# DeepRoadMapper: Extracting Road Topology from Aerial Images

Gellért Máttyus, Wenjie Luo and Raquel Urtasun
Uber Advanced Technologies Group
University of Toronto
{gmattyus, wenjie, urtasun}@uber.com

## Abstract

*Creating road maps is essential for applications such as autonomous driving and city planning. Most approaches in industry focus on leveraging expensive sensors mounted on top of a fleet of cars. This results in very accurate estimates when exploiting a user in the loop. However, these solutions are very expensive and have small coverage. In contrast, in this paper we propose an approach that directly estimates road topology from aerial images. This provides us with an affordable solution with large coverage. Towards this goal, we take advantage of the latest developments in deep learning to have an initial segmentation of the aerial images. We then propose an algorithm that reasons about missing connections in the extracted road topology as a shortest path problem that can be solved efficiently. We demonstrate the effectiveness of our approach in the challenging TorontoCity dataset [23] and show very significant improvements over the state-of-the-art.*

## 1. Introduction

Creating maps of our roads is a fundamental step in many application domains. Having accurate maps is essential to the success of autonomous driving for routing, localization as well as to ease perception. Building smart cities requires understanding the road network as well as the traffic patterns that occur on it to enable faster commute times, better public transportation systems and a healthier environment.

Most self-driving teams and mapping companies rely on expensive sensors mounted on a fleet of vehicles which drive around, mostly capturing LIDAR point clouds. A semi-manual process is then utilized to create the road network. Very accurate results can be achieved, but coverage is very limited. Furthermore, this is a very costly process. Thus HD maps are available for only a small region of the world.

An alternative approach is to use aerial and satellite images as data source. This is appealing as they have much larger coverage. For example, satellites go around the world twice a day, providing up-to-date information. However, extracting road networks from this imagery is very challenging, as the resolution is much lower. Further, occlusion (e.g., trees) and large shadows cast by tall buildings are difficult to handle. Most existing approaches cast the problem as semantic segmentation. Unfortunately, this ignores topology, which is the basic unit needed in order to perform driving. Recently, [13, 14] leveraged existing maps to enhance them with road-width as well as information about the number of lanes, their location, parking spaces and sidewalks. However, these approaches cannot reason about roads that are not present in the initial coarse map.

Figure 1: Road topology from aerial images at a large scale. Our extracted road network is shown in blue.

In contrast, in this paper we propose an approach that directly estimates road topology from aerial images. Towards this goal, we take advantage of the latest developments in deep learning to have an initial segmentation of the aerial images. We then propose an algorithm that reasons about missing connections in the extracted road topology as a shortest path problem that can be solved efficiently. We demonstrate the effectiveness of our approach in the challenging TorontoCity dataset [23], and show very significant improvements over the state-of-the-art.

2380-7504/17 $31.00 © 2017 IEEE
DOI 10.1109/ICCV.2017.372

## 2. Related work

Many approaches have been proposed in the last decades to extract road segmentation from aerial and satellite images. Several methods extract low level features and define heuristic rules (e.g. connectivity, shape) over these to classify road like structures. Geometric-stochastic road models based on assumptions about the width, length and curvature of the road and the pixel intensities have been exploited in [2]. Hinz and Baumgartner [10] use road models and their context including knowledge about their radiometry, geometry and topology. In [11], homogeneous areas are detected based on their shape and a road tree is then grown by tracking the roads. The drawback of these heuristic rule based models is that obtaining the optimal set of rules and parameters is very difficult. This is particularly challenging due to the high variety of roads. As a consequence these methods can work only on areas (e.g. rural) where the used features (e.g. image edge) occur predominantly at roads.

Convolutional neural networks have been used to segment roads from aerial images [15]. The neural network is applied at the patch level in multiple stages (with the previous prediction as input) to capture more context and structure. In [16], existing maps are used for data augmentation. Unfortunately connectivity of roads is not guaranteed in this approach. In [4], the roads are detected by a deep neural network applied to image patches. The extracted road network is then matched to a road database (i.e., OpenStreetMap) and the two road maps are merged.

Connectivity is probably one of the most important road features. However, this has been rarely studied in both the computer vision and photogrametry communities. Chai et al. [3] define a junction point process that reasons about the graph describing the road network. The process uses various priors, e.g. homogeneity of the pixel values, connectivity of the graph, edge orientation and line width. However, optimization is hard as it requires Reversible Jump MCMC sampling. In [19], a Point Process is defined to describe the interaction of line segments (e.g., connectivity). The road network is extracted by minimizing an energy function using simulated annealing. In [22], the road extraction is limited to tree structures. This guarantees the connectivity and the optimization can be solved exactly. Unfortunately roads are not tree-structured, posing a significant limitation. This approach was further extended to loopy graphs in [21], where the NP hard problem is approximately solved by a branch and cut algorithm. Wegner et al. [24, 25] segment the image into superpixels and the ones with high road likelihood are connected by a shortest path algorithm with the goal of creating an overcomplete representation of the road network. These paths are then handled as higher order cliques in a Conditional Random Field (CRF). As shown in our experiments this method does not produce very accurate results, and is an order of magnitude slower than our approach.

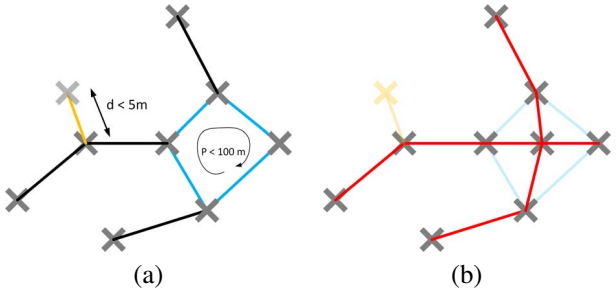



Figure 2: Road graph extraction: Nodes are road segment endpoints (crosses). (a) Graph constructed via thinning. Small branches (orange) are removed and small loops (blue) are replaced by a tree providing the same connectivity to the rest of the graph. (b) final graph (red).

Most approaches considered road segmentation as a binary problem, however roads can have various categories which are important for mapping. Mattyus et al. [13] improve existing freely-available road maps by extracting road width information and by correcting the position of the centerline. In [14], aerial and ground images are utilized jointly in order to extract fine-grained road information like the number of lanes, presence of sidewalks and parking lanes. Crowd-sourced manual labeling as well as GPS trajectories have been exploited to create road topology. This is the case of the OpenStreetMap project [1], in which volunteers have mapped more than half of the world. Recorded GPS tracks were also employed to help the road segmentation [27].

## 3. Road Topology from Aerial Images

In this paper we want to extract a graph representation of the road network from aerial images. In this graph the nodes represent end points of street segments and the edges encode the connections between these points, defining the road segment center lines. Towards this goal, we exploit convolutional neural networks (CNNs) to segment the images into the categories of interest. Then a simple process based on thinning extracts the road centerlines from the segmentation output. Errors in the segmentation can result in discontinuities, which translate into topological errors of the extracted road network. To alleviate this problem we further reason about the presence or absence of edges in an augmented road graph which contains also connection hypotheses covering the disconnects. As shown in our experiments this improves significantly our estimated road network.

### 3.1. Semantic segmentation of Aerial Images

In this section we describe the architecture we employ to segment aerial images. Following current trends in seman-

Figure 3: Segmentation softmax is highlighted in green, the extracted road center line is shown in red, and the connection hypotheses generated by the $A^*$ search are in blue. Dashed yellow shows other possible connections which were not selected by the A* algorithm.

tic reasoning from ground images we develop a variant of ResNet [8] to perform this task. Similar to FCN [18], it consists of an encoder that compresses the image into a small feature map, and a fully convolutional decoder, which generates the segmentation output probabilities.

Our encoder consists of a ResNet block with 55 convolutional layers with $3 \times 3$ kernels. We use a convolutional layer with stride 2 after 6 residual block forming 13 convolutional layers. This divides the whole encoding network in 4 parts. We use 16, 32, 64 and 128 kernels in each of these parts respectively. This gives us a feature map of 128 dimension with 1/8 of the original resolution.

The decoder consists of 3 fully convolutional layers with number of kernels 64, 32 and 16 respectively. Each of these layers upsamples its input to be double its resolution. In order to capture details, each of these layers takes feature maps directly from the encoder network as input as well as two additional skip connections from the stride convolution. The last convolutional layer converts the feature map into scores follow by a softmax with three outputs: road, building and background. Thus the whole network consists of 55 convolutional layers for the encoder, 3 fully convolutional layers for the decoder, follow by a convolutional layer to output the class labels.

Segmentation networks are typically trained via cross-entropy. However, the metric of interest at test time is typically the intersection over union (IoU), which is defined as

$$\frac{1}{|C|}\sum_c \frac{\sum_i \mathbb{1}\{y_i = c\} \cdot \mathbb{1}\{y_i^* = c\}}{\sum_i \mathbb{1}\{y_i = c\} + \mathbb{1}\{y_i^* = c\} - \mathbb{1}\{y_i = c\} \cdot \mathbb{1}\{y_i^* = c\}}$$

where $y_i$ is the prediction, $y_i^*$ the ground truth and $c$ is a class label.

In this paper, we develop a novel soft IoU loss, which is differentiable and thus amenable to back propagation. In particular, it is defined by replacing the indicator functions with the softmax outputs

$$\ell_{soft-IOU} = \frac{1}{|C|}\sum_c \frac{\sum_i p_{ic} \cdot p_{ic}^*}{\sum_i p_{ic} + p_{ic}^* - p_{ic} \cdot p_{ic}^*}$$

where $p_{ic}$ is the prediction score at location $i$ for class $c$, and $p_{ic}^*$ is the ground truth distribution which is a delta function at $y_i^*$, the correct label.

### 3.2. Road graph generation

Once we have an estimate of the semantic segmentation, the next step is to produce a graph representing the topology of the network. Towards this goal, we first generate a binary mask from the softmax output of the deep network by thresholding the road class at $0.5$ probability. Then we apply thinning [28] to extract the road centerlines, i.e., a one pixel wide representation of the road segments preserving the connectivity of the components. This results in a graph, where every pixel is a node. To simplify the graph we employ the Ramer–Douglas–Peucker algorithm [17, 6], which outputs a piecewise linear approximation of our road skeletons. In particular, we use an error tolerance of $\epsilon = 1.5m$. Note that our thinning procedure creates separate branches at topological defects of the segmentation mask. Many of them are small curves, which are not real centerlines. We thus remove curves with length smaller than $5m$. This is illustrated in Fig. 2.

Another potential problem are small holes in the segmentation mask, which cause undesired loops in our connectivity graph. We thus convert each loop of size smaller than $100$ m into a tree (star architecture) which provides the same connectivity to the nodes outside of the loop [1]. We refer the reader to Fig. 2 for an example. Note that we will only violate connectivity at roundabouts, which are rare in North America, where our source imagery is captured. This gives our representation of the road network graph, where nodes are end-points of the road segments and edges define the curves connecting these points.

### 3.3. Generating connection hypotheses by $A^*$ search

Our segmentation algorithm is accurate, but discontinuities of the resulting mask can cause errors in topology. Fig. 3 shows an example where the road in the left is disconnected. To alleviate this problem, we reason about potential missing street segments in order to further improve the topology. We define a *leaf node* as a node with a single connection. This represents the end of a road according to our

[1] The loop nodes connected to nodes outside the loop are preserved, the rest are removed and a new node is inserted in the center of the loop and is connected to all the preserved nodes.

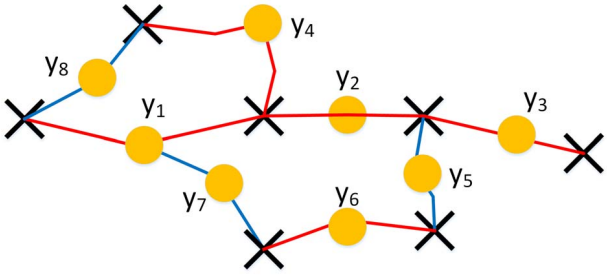



Figure 4: Each road segment (orange dots) is a random variable. Red lines show centerlines extracted from the segmentation. Hypothesis connections are shown as blue lines. This is the dual of the graph that describes the road network.

Figure 5: Two examples for the connection classifier. (Left) negative example, (Right) positive example.


(a) (b)

Figure 6: Illustration of the ground truth (GT) assignment. (a) GT graph in black and the extracted graph in blue. The subfigure shows two types of augmented edges. The dashed red connect the GT endpoints (A and B) to our extracted graph, while the dashed blue connect the discontinuities of our extracted graph. (b) The shortest path between A and B is shown in orange where the dashed line highlights the augmented edges. The shortest path defines the assignment and thus the true positives area, i.e., the projection on the ground truth (green). False positives are segments outside the shortest path (red), false negative length is the missing part of the ground truth, shown in black.

current topology estimate. We generate connections from the leaf nodes to other nodes if they lie within $50m$ and the shortest path in the graph between the two nodes is larger than $100m$, to prevent creating small loops in the graph.

Following this procedure, a leaf node can be connected to many nodes which provide the same connectivity. This is illustrated in Fig. 3, where possible connections are shown in dashed yellow. We exploit the $A^*$ algorithm [7] to select from these connections. A* is a shortest path algorithm that applies a cost heuristic to determine the next nodes to visit. If this heuristic is close to the real cost, then the search is efficient. We utilize the probability score of being non-road as our node cost, the distance as our edge cost and the euclidean distance as our heuristic. The algorithm runs very fast as most nodes are not visited during the search.

### 3.4. Reasoning about the connections

So far we have shown how to estimate possible connections between road segments. We now define an algorithm that decides the validity of these connections. Towards this goal, we reason about the hypothesized connections as well as the original road segments to prune false positives. We represent each road segment/connection with a binary variable $y_i \in \{0, 1\}$ representing the presence/absence of that road segment. Note that this is the dual of the graph which describes our road network. We refer the reader to 4 for an illustration.

To perform this task, we exploit a variety of potentials which depend on a single road segment. Our features are the soft-max scores along the road segment, the distance to the closest non road pixel, the length of the segment, a binary feature encoding if the node represents a connection hypothesis and the number of connections to other road sections. Since a road segment defines a curve, we calculate the features along the curve by employing different pooling strategies. In particular we employ min, max and average pooling to form additional features.

Deciding if a connection hypotheses is true is a difficult classification problem, especially since our deep semantic segmentation algorithm has already failed in this region. We thus utilize an additional network that classifies whether the hypothesized connection is a true connection or not, and employ the output of this classifier as an additional feature. The input to this classifier is a cropped image around the connection with the connection drawn on the image. We refer the reader to Fig. 5 for an illustration. In particular, we use an Inception network [20] to perform this classification.

Inference in this model is trivial and can be done in parallel as all our features depend on a single road hypothesis. We next describe how to perform learning.

## 4. Learning and Metrics

One of the difficulties we need to tackle is the fact that the ground truth graph and the estimated graph have different topology. Furthermore, the road hypothesis on both graphs have also different shape. In order to both do learning and evaluate our results we need to be able to register the two graphs and come up with the true labeling in the

hypothesized graph. In this section we describe how to do this task.

### 4.1. Assignment of GT roads to extracted roads

Our first goal is to assign the ground truth (GT) roads to the extracted roads in the predicted graph. We consider this assignment as a set of shortest path problems defined between each intersection and the road ends connected to that intersection in the ground truth network. To ensure that the connection goes along a similar path as the ground truth road, we only include as hypothesis the extracted roads located in a fix radius around the ground truth.

Note that in principle there will be many cases where there is no possible path, as we might have disconnects in the extracted graph. To handle this, we augment our graph with edges connecting the end points of the ground truth graph to the end points in the extracted graph. Furthermore, we also include edges that encode the missing connections. Fig. 6 shows an illustration of this process, where on (a) the extracted graph is shown in blue and the additional edges are shown in dashed blue and red.

We then solve the assignment problem by calculating the shortest path between the endpoints, where the distance between the adjacent points $p_i$ and $p_{i+1}$ is calculated as

$$D(p_i, p_{i+1}) = \sum_{j=i}^{i+1} \phi_d(p_j) + \lambda(p_i, p_{i+1})||p_i - p_{i+1}||$$

with $\lambda(p_i, p_{i+1}) = 1$ if the edge existed and $\lambda(p_i, p_{i+1}) = c$, with $c$ as a large constant if the edge is an augmented edge. $\phi_d(p_j)$ measurs the distance to the closest ground truth road edge. This ensures that the shortest path lies close to the ground truth. The minimum path can be solved by the Dijkstra algorithm [5]. Since the augmented edges have very high cost, they will only be selected if there is no other choice (the extracted graph is not connected). We refer the reader to Fig. 6 for an example.

### 4.2. Learning

We can utilize the procedure we just described to compute our ground truth in terms of the extracted graph, i.e., our $y_i$'s. In particular, all the original edges on the shortest path are considered as true positives, while the edges that are not part of any shortest path are considered as false positives. We use this assignment as our ground truth to perform learning. We train our model using max margin loss and use the Hamming distance as the task loss.

### 4.3. Evaluating Topology

The most commonly employed metrics in the literature are pixel-based and measure semantic segmentation [15, 4, 24, 25, 3]. However, these metrics do not reflect the quality of the extracted topology. Defining a graph based metric is non-trivial as there is no simple answer to the question of when are two graphs similar.

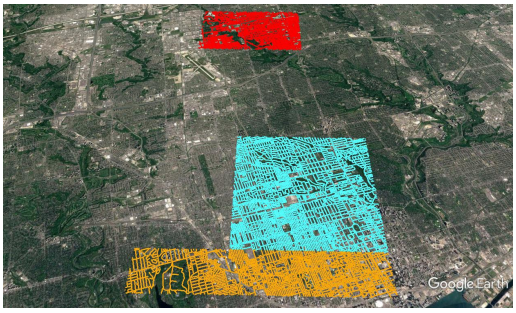



Figure 7: TorontoCity demo area shown on Google Earth. Cyan: train, orange: validation, red: test set.

Wenger et al. [24, 25] propose a connectivity metric measuring if the shortest path distances between randomly sampled correct pixels are the same in the ground truth and the extracted road network. This metric reflects topological similarity, but it is hardly reproducible due to the random selection of end points to form the shortest path problem. Furthermore, only a few end points are selected and thus this can miss many topological changes.

In this paper we propose a new set of metrics, which are based on the assignment of the ground truth roads to the extracted roads. We believe that these metrics better capture the extracted topology. We define the precision of each segment as the ratio of the true positive length $d_p^*$ to the extracted length $d_p$. We compute the final precision by computing the average precision of each segment weighted by its ground truth length $d_p^*$

$$pr = \frac{1}{\sum_p d_p^*} \sum_p \frac{\min(d_p, d_p^*)}{\max(d_p, d_p^*)} d_p^*$$

Note that to take into account the fact that the ground truth length can be smaller or greater than the estimated length we use the ratio of min and max values. We define recall as simply the percentage of road that is recovered with respect to the ground truth $d_{p,gt}$

$$rec = \frac{\sum_p d_p^*}{\sum_p d_{p,gt}}$$

We report $F1$, which is the harmonic mean of precision and recall.

$$F1 = 2\frac{pr \cdot rec}{pr + rec}$$

Additionally, we define a metric capturing the ratio of road segments which were estimated without discontinuities. We call this *Connected Road Ratio (CRR)*.

$$CRR = \frac{N_{con}}{N_{gt}}$$

| | IoU | F1 | precision | recall |
|---|---|---|---|---|
| [25] | 41.6 | 58.8 | 46.3 | 80.5 |
| [25] + Our Deep | 68.4 | 81.2 | 75.7 | **87.7** |
| Ours | **76.4** | **86.6** | **87.7** | 85.6 |

Table 1: **Semantic Segmentation of the road class**: our approach significantly outperforms [25], even when the baseline utilizes our deep semantic segmentation algorithm.

| | F1 | Precision | Recall | CRR |
|---|---|---|---|---|
| OSM (human) | 89.7 | 93.7 | 86.0 | 85.4 |
| [25] | 39.7 | 26.1 | 82.6 | 76.8 |
| [25] DeepUn | 63.1 | 50.0 | 85.7 | 78.4 |
| HED [26] | 42.4 | 27.3 | 94.9 | 91.2 |
| Ours basic | 78.0 | 71.2 | **86.2** | **79.1** |
| Ours - NoDeepCon. | 83.9 | 84.4 | 83.3 | 77.6 |
| Ours full | **84.0** | **84.5** | 83.4 | 77.8 |

Table 2: Topology recovery metrics (percentage).

with $N_{con}$ the number of road segments extracted without discontinuities and $N_{gt}$ the number of GT segments.

## 5. Experiments

We perform our experiments on the demo region of the TorontoCity dataset [23]. We use the pixelwise annotations of this dataset to train our semantic segmentation network with 3 classes (i.e., background, road, building).

**Dataset:** The TorontoCity demo region includes aerial images over $25km^2$ for training, $12km^2$ for validation and $17.5km^2$ for test. All sets are typical North American urban areas with both skyscrapers and family houses. The imagery consists of $5000 \times 5000$ pixel orthorectified images with 10 cm/pixel resolution and RGB color channels. The dataset provides pixelwise annotation of the road surfaces and the buildings plus vector data defining the road center lines and the connectivity between the roads (i.e. the road graph). Fig. 7 shows an overview of the dataset. Note that this is a very large area compared to datasets typically employed in the literature.

**Baselines:** We compare our method to the work of Wenger et al. [24, 25]. We resized the images to 25 cm/pixel (the same as in [25]) and used the default parameter setting provided in the authors' code. Note that without resizing, their approach takes more than an hour per image. We add an additional comparison by combining the Markov random field of [25] with our deep neural network estimates, in order to give this approach the opportunity to leverage deep learning. Towards this goal, we replaced their random forest unary features with our semantic segmentation softmax

| gt\pred. | 0 | 1 |
|---|---|---|
| 0 | 71.6 | 28.4 |
| 1 | 32.9 | 67.1 |

Table 3: Confusion matrix of connection classifier

values. Since [25] does not create a vector representation of the road, we use our graph generation method (described in Section 3.2) to convert it to a graph which we can then evaluate. We compare also to HED [26] which predicts the road centerlines directly by a deep net. We directly employ their code. Additionally, we compare to the freely available OpenStreetMap project [1] road maps as baseline. This can be considered as human performance on this task. We neglect very small road categories in OSM (i.e., path, cycleway, service, footway and path), as they are not labeled in TorontoCity.

**Learning details:** We used the training set for training the deep network and the validation set for training the SVM. Our network takes as input $1440 \times 1440$ resolution patches randomly cropped from the raw images with random flips. The network is trained with Adam [12] for 80 rounds, with the initial learning rate of $1e-3$, which we drop by a factor of 5 at round 40 and 60. We use batch size of 3 perform 300 iterations for each round. The weights are initialized using MSR initialization [9]. Training the network took around 16 hours. To train the Inception network that reasons about whether a connection in the graph exists, we employ a training set of $22,000$ images, which were generated by creating connection hypotheses on the training set ($1/3$ of the samples), by generating additional examples from the road graph ($1/3$) and by adding negative examples ($1/3$) randomly picked around road areas. For training we use a learning rate of $0.001$, a momentum of $0.9$ and train the network for 100 epochs.

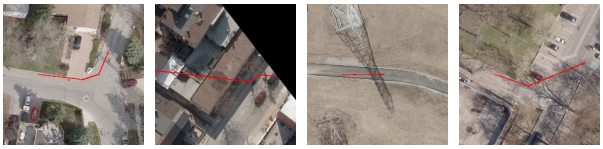
**Metrics:** We report four types of pixel-wise metrics to show the quality of our semantic segmentation. This includes intersection over union (IoU), precision, recall and F1. We additionally report the metrics described in Section 4, which tests the accuracy of the shape and topology of the road network. We apply a radius of $20m$ around the ground truth roads, everything else cannot be a true positive.

**Soft IoU vs. cross-entropy loss:** Table 4 shows a comparison between applying cross-entropy and soft IoU loss for semantic segmentation. Soft IoU is considerably better.

**Comparison to the state-of-the-art:** As shown in Table 1 our method outperforms the baseline [25] by a large margin, even when we provide our segmentation scores as

| | mean | bg | road | building | accuracy |
|---|---|---|---|---|---|
| cross-entr. | 71.6 | 74.8 | 69.9 | 70.3 | 84.1 |
| soft IoU | **75.6** | **80.2** | **75.1** | **71.5** | **87.0** |

Table 4: IoU for 3 classes on the validation set. Soft IoU loss is considerably better.



(a) TP (b) TN (c) FP (d) FN

Figure 8: Examples of connection classifier: TP: true positive, TN: true negative, FP: false positive, FN: false negative. This is a difficult classification problem even for humans. Many road like structures (drivable surfaces) are not part of the road network ground truth, e.g. in (c).

unary potentials. The baseline tends to create an overcomplete road extraction resulting in higher recall but smaller precision. As shown in Table 2 our approach also significantly outperforms the baseline in terms of topology. In particular, [25] has high recall but very bad precision. By applying our deep unary potentials the baseline improves, but is sill behind our approach. HED [26] also predicts roads at every road-like surface, resulting in high Recall and CRR but very low Precision and F1. HED has a difficult time predicting the skeleton, as in contrast to edge detection, it is not aligned with image edges but lies over homogeneous road surfaces. Having low precision is problematic in practice since a manual operator doing quality control would need to delete many extracted roads, which most likely will take longer than adding missing roads. The CRR metric is correlated with Recall and is very sensitive to small discontinuities, while it is agnostic to false positives. Therefore, methods that create over-complete road networks with low precision achieve higher numbers. This is why any single metric is not good in isolation.

**Running time:** Our approach is very efficient. The segmentation network takes approximately $1s$ on an NVIDIA TITAN X GPU. Classifying an image takes around $1s$ running on the same GPU. The rest of the pipeline is implemented in C++ single threaded. It takes $90s$ on average to process a $5000 \times 5000$ pixel image. In contrast, [25] needs around 30 minutes per $2000 \times 2000$ pixel image running multi threaded on an Intel Xeon E5-2690 CPU. We used the Matlab and C++ implementation provided by the authors.

**Comparison to human performance:** OpenStreetMaps have been generated by a combination of manual labeling and recorded crowed-sources GPS trajectories. As shown in Table 2 our approach is not far behind OSM. This is remarkable, taking into account the enormous manual labeling task that OSM required. OSM achieves almost $90\%$ F1. The fact that it is not perfect shows the difficulty of the task.

**Ablation studies:** We compared three different instantiations of our model. Our *basic* algorithm only extracts the road center line from our semantic segmentation without reasoning about connectivity. Our second version reasons about connectivity but does not utilize the Inception classifier. The last version is our full model. As shown in Table 2 our basic method provides decent results. If we employ reasoning about the existence of roads, precision increases and thus also F1, while the recall as well as the ratio of roads covered decrease only slightly. Adding the deep unary connection classifier improves the performance only slightly. This classification task is a very difficult and the accuracy is only around $70\%$ as shown in Table 3.

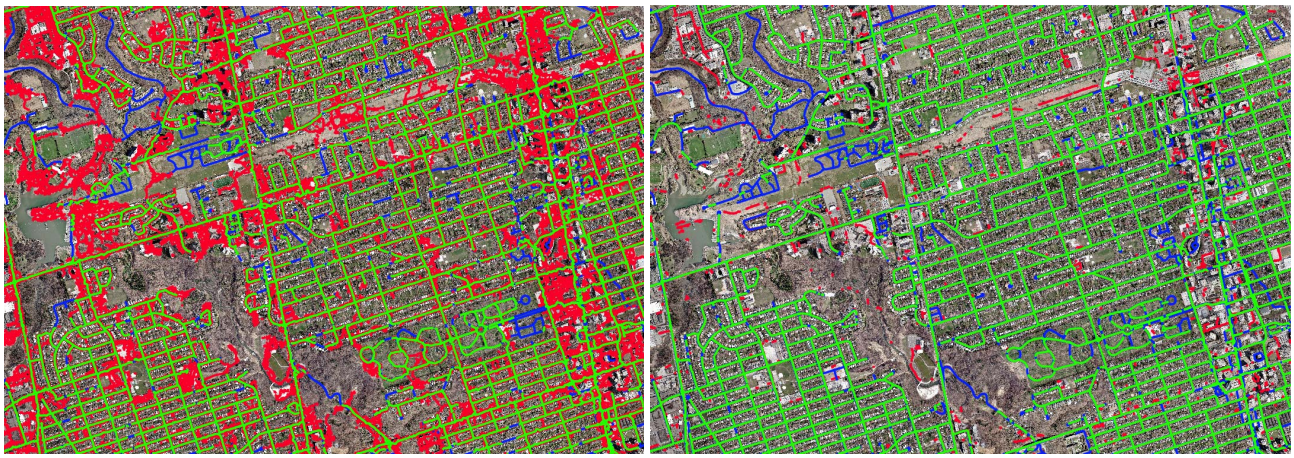
**Qualitative Results:** Fig. 9 shows results over the test set for the baseline and our method. The baseline is much more susceptible to create false positives, which reduces the precision significantly. Finally, Fig. 10 shows details of the extracted road networks. Our method can produce very similar results to the TorontoCity ground truth as well as OSM (human annotation). The extracted centerlines follow the road curves, and very few false positive roads exist. Typical errors are due to tall buildings and wide roads where the segmentation fails, e.g., due to occlusion by the buildings. Small interruptions in the connectivity around intersections are shown in Fig. 10.

## 6. Conclusion

In this paper we have presented an approach that directly estimates road topology from aerial images. This provides us with an affordable solution that has large coverage. Towards this goal, we have taken advantage of the latest developments in deep learning to have an initial segmentation of the aerial images. We have then derived an algorithm that reasons about missing connections in the extracted road topology as a shortest path problem which can be solved efficiently. We have demonstrated the effectiveness of our approach in the challenging TorontoCity dataset [23] and show very significant improvements over the state-of-the-art. In the future we plan to extract additional information such as building footprints in order to enrich our maps.

## 7. Acknowledgments

This research was funded by NSERC, CFI, ORF, ERA, CRC, the RBC Innovation Fellowship. We thank the authors of [24] for providing their code, and NVIDIA for donating GPUs.



(a) [25] (b) Ours

Figure 9: Visualization of the results on the entire test set ($17.5km^2$, $1.75 \cdot 10^9$ pixels). Green: True positive, red: false positive, blue: false negative. On the left (a): baseline [25], on the right (b): our method.

(a) [25] (b) Ours (c) OSM (human) (d) GT

Figure 10: Road centerlines (blue). We can produce similar results to humans and much better estimates than the baseline.

## References


[1] https://www.openstreetmap.org. 2, 6

[2] M. Barzohar and D. Cooper. Automatic finding of main roads in aerial images by using geometric-stochastic models

and estimation. *PAMI*, 1996. 2

[3] D. Chai, W. Forstner, and F. Lafarge. Recovering line-networks in images by junction-point processes. In *CVPR*, 2013. 2, 5

[4] D. Costea and M. Leordeanu. Aerial image geolocalization from recognition and matching of roads and intersections. *CoRR*, abs/1605.08323, 2016. 2, 5

[5] E. W. Dijkstra. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1959. 5

[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*. John Wiley & Sons, Ltd, 2011. 3

[7] P. E. Hart, N. J. Nilsson, and B. Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE TSSC*, 1968. 4

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015. 3

[9] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *CoRR*, abs/1502.01852, 2015. 6

[10] S. Hinz and A. Baumgartner. Automatic extraction of urban road networks from multi-view aerial imagery. *ISPRS JPRS*, 2003. 2

[11] 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. *TGRS*, 2007. 2

[12] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. 6

[13] G. Mattyus, S. Wang, S. Fidler, and R. Urtasun. Enhancing road maps by parsing aerial images around the world. In *ICCV*, 2015. 1, 2

[14] G. Mattyus, S. Wang, S. Fidler, and R. Urtasun. Hd maps: Fine-grained road segmentation by parsing ground and aerial images. In *CVPR*, 2016. 1, 2

[15] V. Mnih and G. E. Hinton. Learning to detect roads in high-resolution aerial images. In *ECCV*, 2010. 2, 5

[16] V. Mnih and G. E. Hinton. Learning to label aerial images from noisy data. In *ICML*, 2012. 2

[17] U. Ramer. An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing*, 1972. 3

[18] E. Shelhamer, J. Long, and T. Darrell. Fully convolutional networks for semantic segmentation. *TPAMI*, 2016. 3

[19] R. Stoica, X. Descombes, and J. Zerubia. A gibbs point process for road extraction from remotely sensed images. *IJCV*, 2004. 2

[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 4

[21] E. Turetken, F. Benmansour, B. Andres, H. Pfister, and P. Fua. Reconstructing loopy curvilinear structures using integer programming. In *CVPR*, 2013. 2

[22] E. Turetken, F. Benmansour, and P. Fua. P.: Automated reconstruction of tree structures using path classifiers and mixed integer programming. In *CVPR*. 2

[23] S. Wang, M. Bai, G. Máttyus, H. Chu, W. Luo, B. Yang, J. Liang, J. Cheverie, S. Fidler, and R. Urtasun. Torontocity: Seeing the world with a million eyes. *ICCV*, 2017. 1, 6, 7

[24] J. D. Wegner, J. A. Montoya-Zegarra, and K. Schindler. A higher-order crf model for road network extraction. In *CVPR*, 2013. 2, 5, 6, 7

[25] J. D. Wegner, J. A. Montoya-Zegarra, and K. Schindler. Road networks as collections of minimum cost paths. *ISPRS JPRS*, 2015. 2, 5, 6, 7, 8

[26] S. Xie and Z. Tu. Holistically-nested edge detection. In *ICCV*, December 2015. 6, 7

[27] J. Yuan and A. M. Cheriyadat. Image feature based gps trace filtering for road network generation and road segmentation. *Machine Vision and Applications*, 2016. 2

[28] T. Y. Zhang and C. Y. Suen. A fast parallel algorithm for thinning digital patterns. *Commun. ACM*, 1984. 3