's practical applicability. The root cause of these inaccuracies lay in the skeletonisation process, which reduced binary masks to single-pixel-width lines [27], inadvertently compromising precision. Despite Douglas-Peucker's capability to enhance recall—achieving a remarkable 42% improvement compared to the Hough transform, its precision varied significantly, culminating in an overall F1 score that offered no significant advantage over the Hough transform.

5.2. Vectorisation with iterative graph generation

The iterative graph generation approach, inspired by iCurb, demonstrates significant improvements. It produces more accurate vectors as evident in Figure 5. Although there is a decline of 24% in recall compared to the previous Douglas Peucker method, the detected vectors exhibit a 10% increase in precision over those produced by Douglas Peucker. Consequently, there is a 12% improvement in the F1 scores.

Incorporating the Slope Penalty Loss further enhances the quality, smoothness and continuity of the vector representations, leading to a substantial 30% increase in precision and a 25% improvement in the F1 score compared to the basic iCurb algorithm as illustrated in an example in Figure 4. While there is a slight decrease in recall, the overall performance is significantly better.

5.3. Limitations

Although strides have been made to improve the results, there remain several scenarios where the algorithm reveals limitations. As shown in Figures 6a and 6b, for instance, errors in graph growth when dealing with overpasses are observed. Here, the algorithm struggles as it perceives the graph instance growing along the overpass and the one growing under it as a single instance. Figures 6b and 6d present another challenge, wherein the high number of false positive detections in the segmentation results significantly impacts the final outcomes, with numerous detections occurring where none exist. Additionally, Figures 6e and 6f depict instances of parallel curbstone instances. While the latter instance showcases promising results possibly due to better segmentation in those areas, the former one exhibits poor outcomes.

6. Conclusion and Outlook

In this paper, we further improve iCurb [25] pipeline by adding a direction guide for detecting and generating graphs

Methods	Precision			Recall			F1-score		
	2.0	5.0	10.0	2.0	5.0	10.0	2.0	5.0	10.0
Hough transform	0.071	0.092	0.127	0.249	0.322	0.438	0.108	0.140	0.193
Douglas Peucker	0.093	0.106	0.112	**0.711**	**0.806**	**0.856**	0.163	0.185	0.196
iCurb	0.148	0.191	0.220	0.421	0.536	0.616	0.212	0.272	0.313
iCurb (+ Slope Penalty loss)	**0.391**	**0.506**	**0.530**	0.468	0.605	0.636	**0.417**	**0.539**	**0.565**

Table 1. Quantitative results for the comparison between segmentation-based and graph-based approaches. The best results are highlighted in bold format. 1.0, 2.0 and 5.0 are the different thresholds in pixels used to calculate the different metrics as mentioned in [25].

(a) Ground truth (b) Hough transform (c) Douglas Peucker (d) iCurb (e) iCurb (+Slope Penalty loss)

Figure 5. Qualitative demonstrations (Blue lines indicate the ground truth, red lines indicate the predictions)

of curbstones from ortho-images. Comparative results with segmentation-based approaches revealed the superiority of the iCurb pipeline, especially when incorporating the Slope Penalty loss during training. The incorporation of the Slope Penalty loss serves as a valuable addition to the training regimen, contributing to the model's ability to learn and

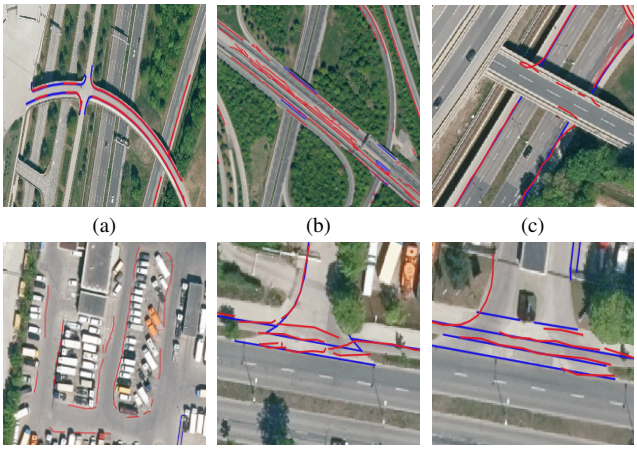


Figure 6. Examples where limitations of the iCurb algorithm can be seen. Blue lines indicate the ground truth, and red lines indicate the iCurb (+ Slope Penalty Loss) predictions.

adapt to variations in curbstone slopes. The iCurb algorithm showed substantial improvements in precision and F1-score compared to traditional methods.

Despite these promising results, several limitations and challenges remain. The algorithm struggles in certain scenarios, such as highly complex intersections, graph growing around overpasses, detection of high false positives in the segmentation results, and the handling of parallel curbstone instances. There are several avenues for future research and development to further improve the performance and applicability of the iCurb pipeline like using a more advanced segmentation model like Dense UNet [4] or transformers [13] or combinations, developing post-processing techniques to refine the generated graphs, such as graph pruning [16] or smoothing algorithms, integrating complementary sensor modalities, such as LiDAR and radar data, to provide additional information or by expanding and diversifying the dataset with images from different cities, seasons, and weather conditions.

By addressing these challenges and exploring these opportunities, the direction-guided iCurb pipeline has the potential to significantly improve the way curbstones are detected and represented from ortho-images. This could lead to advancements in urban planning and navigation systems, ultimately contributing to safer and more efficient urban environments.

References

[1] M. Barzohar and D. B. Cooper. Automatic finding of main roads in aerial images by using geometric-stochastic models and estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7):707–721, (1996). 2

[2] 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, (2018). 1, 2

[3] D. Belli and T. Kipf. Image-conditioned graph generation for road network extraction. In *NeurIPS 2019 workshop on Graph Representation Learning*, (2019). 2

[4] S Cai, Y Tian, H Lui, H Zeng, Y Wu, and G Chen. Dense-unet: a novel multiphoton in vivo cellular image segmentation model based on a convolutional neural network. *Quantitative Imaging in Medicine and Surgery*, 10(6):1275–1285, (2020). 7

[5] 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*, 55(6):3322–3337, (2017). 2

[6] D. H. Douglas and T. 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

[7] S. Hinz and A. Baumgartner. Automatic extraction of urban road networks from multi-view aerial imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 58(1):83–98, (2003). 2

[8] N. Homayounfar, W.-C. Ma, J. Liang, X. Wu, J. Fan, and R. Urtasun. Dagmapper: Learning to map by discovering lane topology. In *Proc. IEEE Int. Conf. Comput. Vis.*, pages 2911–2920, (2019). 2

[9] P. V. C. Hough. *Method and means for recognizing complex patterns*. U.S. Patent 3,069,654, Dec 1962. 2

[10] J. Hu, A. Razdan, J. C. Femiani, M. Cui, and P. Wonka. Road network extraction and intersection detection from aerial images by tracking road footprints. *IEEE Transactions on Geoscience and Remote Sensing*, 45(12):4144–4157, (2007). 2

[11] Z. Li, J. D. Wegner, and A. Lucchi. Topological map extraction from overhead images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1715–1724, (2019). 2

[12] Tsung-Yi Lin, Piotr Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. *CoRR*, abs/1612.03144, (2016). 2

[13] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *CoRR*, abs/2103.14030, (2021). 7

[14] G. Mattyus, W. Luo, and R. Urtasun. Deeproadmapper: Extracting road topology from aerial images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3438–3446, (2017). 2

[15] V. Mnih and G. E. Hinton. Learning to detect roads in high-resolution aerial images. In *European Conference on Computer Vision*, pages 210–223. Springer, (2010). 2

[16] Sean Murray, George D. Konidaris, and Daniel J. Sorin. Roadmap subsampling for changing environments. In *2020*

IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pages 5664–5670, 2020. 7

[17] PyTorch. Bcewithlogitsloss, Accessed on: 11.05.2024. 4, 5

[18] Pytorch. Smoothl1loss, Accessed on: 11.05.2024. 4

[19] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In Nassir Navab, Joachim Hornegger, William M. Wells, and Alejandro F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham, (2015). Springer International Publishing. 2

[20] Stéphane Ross, Geoffrey J Gordon, and J Andrew Bagnell. A reduction of imitation learning and structured prediction to no-regret online learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 627–635, (2011). 3, 5

[21] 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*, pages 8910–8918, (2020). 2

[22] Jiaojiao Tian, Xiangyu Zhuo, Stefan Auer, Franz Kurz, and Peter Reinartz. Fusion of stereo aerial images and official surveying data for mapping curbstones using ai. In *2023 8th International Conference on Signal and Image Processing (ICSIP)*, pages 176–180, (2023). 2

[23] J. D. Wegner, J. A. Montoya-Zegarra, and K. Schindler. Road networks as collections of minimum cost paths. *ISPRS Journal of Photogrammetry and Remote Sensing*, 108:128–137, (2015). 2

[24] 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. 2

[25] 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, Apr. (2021). 1, 2, 3, 5, 6

[26] T. Zhang and C. Y. Suen. A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, 27(3):236–239, (1984). 2

[27] T. Y. Zhang and C. Y. Suen. A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, 27(3), 3 (1984). 2, 5