Roadmap Generation using a Multi-Stage Ensemble of Deep Neural Networks with Smoothing-Based Optimization

Dragoş Costea[1,2,*] Alina Marcu[2,3,*] Emil Sluşanschi[1] Marius Leordeanu[1,2,3]

[1] University Politehnica of Bucharest [2] Autonomous Systems
[3] "Simion Stoilow" Institute of Mathematics of the Romanian Academy
{dragos.costea, alina.marcu}@autonomous.ro, {emil.slusanschi, marius.leordeanu}@cs.pub.ro

Abstract

Road detection from aerial images is a challenging task for humans and machines alike. Occlusion, the lack of visual cues and slim class borders for other road-like structures (such as pathways or private alleys) make the problem inherently ambiguous, requiring logic that goes beyond the input image. We propose a three-stage method for the task of road segmentation - first, an ensemble of multiple U-Net like CNNs generate binary road masks. Second, another CNN learns to refine roads segmentations based on the fusion of the road maps from the first stage. Third, missing links are added based on the inferred graph to improve segmentation.

1. Introduction

Although, remarkable improvements have been made in semantic segmentation of remote sensing imagery, the problem is far from being solved. Thanks to the advances of deep convolutional neural networks ([6], [19], [2]) and large labeled datasets [10], obtaining good object segmentations has come down to training a single CNN.

The U-net architecture [17], based on an encoder-decoder scheme with skip connections, has been extensively used for image segmentation, yielding state-of-the-art results with minimal alterations [12].

We leverage the road segmentation from the initial aerial image and extract road vectors using a smoothing-based optimization algorithm that reasons about missing connections, further improving the road topology.

2. Related work

Road segmentation from aerial image has been extensively studied in the literature ([9], [15], [7]). Given the inherent ambiguity of the problem, there can never be a perfect solution - educated guesses need to be made in order to determine the most probable road layout.

Pixelwise semantic segmentation is generally a first step. Given an RGB or multispectral image, a binary road mask is produced. State-of-the-art methods use multi-stage deep convolutional neural networks ([12], [13]).

Most of the times, we are interested in a map - that is, roads vectors. For this purpose, several methods have been proposed that aim to generate a road graph. Some start with the skeletonized version and add all the pixels as nodes [14], others attempt to simplify the road structure using road junctions, generating a sparse graph [4], resulting in a similar representation as OpenStreetMap [16].

Other methods propose a CNN-based, iterative graph construction method [1]. Starting from a point known to be on the road, it receives the RGB image centered on that point. It decides either to walk a fixed distance at an angle inferred by the CNN or step back to the previous node. Although the authors claim it finds 45% more junctions, it does not have a dedicated junction finder and since it is a patch-based, local algorithm, it has issues with both high curvature and long, straight roads.

3. Proposed Method

We propose a three-stage method for roads extraction (segmentation and vectors). Firstly, we independently train various U-net-like networks on the task of roads segmentation and intersections detection. Next, we combine these partial predictions, along with the RGB input and feed them to another network to produce a new road segmentation map. Road vectors are obtained using our smoothing-based optimization module. In the third stage, we use both road segmentation and road vectors to add missing links (especially around intersections).

Road segmentation. We train various U-net-like architectures for the task of road segmentation. We reduce the spatial resolution of our input using (2, 2) max-pooling operations. After each downsampling layer we double the number of learned features, in the same manner as [18]. We equally reduce the spatial resolution of the input by a factor

*Equal contribution

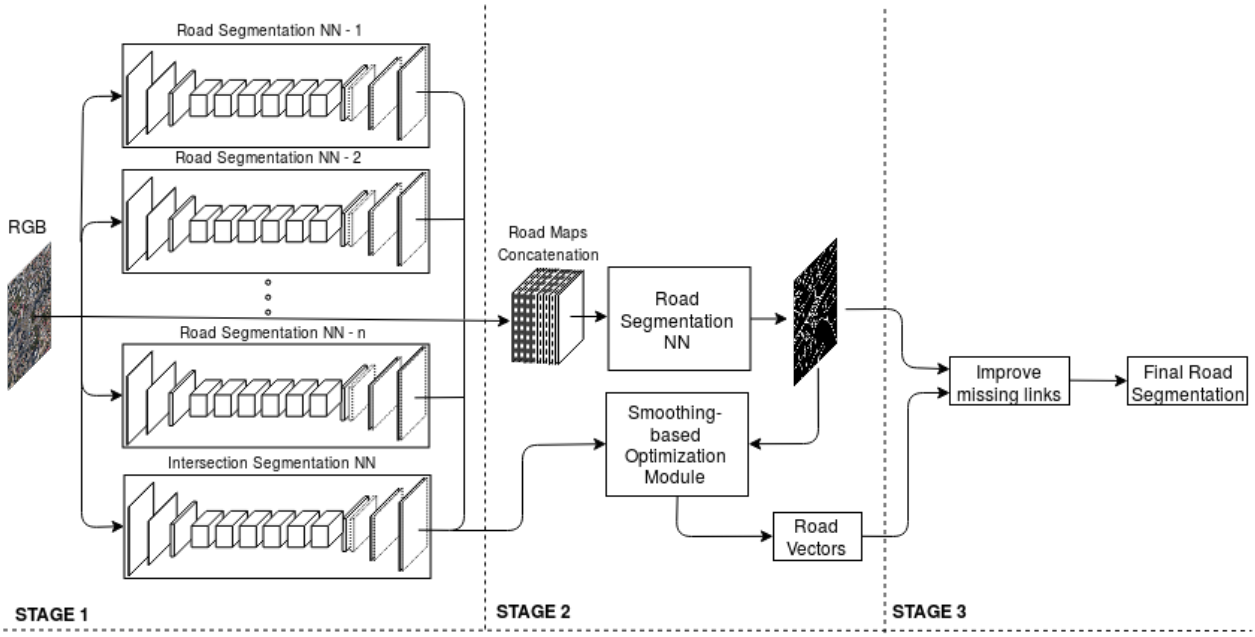

Figure 1. Overview of our method. In the first stage, we train different road segmentation networks. The generated road maps, along with the RGB and detected intersections are concatenated and fed to another U-net-like architecture trained for road segmentation. This improved segmentation map is used for road vectors generation. In the last stage, we add missing links to the segmentation map using the previously generated road vectors.

of 8. In order to capture multi-scale information within the network, we adjust the field-of-view of deeper filters by using chained dilated convolutions with various dilation rates (referred as Atrous Spatial Pyramid Pooling in [3]). We experiment with different dilation rates for each network. The core network has a bottleneck of 6 dilated convolutions with progressively increasing rates (1, 2, 4, 8, 16, 32). Afterwards, the decoder branch is built in the same manner as in U-net. We upscale the feature maps and add skip connections until we reach the size of the input. From now on, we will refer to this model as *Max dilation 32*. We build 2 more variants, adding one dilated convolution with rate of 48 (termed *Max dilation 48*) and another, with two more convolutions with dilation rates of 48 and 64 (*Max dilation 64*). Each convolutional layer is followed by batch normalization [8] and ReLU non-linearity.

Intersection detection. Our experiments (see Figure 2) showed that segmentation masks are weak around road junctions. Therefore, we trained a separate network for intersections detection to correct such mistakes. We have collected the ground truth for training this network from the road ground truth, by using a heuristic method based on skeletonization - branched points having at least three ramifications spawning for at least 150 meters were considered intersections. Having the point locations, we trained a network to place a dot centered on each. Due to the small distance between certain intersections, we have chosen a 10 meters-wide dot as label. This resulted in a small number of overlapping dots, and thus accurate intersection location on the testing set. We trained Max dilation 32 using roads mask as input and also using the RGB image and roads combined. We report the results of intersection detection in Table 3.

3.1. Road refinement with optimization

We propose a two-stage refinement algorithm. First, we generate road vectors using the binary segmentation and an optimization algorithm. Second, we post-process the graph in order to add the missing links.

Faced with occlusion, humans typically take a distance-based guess whether there should be a connection between two road links or not. Since the purpose of a road network is to provide connectivity in order to facilitate land access, it would make little sense to end a secondary road just before connecting to a main road.

In order to improve road connectivity (and overall segmentation performance), we use the binary mask from the CNN as input for our optimization algorithm. Starting from sampled points, it reasons about links by scoring each connection and moving points to find the best fit.

We have chosen smoothing-based optimization [11] for this task. The method can maximize a non-negative function, for which the only requirement is the ability to be evaluated at a given point. Our scenario is simple: given a point or midpoint on a road link, find the best position in order to match the binary mask.

The SBO-based method returns a road graph, given a

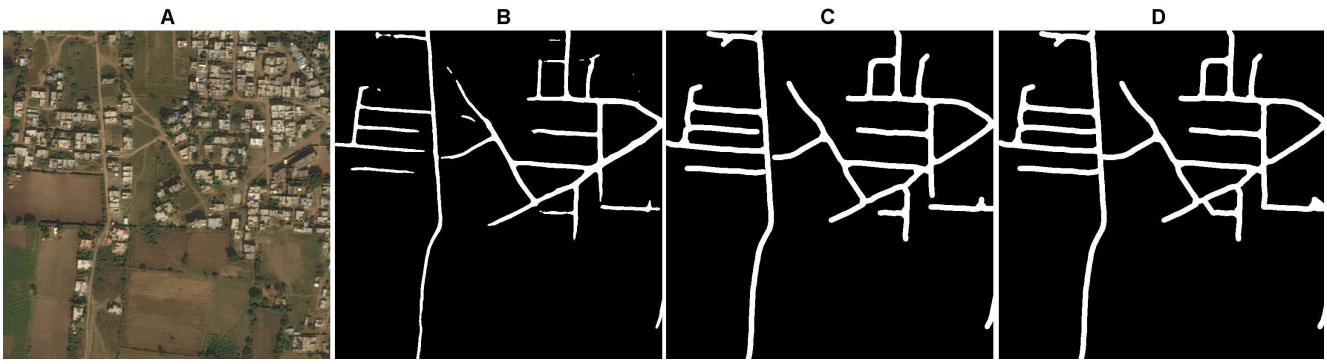

Figure 2. Qualitative results of stage 3 of our method (adding missing links). (A) RGB input, (B) binary mask from CNN, (C) plotted road vectors from SBO, (D) final segmentation with missing links added to the road vectors.

minimum road link length. However, due to the scoring method (IoU) and significant road segment differences, we noticed this slightly reduces overall performance, even though it improves topological accuracy and significantly reduces storage requirements. For the final submission, we decided to keep only the non-overlapping links from the graph.

3.2. Adding missing links

Having generated the graph, several links are still missing, mostly due to occlusion. A module is needed to connect the roads based on distance or texture information. At this stage, we use the graph nodes generated by the SBO algorithm to infer an improved road layout, based on the added links (see Figure 2). This would be a challenging task using classical methods, such as distance transform, since some connections could be easily missed without knowing the graph structure and possible connection points. For example, connecting a road to itself sounds like a bad idea, but it might occur in occluded roundabouts.

4. Experiments

DeepGlobe Dataset [5]. The training set comprises of 6226 images spanning a total of 1632 km^2. A validation set of 1243 images (covering an area of 362 km^2) was provided in the first round of the competition. Another 1101 images were chosen for testing and released in the second round. They cover a total land area of 288 km^2. The images were collected by DigitalGlobe's satellites at a spatial resolution of 50cm. The novelty of this dataset consists in the road labels, of various width, provided for each image in the training set. The task is to detect road pixels present in each satellite image from the validation and test set (without any road labels). The evaluation metric used in this competition is mean Intersection over Union (IoU) [5].

Training in iterations. For the first round of the competition, we used 5603 randomly sampled images ($\approx 90\%$) from the original training set to train our models. The remaining 623 images were used for validation. This is iteration 1 of our training process. We trained both our Max dilation 32 and Max dilation 48 networks. After convergence, we used the predictions of each network on the validation set as labels and trained the networks again. For the second iteration, we keep the same splitting ratio of 90%-10%. In the second iteration, the models were trained on 6722 images and validated on 747. We report the IoU scores of each iteration on the validation set in Table 1.

Table 1. **Round 1**. Roads segmentation results on the official validation set, 1243 images. The results were given by the submission site.

Model	Iteration	IoU Validation
Max dilation 32	1	0.5924
	2	0.5975
Max dilation 48	1	0.6039
	2	**0.6058**

Further on, in our experiments, we used the original ground truth labels mixed with the predictions provided by our Max dilation 32 network, in iteration 1, on the validation set. In Table 2 we report the segmentation results produced by our networks after the second iteration.

One intriguing particularity of this dataset is the variable road width. We tried to assess the impact of thickness on detection performance. Therefore, we trained another Max dilation 32 network, using constant width roads ($\approx$ 4 meters), generated from the skeletonized version of the ground truth. We extended the road width experiments after the submission - see section 5.1 and Table 6.

Building ensembles. For the second iteration, we used the networks trained in stage 1 and combined them using two different approaches. Ensemble 1 is built by summing over the results. For our second ensemble, we train a new Max dilation 32 network, by fusing the RGB satellite image with the outputs of all our networks from Ensemble 1. Different from the previous ensemble is the presence of the intersection map. The first ensemble tends to predict roads

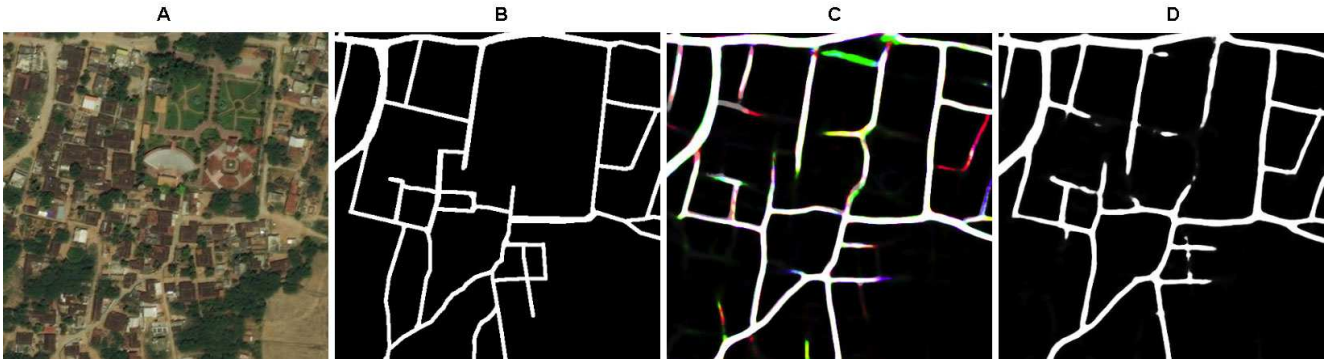

Figure 3. Qualitative results of roads segmentation using our ensembles. (A) RGB input, (B) Ground truth, (C) Ensemble 1, shown as sum of 4 CNN outputs: Max dilation 32 - blue, Max dilation 48 - green, Max dilation 64 - red, Max dilation 32, same width thin - grey. (D) Ensemble 2, the output of Stage 2. This example depicts numerous problems, such as: wrong or missing label, prolonged roadside occlusion, ample road width variations. It also highlights the gains of multiple dilation rates and fixed width training.

Table 2. Roads segmentation results. Results reported on 5603 training images and 623 validation images, randomly selected from the original training set, for which ground truth was provided.

Model	Our Training		Our Validation	
	IoU	F1	IoU	F1
Max dilation 32	0.6432	0.7824	0.6483	0.7883
Max dilation 48	0.6577	0.7913	0.6601	0.7957
Max dilation 64	**0.6591**	**0.7919**	**0.6640**	**0.7966**

Table 3. Intersection segmentation results. We report IoU scores for our training and validation split.

Input	Our Training	Our Validation
Roads only	0.5627	0.5517
RGB + Roads	**0.7112**	**0.6492**

Table 4. Roads segmentation results using our ensembles.

Model	Our Training		Our Validation	
	IoU	F1	IoU	F1
Ensemble 1	0.6356	0.7749	0.6345	0.7769
Ensemble 2	**0.7287**	**0.8506**	**0.6920**	**0.8239**
Ensemble 1+2	0.6514	0.7882	0.6412	0.7829

thicker than the label, therefore we experimentally determined that eroding the roads with 2 pixels (on both sides) yielded best results. Ensemble 2 was trained using the same setup as our previous models. The second ensemble learns the label distribution and has no need for additional processing. We only binarized the results at a fixed threshold of 128. Our best results were thus obtained using Ensemble 2 (as shown in Table 4 and Table 5).

4.1. Road thickness

Label thickness is an important aspect for detection - thin roads and large pavement areas generally yield poor detection performance. Furthermore, training with variable width can exacerbate the problem, resulting in even more missed thin roads. In order to investigate the impact of this issue on detection, we trained an additional network with thicker roads. As confirmed by Table 6, roads as thick connections are better than having a variable or thin road that misses out road segments.

Table 5. **Round 2**. Roads segmentation results on the official testing set, 1101 images. The results were provided by the submission site. Results reported after adding the missing links.

Model	IoU Testing
Baseline [5]	0.545
Ensemble 1	0.5788
Ensemble 2	**0.5862**
Ensemble 1+2	0.5785

Table 6. Road thickness study using Max dilation 32 model. Results reported on 5603 training images and 623 validation images, randomly selected from the original training set.

		IoU Our Training	IoU Our Validation
Same width	Thin ($\approx$ 4m)	0.6282	0.6123
Variable width	Thin (original)	0.6432	0.6483
	Thick (2x thin)	0.7254	0.6889

5. Conclusions

We propose a three stage approach for road segmentation. First, a multi-stage CNN produces multiple road segmentation maps. Second, their outputs are fused using another CNN and road vectors are generated from the binary mask. Third, based on the road vectors and the binary segmentation, we add missing links to improve the overall segmentation mask.

Acknowledgements: This work was supported in part by UEFISCDI, project PN-III-P4-ID-ERC-2016-0007 and the Romanian Ministry of European Funds, project IAVPLN POC-A1.2.1D-2015-P39-287.

References

[1] F. Bastani, S. He, M. Alizadeh, H. Balakrishnan, S. Madden, S. Chawla, S. Abbar, and D. DeWitt. RoadTracer: Automatic Extraction of Road Networks from Aerial Images. In *Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, June 2018.

[2] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4):834–848, 2018.

[3] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. *arXiv preprint arXiv:1802.02611*, 2018.

[4] D. Costea, A. Marcu, E.-I. Slusanschi, and M. Leordeanu. Creating roadmaps in aerial images with generative adversarial networks and smoothing-based optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2100–2109, 2017.

[5] I. Demir, K. Koperski, D. Lindenbaum, G. Pang, J. Huang, S. Basu, F. Hughes, D. Tuia, and R. Raskar. Deepglobe 2018: A challenge to parse the earth through satellite images. *arXiv preprint arXiv:1805.06561*, 2018.

[6] 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*, pages 770–778, 2016.

[7] C. Henry, S. M. Azimi, and N. Merkle. Road segmentation in sar satellite images with deep fully-convolutional neural networks. *arXiv preprint arXiv:1802.01445*, 2018.

[8] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

[9] P. Kaiser, J. D. Wegner, A. Lucchi, M. Jaggi, T. Hofmann, and K. Schindler. Learning aerial image segmentation from online maps. *IEEE Transactions on Geoscience and Remote Sensing*, 55(11):6054–6068, 2017.

[10] D. Lam, R. Kuzma, K. McGee, S. Dooley, M. Laielli, M. Klaric, Y. Bulatov, and B. McCord. xview: Objects in context in overhead imagery. *arXiv preprint arXiv:1802.07856*, 2018.

[11] M. Leordeanu and M. Hebert. Smoothing-based optimization. 2008.

[12] A. Marcu, D. Costea, E. Slusanschi, and M. Leordeanu. A multi-stage multi-task neural network for aerial scene interpretation and geolocalization. *arXiv preprint arXiv:1804.01322*, 2018.

[13] D. Marmanis, K. Schindler, J. D. Wegner, S. Galliani, M. Datcu, and U. Stilla. Classification with an edge: improving semantic image segmentation with boundary detection. *ISPRS Journal of Photogrammetry and Remote Sensing*, 135:158–172, 2018.

[14] G. Máttyus, W. Luo, and R. Urtasun. Deeproadmapper: Extracting road topology from aerial images. In *International Conference on Computer Vision*, volume 2, 2017.

[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.

[16] OpenStreetMap contributors. Planet dump retrieved from https://planet.osm.org . `https://www.openstreetmap.org`, 2017.

[17] 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*, pages 234–241. Springer, 2015.

[18] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[19] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *AAAI*, volume 4, page 12, 2017.