Promoting Connectivity of Network-Like Structures by Enforcing Region Separation

Doruk Oner, Mateusz Koziński, Leonardo Citraro,
Nathan C. Dadap, Alexandra G. Konings, Pascal Fua

Abstract—We propose a novel, connectivity-oriented loss function for training deep convolutional networks to reconstruct network-like structures, like roads and irrigation canals, from aerial images. The main idea behind our loss is to express the connectivity of roads, or canals, in terms of disconnections that they create between background regions of the image. In simple terms, a gap in the predicted road causes two background regions, that lie on the opposite sides of a ground truth road, to touch in prediction. Our loss function is designed to prevent such unwanted connections between background regions, and therefore close the gaps in predicted roads. It also prevents predicting false positive roads and canals by penalizing unwarranted disconnections of background regions. In order to capture even short, dead-ending road segments, we evaluate the loss in small image crops. We show, in experiments on two standard road benchmarks and a new data set of irrigation canals, that convnets trained with our loss function recover road connectivity so well, that it suffices to skeletonize their output to produce state of the art maps. A distinct advantage of our approach is that the loss can be plugged in to any existing training setup without further modifications.

Index Terms—Road Network Reconstruction, Aerial Images, Map Reconstruction, Connectivity.

1 Introduction

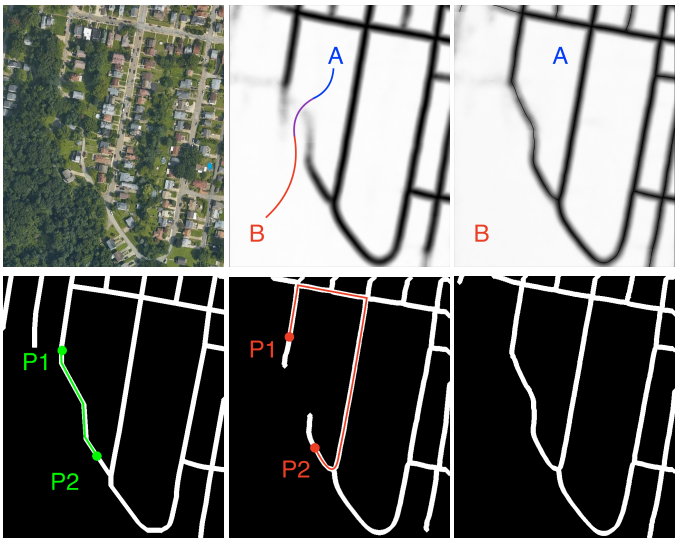
Reconstruction of road networks from aerial images is a classic computer vision problem [1], [2], [3], [4], which remains actively studied to this day [5], [6], [7], [8], [9], [10], [11], [12]. By contrast, the reconstruction of drainage canals was, so far, out of focus of the vision community. However, it is of practical importance for hydrologic analyses [13], [14], which are becoming even more crucial at times of rapid climate changes. Due to their network-like structure, canals are amenable to reconstruction by the same algorithms as roads, and we address these two problems jointly. Most of the existing approaches [5], [15], [9], [12] rely on convolutional networks to extract from images binary masks denoting which pixels belong to roads and which do not. Unfortunately, they do not guarantee that the connectivity of the produced masks corresponds to that of the real road network. This is because these methods are trained to minimize losses, such as cross-entropy and mean squared error, that do not explicitly enforce topological consistency. When the annotations do not perfectly coincide with the imaged structures, which is always the case of satellite image annotations, networks trained with the per-pixel losses produce binary masks plagued by topological errors, such as road interruptions, missed junctions, and false positive connections.

In recent literature, this problem has been addressed by combining a convolutional encoder with a decoder that represents a network of roads as a graph, as opposed to a binary mask [8], [7], [10]. At inference time, the graph is grown iteratively: At each step, the neural network adds a new node to the graph by taking image features and the current state of the graph into account. By contrast to the approach based on representing a road map as a binary mask, these graph-based methods make it easy to prevent excessively penalizing predicted roads that deviate slightly from their ground truth models, and to account for existing connectivity when growing the graph. However, the non-differentiability of the node insertion operation makes training these networks more difficult and brittle than training convnets.

In this paper, we show that connectivity of road and drainage canal networks can be enforced directly on a convolutional neural net, in a fully differentiable manner, and without the need to represent the graph explicitly. This allows end-to-end training and results in increased performance. Our approach consists in relaxing the usual requirement of coincidence of annotated and predicted foreground pixels. Instead, we require that predictions contain uninterrupted sequences of foreground pixels that can deviate by a few pixels from the ground-truth annotations. This enforces connectivity while dealing with possibly imprecise annotations.

The difficulty is to express this requirement in the form of a differentiable loss function that can be used to train a deep network. The central idea of our approach is to forgo enforcing connectivity of the pixels annotated as centers of roads or canals, which may not coincide with true roads or canals. Instead, we express the connectivity of the annotated structures in terms of the disconnections that they create between regions annotated as background. More precisely, we require that two regions separated by a line in the ground truth, are also separated in the prediction. As shown in Fig. 1, this effectively enforces continuity of the predicted road or canal networks. By requiring that connected components of pixels annotated as background remain connected in the prediction, we prevent predicting false positive road or canal segments. To capture dead-ending segments, we compute our loss in small image windows, which are likely to be subdivided even by short road and canal sections. To enforce the (dis-)connectivity of image regions, we re-purpose the differentiable machinery

- *D.Oner, M.Koziński, L.Citraro and P. Fua are with the Computer Vision Laboratory, École Polytechnique Fédérale de Lausanne.*
- *N.C.Dadap and A.G.Konings are with the Remote Sensing Ecohydrology Group, Departemnt of Earth System Science, Stanford University.*



(a) (b) (c)

Fig. 1: **We enforce road connectivity by penalizing connections between background regions.** (a) Input image and ground truth. (b) A distance map predicted by a U-Net trained *without* our connectivity loss, and its skeletonization, thickened for visibility. Note that, even though there is a gap between road pixels P_1 and P_2, they remain connected both in the ground truth and in the prediction, because alternative paths exist in the loopy road network. By contrast, background regions A and B connect in the prediction, but not in the ground truth. (c) A distance map predicted by a U-Net trained using our disconnectivity loss its skeletonization. Our loss function penalizes connections between A and B, preventing gaps in the predicted road.

proposed in the MALIS segmentation algorithm [16], [17].

Our contribution therefore is a novel approach to enforcing global connectivity of reconstructions of network-like structures from images. It can be used to boost the performance of *any* road delineation deep network that outputs a binary mask of road versus non-road pixels, without having to change the network itself. This is in stark contrast to the graph networks that do require changing both the network architecture and the training procedure. We demonstrate on both roads and drainage canals, that a simple U-Net [18] trained with our loss function, and combined with a standard skeletonization algorithm, attains state of the art performance in terms of the connectivity of the reconstructed networks.

2 Related Work

The existing approaches to reconstruction of networks of drainage canals rely on dedicated sensing modalities, like multi-spectral imaging and lidar [19], [20], and require extensive user interaction. We show that the canals can be reconstructed from visual spectrum satellite images and with little required correction, just like roads.

Many existing road segmentation algorithms rely on convnets [5], [9], [15], [12], [11] and all of them face the same difficulty: Training them by minimizing a cross-entropy loss, which is a *local*, pixel-wise measure, does not guarantee that their output preserve the global connectivity of road networks. Training the network to multi-task and to find not only the road centerline but also its spatial extent [5], [9] or its orientation [9] mitigates the problem but does not explicitly enforce better connectivity. We instead propose to explicitly define the loss function to evaluate the connectivity.

2.1 Connectivity-oriented loss functions

Ours is not the first attempt to make a convnet capture connectivity of linear structures in images by incorporating connectivity-oriented terms in the loss function. One existing approach, is to use a perceptual loss function that depends on the statistical differences between features computed by forwarding either the ground truth or the prediction through a pre-trained neural network [15]. While this loss is indeed non-local, and has been shown to improve the connectivity of the predictions, it does not model connectivity explicitly. Instead, it heavily relies on the assumption, that a pre-trained neural network implicitly captures some topological properties of the input. By contrast, our loss function models connectivity explicitly.

Loss functions explicitly evaluating the topology of the predicted masks have been proposed for medical image segmentation [21], [22]. However, strictly topological techniques are focused on counting loops and connected components in the data, irrespectively of their spatial position, and cannot distinguish between different branching patterns. That makes these loss functions a good choice when the segmented object has a relatively simple topology, like the aortic valve, but not well suited for roads, which exhibit complex branching patterns and form numerous loops. Our loss function is intended for linear structures forming complex topologies, like roads.

2.2 Connectivity-oriented neural architectures

Problems with connectivity can be addressed by designing predictors that output graphs instead of per-pixel masks, and explicitly decide about the presence of connections between map nodes. This can be done as a post-processing step by generating a pool of potential additional connections and training a classifier to decide which of the candidates should be inserted into the network [6], [11]. One drawback of this approach is that it is not end-to-end trainable.

A more elegant alternative is to use graph neural networks to predict the road graphs directly from the images [8], [7], [10]. This approach has certain disadvantages. Inference consists in a sequence of non-differentiable node insertion operations, which makes such networks slower than convnets. They are also more difficult to train, because node insertion is conditioned on the current state of the graph, and heuristics are needed to decide what is the optimal operation when the graph built so far is inconsistent with the ground truth. In our experimental evaluation, we show that a simple convnet can outperform these approaches when trained with our loss function and post-processed with a vanilla skeletonization. However, we still think that predicting graphs from images has merit, and the idea could be applied on top of a convnet trained with our loss.

2.3 Affinity learning

To enforce region connectivity, we use the maximin formulation of MALIS [16], [17], a connectivity-oriented approach to segmenting cells in electron microscopy images of neural tissue. It relies on the observation, that the predicted strength of connection between a pair of pixels can be expressed as the lowest value that needs to be

crossed when traveling between the pixels in the prediction. If this value equals θ, thresholding the prediction with $\theta' < \theta$ produces a connected component containing both pixels. Thresholding the prediction with $\theta'' > \theta$ breaks the connection between the pixels. Formally, θ is called a maximin cost of a pixel pair, and MALIS incorporates it into a differentiable loss term which is maximized for all pairs of pixels that belong to the same annotated cell, and minimized for all pairs of pixels from different cells.

We could have used the same approach to enforce the connectivity of road or canal pixels in the output of a segmentation network. This would have been ineffective for two reasons. First, both roads and canals often form loops and even if a connection between two road pixels is missed, they may still be connected via a different path. As illustrated by Fig. 1, there is a gap between pixels P_1 and P_2. Yet they are still connected to each other. Hence, this disconnection cannot be fixed simply by enforcing connectivity of any road pixel pairs. Second, road and canal annotations usually take the form of one-pixel-thick centerline delineations that are rarely precise. Strictly enforcing the connectivity of centerline pixels would confuse the network and negatively impact its precision.

3 Method

Given a training set of N aerial images $\{x_i\}_{1\leq i\leq N}$ and corresponding ground-truth binary masks $\{y_i\}_{1\leq i\leq N}$ representing the roads or drainage canals in these images, we want to train a deep network $f_\Theta(\cdot)$, with weights Θ, that takes an image x as input and returns a distance map $\hat{y}$, consistent with the ground-truth. Our goal is to ensure that $\hat{y}$ represents the same connectivity as y. To this end, we minimize

$$R(\Theta) = \sum_i L\big(y_i, f_\Theta(x_i)\big) \ , \tag{1}$$

$$L(y,\hat{y}) = L_{\text{MSE}}(y,\hat{y}) + \alpha L_{\text{TOPO}}(y,\hat{y}) \ , \tag{2}$$

with respect to the network weights Θ. Here the loss function L is the sum of two terms L_{MSE} and L_{TOPO}, and α is a parameter of the method that we set empirically using a validation set. L_{MSE} is a regression loss, used to train the network to predict the distance from each pixel to the center of the closest road or canal as in [23]. This lets us penalize the deviation of the predicted road center from its annotated position more gently than when using the more standard cross entropy. Allowing for these deviations enables the connectivity-oriented L_{TOPO} to force the network to predict uninterrupted roads and canals even if they do not coincide perfectly with the annotations. We describe both terms below in more detail.

3.1 Regression Loss: L_{MSE}

We define L_{MSE} as the Euclidean norm of the difference between the predicted distance map $\hat{y}$ and the distance map y_D generated from the ground truth binary mask y

$$L_{\text{MSE}}(y,\hat{y}) = \sum_{p\in I} \big(\hat{y}[p] - \min(y_D[p], D_{\text{max}})\big)^2, \tag{3}$$

where $X[p]$ denotes the value of image X at pixel p, and I is the set of pixel indices in the input image. In practice, we found it advantageous to cap the ground truth distance at $D_{\text{max}} = 20$ pixels. Otherwise, the loss was strongly affected by large values of the distance map far away from foreground structures.

L_{MSE} can be used by itself to train the network. As shown in Fig. 1, this gives good results in terms of per-pixel precision but the resulting binary masks feature many unwarranted interruptions. To prevent this, we now turn to the second term of Eq. (2).

3.2 Connectivity Loss: L_{TOPO}

L_{TOPO} is the topology term that comprises the main contribution of this paper. Its purpose is to penalize in a differentiable manner unwanted interruptions and false connections in the output distance maps. Instead of explicitly penalizing the interruptions and false connections of the foreground, we formulate our loss function in terms of connectivity of the background regions. As shown in Fig. 1, an erroneous break in a predicted road causes two background regions, separated by a road in the ground truth mask y, to touch in the distance map $\hat{y}$ produced by the network. The first component of our loss, L_{disc}, penalizes such contacts. Similarly, a false positive road divides a small crop of the predicted distance map into two background regions, while the same crop of the ground truth distance map contains a single connected component of the background. Such errors are penalized by the second component of our loss, L_{conn}. The full topological loss takes the form

$$L_{\text{TOPO}}(y, f(x)) = L_{\text{disc}}\big(y, f(x)\big) + \beta L_{\text{conn}}\big(y, f(x)\big) \ , \tag{4}$$

where β is a parameter of the loss. We introduce L_{disc} and L_{conn} below.

3.2.1 *Maximin Dis-Connectivity.*

As illustrated in Fig. 1, in order to discourage interruptions of a predicted road, we identify all pairs of background regions that the road separates in the ground truth, and penalize connections between these regions in the predicted distance map. To that end, we follow the *maximin* approach of Turaga et al. [16]. Intuitively, since the value of a road or canal pixel in a correct distance map should be small, and the background pixels should be large, two background pixels in an image can be considered connected if there exists a path of large-valued pixels between them. The 'strength' of this connection, can be evaluated as the value of the smallest pixel on the path with the largest smallest pixel connecting the end points. Therefore, for each pair of pixels that are separated by a road or canal in the ground truth, L_{disc} contains the 'strength' of the connection between them. As a result, minimizing L_{disc} ensures the disconnectivity of regions on the opposite sides of roads and canals and, indirectly, improves the connectivity of roads and canals.

The detailed computation of L_{disc} is depicted in Fig. 2. We first dilate the centerline annotations by 5 pixels, which corresponds to the largest displacement between the image and the annotation that we have observed in our training data. We can therefore assume all the road pixels belong to this dilated region, which we denote as $\mathcal{R}$ and which can also contain non-road pixels. Let $\mathcal{B}$ be the set of background regions, that is, connected components in the remainder of the image. Let us consider two pixels $q \in A$ and $r \in B$ such that $A, B \in \mathcal{B}$ and $A \neq B$. Intuitively, q and r lie on different sides of an annotated road. Since road pixels should receive low predictions, a path π connecting q and r crosses a predicted road in the distance map $\hat{y}$ if, for at least one point p along the path, $\hat{y}[p]$ is close to zero. We therefore define the cost of path π in the predicted distance map $\hat{y}$ as $c(\pi,\hat{y}) = \min_{p\in\pi} \hat{y}[p]$, and measure the 'connectivity'

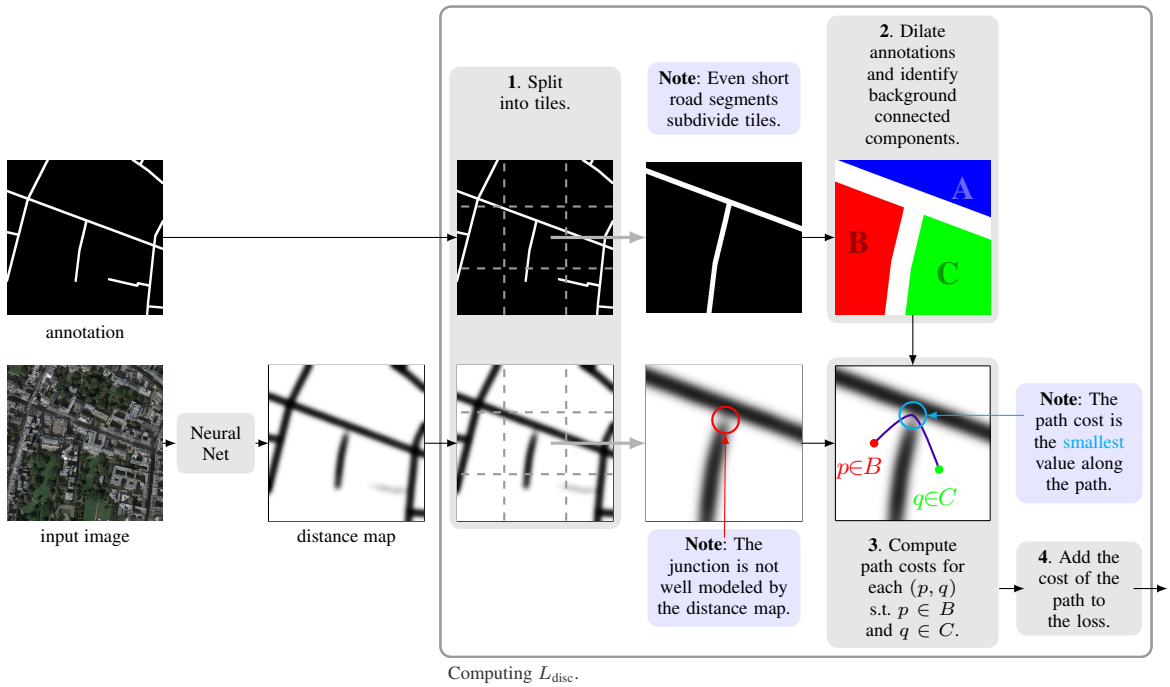


Fig. 2: **Computing L_{disc}.** We first tile the ground truth annotation and the distance map computed by our network (1). We use the ground-truth roads to segment each tile into separate regions (2). When there are unwarranted gaps in the distance map, there is a least one path connecting disjoint regions such that the the minimum distance map value along that path is not particularly small. We therefore take the cost of the path to be that minimum value (3) and we add to our loss function a term that is the maximum such value for all paths connecting points in the two distinct regions (4). This penalizes paths such as the one shown here and therefore promotes the road graph connectivity.

between background pixels q and r, in terms of the maximin cost $d_{\text{maximin}}(\hat{y}, q, r) = \max_{\pi \in \Pi(q,r)} c(\pi, \hat{y})$, where $\Pi(q, r)$ is the set of all paths connecting q and r. We enforce road connectivity by minimizing the maximin cost for all pairs of pixels that are separated by a road in the ground truth. To that end, we define our connectivity-enforcing loss as

$$L_{\text{disc}}(y, \hat{y}) = \sum_{A,B \in \mathcal{B},\ A \neq B} \sum_{q \in A, r \in B} d_{\text{maximin}}(\hat{y}, q, r)^2 \ . \quad (5)$$

When computed naively, the loss (5) requires summing costs over pairs of pixels, which would be computationally expensive. However, Turaga et al. [16], [17] have shown that, because $d_{\text{maximin}}(\hat{y}, q, r)$ is equal to the value of the smallest pixel that has to be visited when traveling between q and r in the prediction $\hat{y}$, L_{disc} can be computed efficiently as a sum over pixels, as opposed to pixel pairs, as

$$L_{\text{disc}}(y, \hat{y}) = \sum_{p \in \mathcal{R}} w_p \hat{y}[p]^2 \ , \quad (6)$$

where w_p counts the pairs of pixels whose maximin cost is equal to $\hat{y}[p]$. Formally, we denote the maximin path between a pair of pixels q, r by $\pi(q, r)$ and define

$$w_p = \sum_{A,B \in \mathcal{B},\ A \neq B} \sum_{q \in A, r \in B} \mathbb{1}\left[p = \arg \min_{\rho \in \pi(q,r)} \hat{y}[\rho]\right] \ , \quad (7)$$

where $\mathbb{1}[\cdot]$ is the indicator function. The algorithm for computing the w_p's is based on the Kruskal's maximum spanning tree algorithm, and we refer the reader to [17] for details. Note that, following [17], we constrain the computation of the loss L_{disc} to the dilated road regions $\mathcal{R}$. This speeds up convergence in the early stages of the training, when path minima may be found far away from true roads.

3.2.2 *Penalizing False Connections*

We could take L_{TOPO} to simply be L_{disc} but we have observed that this results in many false positive road segments and that this behavior is difficult to counteract only by balancing the regression and connectivity losses with the coefficient α in Eq. (2). To remedy this, we introduce another loss term that enforces connectivity of background regions, preventing false positive roads, as

$$L_{\text{conn}}(y, \hat{y}) = \sum_{A \in \mathcal{B}} \sum_{p \in A} v_p (\hat{y}[p] - y_D[p])^2 \ , \quad (8)$$

where v_p is the number of pairs of pixels $q, r \in A$, for which p is the smallest pixel on the maximin path between q and r, and is computed similarly to w_p, and $y_D[p]$ is the value of the ground truth distance map at pixel p.

3.2.3 *Introducing Sliding Windows.*

We can compute L_{TOPO} as described above on the whole image. However, when we do that, almost all pixels are assigned weights $w = 0$ in Eq. (6) and a single road pixel gets a weight equal to the product of the size of the connected components that the road should separate. This is because, in the presence of a an evident road interruption, all maximin paths go through this interruption. This might seem desirable in theory, but in practice it makes learning unstable. Since only a small minority of pixels generate extremely large gradients, no error signal is distributed among the remaining ones.

To overcome this problem we compute L_{TOPO} independently for 64×64 image patches that cover the image, and sum the

results. This ensures that at least one road pixel per window is taken into account and that its weight is not larger than $N^2/4$, where N is the number of pixels in the window. As shown in Fig. 2, this also lets us handle dead-ending roads that do not separate the global map into disjoint areas.

4 Experiments

We now describe the dataset we have tested our approach on, the baselines to which we compare our results, and the metrics we used to assess the quality of the reconstructions. We then demonstrate that our new loss improves the results of networks that rely solely on conventional losses and substantially outperform recently proposed road reconstruction methods.

4.1 Datasets

We performed experiments on three publicly available datasets.

- *RoadTracer*. A recently published dataset of high-resolution satellite images covering urban areas of forty cities in six different countries [8]. Fifteen cities are used as a validation set. The ground truth is generated using OpenStreetMap.
- *DeepGlobe*. Aerial images of rural areas in Thailand, Indonesia and India [24]. The dataset comprises around 8500 images, 6200 of which are used for training, 1200 for validation and 1100 for testing. For a fair comparison to [9], we use the same split, consisting in 4695 training and 1530 test images.
- *Canals*. Aerial images of water drainage canals in rural areas of Malaysia [25]. The dataset comprises a single large orto-photograph, 9768x10718 pixels large. 95% of the image is used as training data and the rest is for testing.

Together, these datasets exhibit a very large variation of terrain type, which makes them an exhaustive benchmark for aerial road and drainage canal network reconstruction.

4.2 Baselines

We compare the results of our algorithm to the following state-of-the-art methods.

- *Segmentation*. A baseline algorithm from [8], combining segmentation, thresholding, skeletonization, and conversion of the skeleton to a graph. Road network reconstructions for the RoadTracer dataset were made available online by the authors [26].
- *RoadTracer*. Iterative graph construction where node locations are selected by a CNN [8]. The road network reconstructions were released publicly by the authors [26].
- *Seg-Path*. A unified approach to segmenting linear structures and classifying potential connections [11]. The road network reconstructions were provided to us by the authors.
- *RCNNU-Net*. Recursive image segmentation with post-processing for graph extraction [12]. The authors provided the probability maps.
- *DeepRoad*. Image segmentation followed by post-processing focused on fixing missing connections [6]. The graphs for the RoadTracer dataset were published by the authors of this data set [26].
- *PolyMapper*. Reconstructing a map by sequential construction of closed polygons [7]. The graphs were provided to us by the authors.
- *MultiBranch*. A recursive architecture co-trained in road segmentation and orientation estimation [9]. To obtain the road network reconstructions, we trained the network using the code published by the authors.
- *LinkNet*. An encoder-decoder architecture [27] co-trained in segmentation and orientation estimation [9]. We trained the network using the code made available by the authors.
- *U-Net*. Our own implementation of U-Net [18] trained with mean squared error.
- *DRU* A recurrent U-Net iteratively refining segmentation output [28], trained by us with the mean squared error.

4.3 Network architecture and training details

We compare these baselines against three variants of our approach introduced in Section 3.

- *U-Net + TOPO-global*. A U-Net, trained with our connectivity loss computed in the full image.
- *U-Net + TOPO-windowed*. A U-Net, trained with our loss computed in windows of size 64×64 pixels.
- *DRU + TOPO-windowed*. A recurrent U-Net [28], trained with the windowed version of our loss.

In the *U-Net* experiments, we used the standard U-Net [18] architecture, with five blocks, each with three sequences of convolution-ReLU-batch normalization. Max-pooling in 2×2 windows followed each of the blocks. The initial feature size was set to 32 and grew to 1024 in the smallest feature map in the network. We augmented the input data with vertical and horizontal flips and random rotations.

In the experiments with *DRU*, we used a recurrent U-Net with the same architectural features that we used in *U-Net* experiments. There is a dual-gated recurrent unit in the bridge part of the network [28]. During training, we used three recurrent iterations. After each recurrent iteration, the output of the network is used as an additional channel to the input of the next iteration. For the first iteration, this additional channel is set to 0. During inference, we used the output of the second iteration which produced the best results.

We trained the network with the ADAM algorithm [29], with the learning rate set to $1e-4$. We set the mixing coefficients $\alpha = 1e-4$ in Eq. (2) and $\beta = 0.1$ in Eq. (4), empirically.

4.4 Performance measures

Comparing connectivity of road reconstructions is difficult, because the reconstructions rarely overlap with the ground truth, and often deviate from it significantly. There seems to be no consensus concerning the single best evaluation technique in the existing literature – we have found five different connectivity-oriented metrics in concurrently published recent work. To provide exhaustive evaluation, we used all of them in our experiments.

- *APLS* Average Path Length Similarity, defined as a aggregation of relative length difference of shortest paths

between pairs of corresponding points in the reconstructed and predicted maps [30].

- *TLTS* Statistics of lengths of shortest paths between corresponding pairs of end points randomly selected in the predicted and ground-truth networks [31]. We report the fraction of paths where the relative length difference is within 5%.
- *JCT* A junction score, evaluating the number of roads intersecting at each junction [8]. Consists of road recall, averaged over the intersections of the ground-truth and road precision, averaged over the intersections of the prediction. We report the corresponding F1 score.
- *HM* Compares the sets of graph locations accessible by traveling away from randomly chosen pairs of corresponding points in both graphs [32]. We report the corresponding F1-score.
- *CCQ* To complement the connectivity-oriented evaluation, we also computed the most popular metric that measures spatial co-occurrence of annotated and predicted road pixels, rather than connectivity. The Correctness, Completeness and Quality are equivalent to precision, recall and intersection-over-union, where the definition of a true positive has been relaxed from spatial coincidence of prediction and annotation to co-occurrence within a distance of 5 pixels [33]. We report the Quality as our single-number metric.

TABLE 1: Results of experiments on the *RoadTracer* dataset [8]. Our loss function makes even the simple *U-Net* attain state of the art performance. Computing the loss in windows results in improvement of four out of five performance criteria.

	Connectivity-oriented				pixel-based
Method	*APLS*	*TLTS*	*JCT*	*HM*	*CCQ*
Segmentation [8]	62.5	33.0	78.2	69.4	54.4
RoadTracer [8]	59.1	40.6	81.2	70.5	47.8
Seg-Path [11]	68.1	46.5	75.4	67.6	54.0
RCNNU-Net [12]	48.2	18.4	75.9	68.8	62.8
DeepRoad [6]	24.6	6.4	51.4	46.8	43.6
PolyMapper [7]	61.3	31.5	80.0	53.7	35.7
U-Net [18]	66.3	40.0	77.5	68.2	59.3
U-Net+TOPO-global	72.5	46.3	**84.7**	70.3	63.8
U-Net+TOPO-windowed	**75.8**	**49.7**	82.8	**76.0**	**68.6**

TABLE 2: Results of experiments on the *DeepGlobe* dataset. Our loss function improves the performance of both *U-Net* and *DRU* in terms of all the metrics, with *DRU* attaining the state-of-the-art performance.

	Connectivity-oriented				pixel-based
Method	*APLS*	*TLTS*	*JCT*	*HM*	*CCQ*
LinkNet [9]	67.7	60.6	66.2	73.4	77.2
MultiBranch [9]	70.8	65.2	71.1	75.6	79.4
U-Net [18]	62.3	59.9	66.4	72.7	68.8
DRU [28]	75.2	65.4	67.2	76.6	80.1
U-Net+TOPO-windowed	75.2	**69.8**	**71.2**	**79.8**	77.0
DRU + TOPO-windowed	**77.1**	68.4	**71.2**	79.6	**80.7**

TABLE 3: Results of experiments on the *Canals* dataset. Our loss function boosts the performance of both *U-Net* and *DRU* in terms of all the five metrics.

	Connectivity-oriented				pixel-based
Method	*APLS*	*TLTS*	*JCT*	*HM*	*CCQ*
U-Net [18]	70.9	30.3	76.4	61.4	81.4
DRU [28]	71.4	32.7	76.9	62.1	80.3
U-Net+TOPO-windowed	76.3	35.7	**79.3**	63.4	**85.1**
DRU + TOPO-windowed	**78.2**	**43.1**	78.8	**67.0**	84.5

4.5 Comparative evaluation

We report the performance of our method on the *RoadTracer*, *DeepGlobe* and *Canals* datasets in Tabs. 1, 2 and 3. These corresponding network reconstructions are depicted qualitatively in Figs. 3, 4 and 5. On average, *DeepGlobe* features simpler roads with fewer opportunities for mistakes than *RoadTracer* and *Canals*. Yet, in all datasets, we can capture connectivity more reliably than competing methods.

Using the windowed version of our loss function boosts the performance of a simple U-Net past that of *all* the baselines on *all measures*, except CCQ on the *DeepGlobe* dataset. The last is not surprising because our loss is designed to enforce connectivity, which CCQ does not measure. What is remarkable is that we were able to achieve this result using the comparatively simple U-Net architecture, whereas many of the competing architectures are far more sophisticated. When the network is more powerful, our loss function boosts its performance even further. As can be seen in Tab. 2, *DRU* yields results as good as *MultiBranch*, the best performer on the *DeepGlobe* dataset, already when trained with the mean squared error. Training *DRU* with our loss increases its performance in terms of *all the scores*. Thanks to its increased stability, the windowed version of the loss slightly outperforms the global one. The one exception is the *JCT* measure computed on the *RoadTracer* dataset, where the global version performs better than the windowed one. We attribute this to the slightly increased tendency of the network to create road bifurcations when using the windowed-loss, which has little effect on the other metrics.

4.6 Ablation Studies

We run a number of ablation studies to investigate the impact of the hyper-parameters of our method on performance.

4.6.1 *Varying the impact of the connectivity loss*

As defined in Eq. (2), our loss function is a combination of the mean square error with the connectivity term L_{TOPO}. The influence of the connectivity term on the loss is controlled by coefficient α. We varied α to investigate its impact on the distance maps produced by the network. The results presented in Tab. 4 show that setting this coefficient too-low or too-high adversely affected performance, and its optimal value is in the order of $1e-4$. The explanation of this phenomenon is provided in Fig. 6. For low values of α, the effect of the connectivity-oriented component of the loss function is negligible. When α is increased, more and more connections are represented in the distance map. However, when α is set very high, the network starts to privilege disconnecting background image regions, even with no obvious roads in the input, creating false positive road segments.

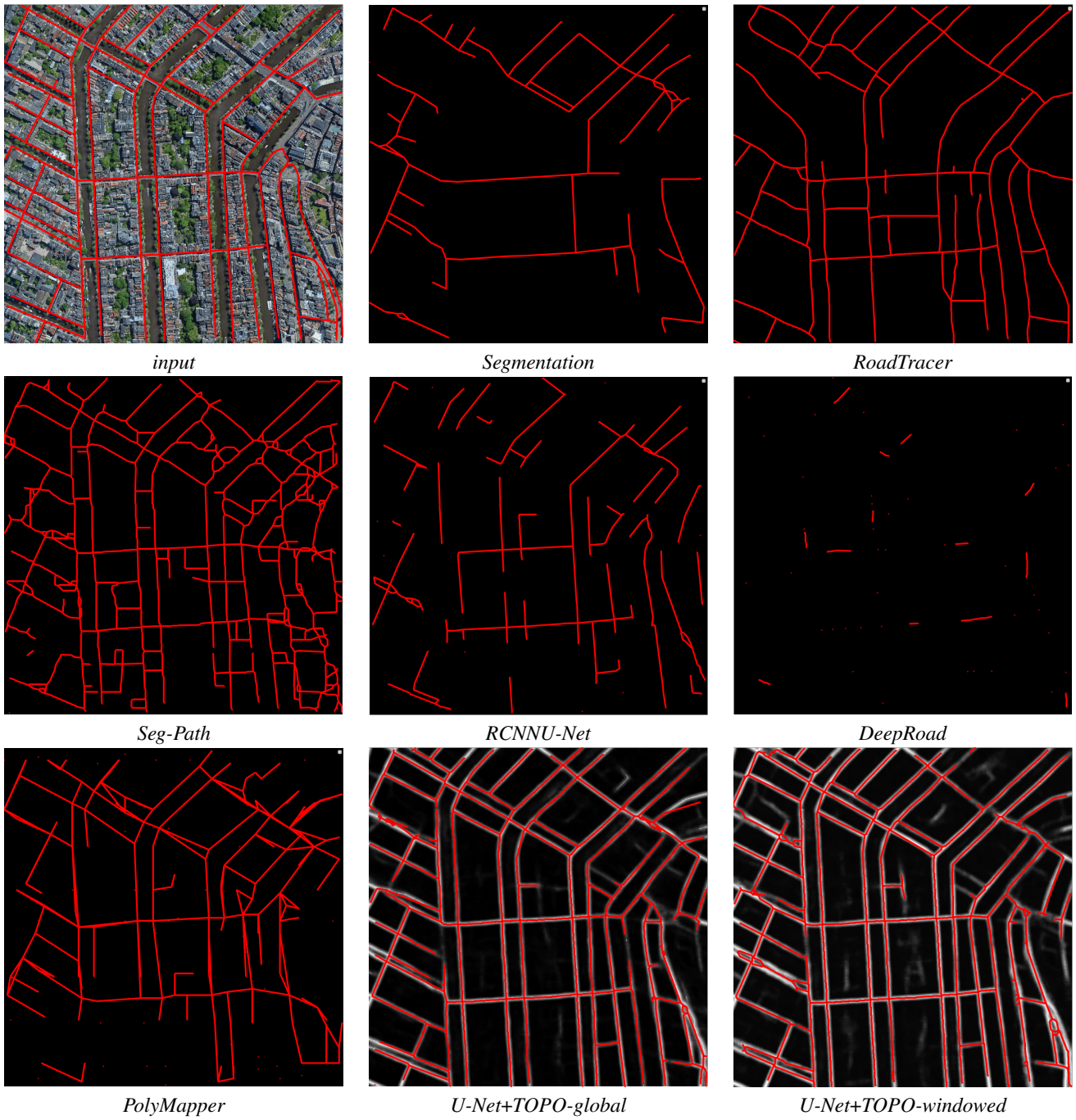


Fig. 3: Comparative results on the *RoadTracer* dataset. For our results, we overlaid the graphs on the inferred distance maps.

4.6.2 Balancing connectivity versus dis-connectivity

We counteract the tendency to produce excessively connected road networks by incorporating into the connectivity term a component penalizing false positive road segments, in addition to the one encouraging connectivity of true roads. As specified in Eq. (7), the coefficient β balances these two terms. We varied β to investigate its impact on performance. We present the results in Tab. 5. According to all the performance measures, the best results are obtained for $\beta = 0.1$, meaning that the term preventing disconnections should have ten times more impact on the loss than the term preventing false positive roads.

4.6.3 Varying the window size

The third, and last, hyper-parameter of our method is the window size. Computing the loss in windows, or image crops, as opposed to globally in the entire image, has the advantage of preventing accumulating all the error signal in a single pixel. The smaller the window, the more evenly the gradient is distributed among road pixels. The windowed version of the loss also enables enforcing connectivity of dead-ending roads, as small windows are often subdivided even by dead-ending roads. Large window sizes do not have this effect, as roads shorter than the window size end in the middle of the window, without splitting it into disjoint tiles. To discover the optimal window size, we tested its effect on performance. The results, presented in Tab 6 and 7, confirm

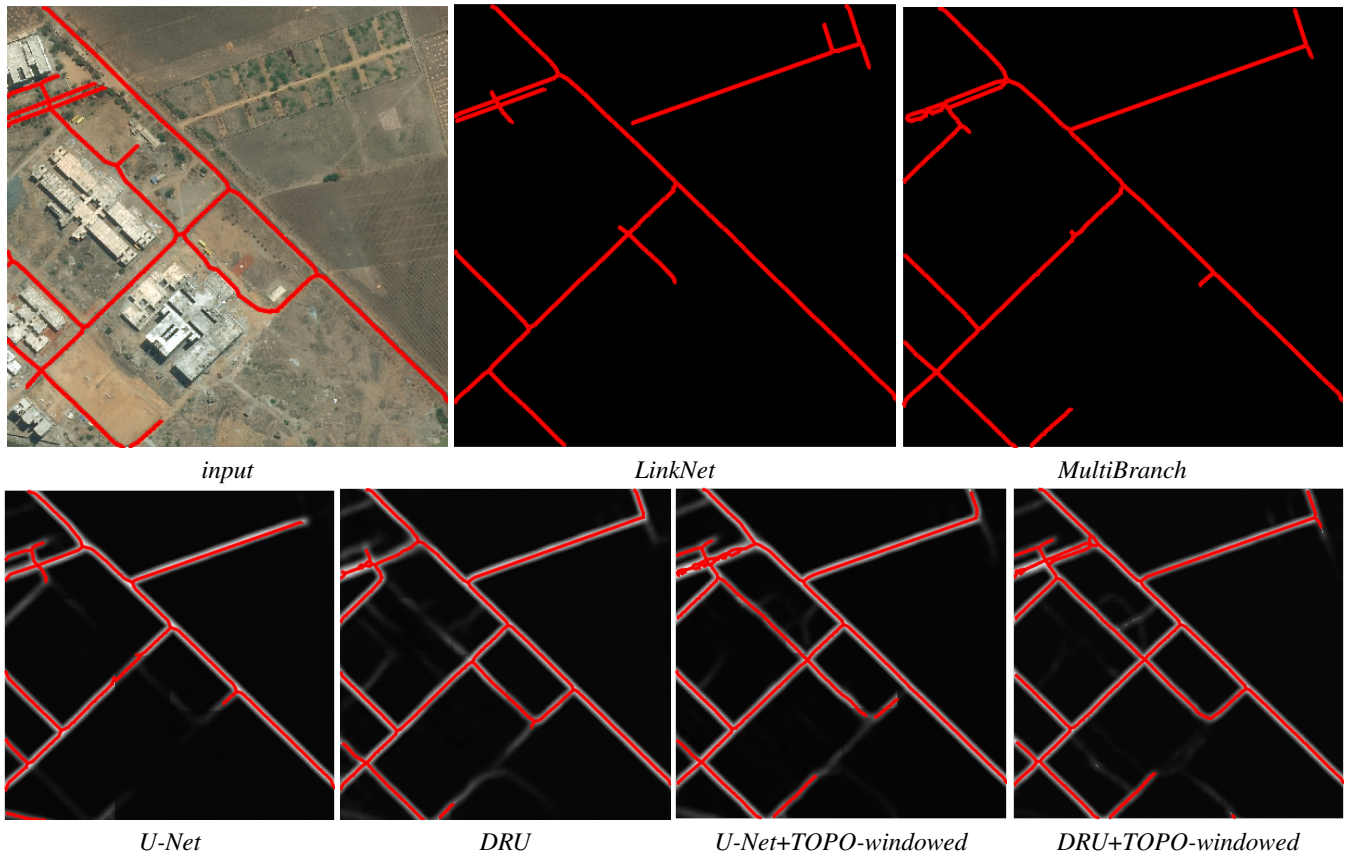


Fig. 4: Comparative results on the *DeepGlobe* dataset. For the results of our method, we overlaid graphs on the inferred distance maps.

TABLE 4: The impact of changing the α coefficient, balancing L_{MSE} and L_{TOPO}, on performance. Results of experiments on the *RoadTracer* dataset. Window size is fixed to 64x64 and β to 0.1. *U-Net+TOPO-windowed* is used in all experiments. Visualization of the corresponding results can be found in Fig. 6.

	Connectivity-oriented				pixel-based
α	*APLS*	*TLTS*	*JCT*	*HM*	*CCQ*
1e-3	72.3	46.3	80.3	73.1	66.5
1e-4	**75.8**	**49.7**	**82.8**	**76.0**	**68.6**
1e-5	71.4	45.9	81.9	73.4	67.1
0.0	66.3	40.0	77.5	68.2	59.3

TABLE 5: The impact of β, balancing the connectivity and disconnectivity components of our loss, on performance. Results of experiments on the *RoadTracer* dataset. *U-Net+TOPO-windowed* is used in all experiments.

	Connectivity-oriented				pixel-based
β	*APLS*	*TLTS*	*JCT*	*HM*	*CCQ*
1e-0	71.4	43.8	80.9	73.2	66.5
1e-1	**75.8**	**49.7**	**82.8**	**76.0**	**68.6**
1e-2	74.3	46.2	79.5	74.5	65.3

that mid-size windows work best. Setting the window size to 64×64 pixels resulted in the highest performance, and increasing or decreasing the window decreases performance.

TABLE 6: The impact of window size on performance. Results of experiments on the *RoadTracer* dataset. α is fixed to $1e-4$ and β to 0.1. *U-Net+TOPO-windowed* is used in all experiments.

	Connectivity-oriented				pixel-based
Window Size	*APLS*	*TLTS*	*JCT*	*HM*	*CCQ*
(16x16)	68.3	39.2	79.2	67.4	59.4
(32x32)	72.1	45.8	78.9	72.7	65.7
(64x64)	75.8	**49.7**	**82.8**	**76.0**	**68.6**
(128x128)	**76.1**	46.4	81.7	74.5	68.3

TABLE 7: The impact of changing the window size on performance. Results of experiments on the *DeepGlobe* dataset. α is fixed to $1e-4$ and β to 0.1. *U-Net+TOPO-windowed* is used in all experiments.

	Connectivity-oriented				pixel-based
Window Size	*APLS*	*TLTS*	*JCT*	*HM*	*CCQ*
(32x32)	74.1	67.3	65.2	73.4	74.8
(64x64)	**75.2**	**69.8**	71.2	**79.8**	77.0
(128x128)	74.3	68.2	**72.0**	79.6	**77.2**

4.6.4 Comparing Mean Squared Error to Cross Entropy

Our loss function combines a connectivity-oriented term with mean squared error. This combination outperforms a number of existing networks, trained with cross entropy. We therefore investigated if just switching from the more common cross entropy to mean squared error, without our connectivity-oriented loss,

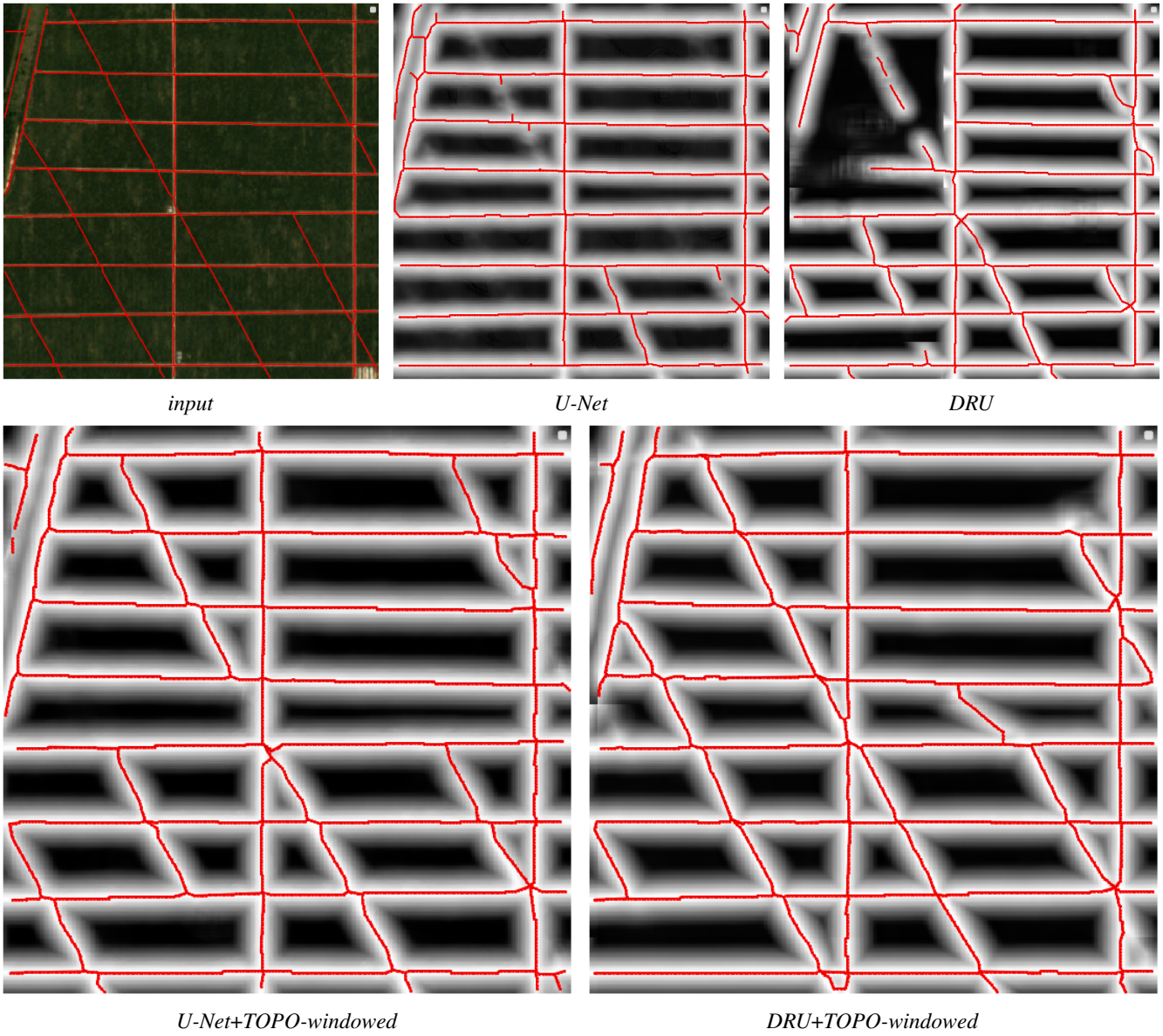


Fig. 5: Comparative results on the *Canals* dataset. For the results of our method, we overlaid graphs on the inferred distance maps.

TABLE 8: Comparison of Cross Entropy and Mean Square Error. Results of experiments on the *RoadTracer* dataset [8].

	Connectivity-oriented				pixel-based
Method	*APLS*	*TLTS*	*JCT*	*HM*	*CCQ*
U-Net-CE [18]	60.4	30.6	79.2	74.2	63.3
U-Net-MSE [18]	66.3	40.0	77.5	68.2	59.3
U-Net+TOPO-global	72.5	46.3	**84.7**	70.3	63.8
U-Net+TOPO-windowed	**75.8**	**49.7**	82.8	**76.0**	**68.6**

impacts the performance. We present the results in Tab. 8. We conclude that solely switching from pixel classification to distance map estimation does not warrant the increased connectivity, and its the addition of our connectivity-oriented term that does it.

5 Conclusion and Future Work

We have introduced a differentiable loss function that effectively enforces proper connectivity on the output of binary segmentation ConvNets for the purpose of road network delineation. Using this loss function to train a simple U-Net allows us to outperform far more sophisticated architectures on challenging benchmark datasets. This suggests that we may not yet have unleashed the full power of these simpler networks and that adding appropriate constraints during training might be a way to do so.

We have so far limited ourselves to networks of roads and drainage canals, but networks of linear structures are also pervasive in biomedical 3D imagery. They range from neural structures to blood vessels and many others. In future work, we will therefore expand our approach to handle 3D image stacks and address a much broader range of applications.

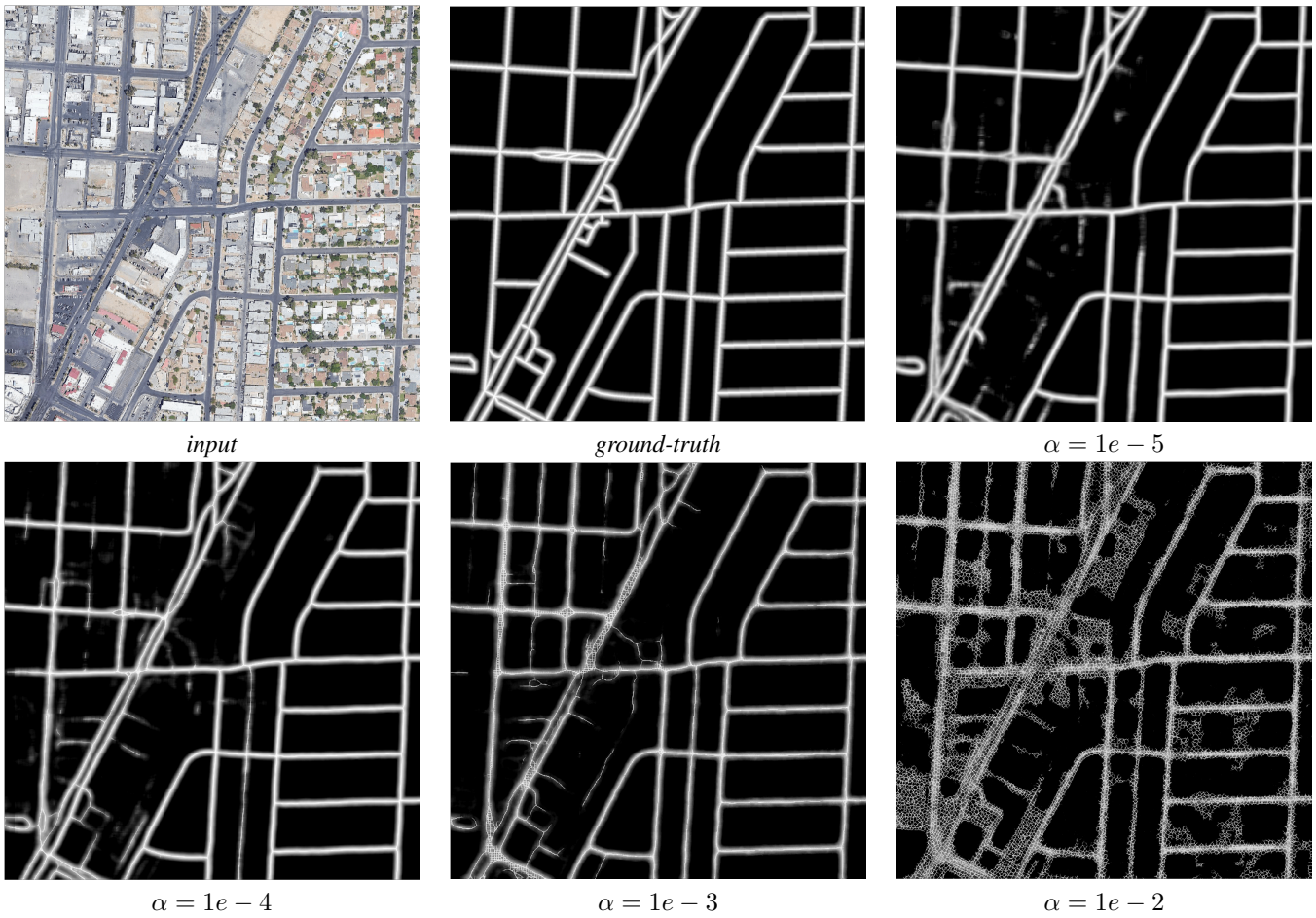


Fig. 6: Effect of α on the distance map output by the neural network. As α is increased, the road map becomes more complete. However, high values of α promote creating erroneous connections even where no roads are present in the image. The corresponding numerical results can be found in Tab. 4.

REFERENCES

[1] R. Bajcsy and M. Tavakoli, "Computer Recognition of Roads from Satellite Pictures," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. SMC-6, no. 9, pp. 623–637, 1976.

[2] G. Vanderbrug, "Line Detection in Satellite Imagery," *IEEE Transactions on Geoscience Electronics*, vol. 14, no. 1, pp. 37–44, January 1976.

[3] L. Quam, "Road Tracking and Anomaly Detection," in *DARPA Image Understanding Workshop*, May 1978, pp. 51–55.

[4] M. Fischler, J. Tenenbaum, and H. Wolf, "Detection of Roads and Linear Structures in Low-Resolution Aerial Imagery Using a Multisource Knowledge Integration Technique," *Computer Vision, Graphics, and Image Processing*, vol. 15, no. 3, pp. 201–223, March 1981.

[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 Trans. Geoscience and Remote Sensing*, vol. 55, no. 6, pp. 3322–3337, 2017.

[6] G. Máttyus, W. Luo, and R. Urtasun, "Deeproadmapper: Extracting Road Topology from Aerial Images," in *International Conference on Computer Vision*, 2017, pp. 3458–3466.

[7] Z. Li, J. Wegner, and A. Lucchi, "Topological Map Extraction from Overhead Images," in *International Conference on Computer Vision*, 2019.

[8] 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 *Conference on Computer Vision and Pattern Recognition*, 2018.

[9] A. Batra, S. Singh, G. Pang, S. Basu, C. Jawahar, and M. Paluri, "Improved Road Connectivity by Joint Learning of Orientation and Segmentation," in *Conference on Computer Vision and Pattern Recognition*, June 2019.

[10] H. Chu, D. Li, D. Acuna, A. Kar, M. Shugrina, X. Wei, M. Liu, A. Torralba, and S. Fidler, "Neural Turtle Graphics for Modeling City Road Layouts," in *International Conference on Computer Vision*, 2019.

[11] A. Mosińska, M. Kozinski, and P. Fua, "Joint Segmentation and Path Classification of Curvilinear Structures," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 6, pp. 1515–1521, 2020.

[12] X. Yang, X. Li, Y. Ye, R. Y. K. Lau, X. Zhang, and X. Huang, "Road Detection and Centerline Extraction via Deep Recurrent Convolutional Neural Network U-Net," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–12, 2019.

[13] D. Murdiyarso, K. Hergoualc'h, and L. V. Verchot, "Opportunities for reducing greenhouse gas emissions in tropical peatlands," *Proceedings of the National Academy of Sciences*, vol. 107, no. 46, pp. 19 655–19 660, 2010.

[14] J. Leifeld, C. Wüst-Galley, and S. Page, "Intact and managed peatland soils as a source and sink of ghgs from 1850 to 2100," *Nature Climate Change*, vol. 9, no. 12, pp. 945–947, 2019.

[15] A. Mosińska, P. Marquez-Neila, M. Kozinski, and P. Fua, "Beyond the Pixel-Wise Loss for Topology-Aware Delineation," in *Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3136–3145.

[16] K. Briggman, W. Denk, S. Seung, M. Helmstaedter, and S. Turaga, "Maximin Affinity Learning of Image Segmentation," in *Advances in Neural Information Processing Systems*, 2009, pp. 1865–1873.

[17] J. Funke, F. D. Tschopp, W. Grisaitis, A. Sheridan, C. Singh, S. Saalfeld, and S. C. Turaga, "Large Scale Image Segmentation with Structured Loss Based Deep Learning for Connectome Reconstruction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 41, no. 7, pp. 1669–1680, 2018.

[18] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Conference on Medical Image Computing and Computer Assisted Intervention*, 2015, pp. 234–241.

[19] R. Vernimmen, A. Hooijer, D. Mulyadi, I. Setiawan, M. Pronk, and A. T. Yuherdha, "A new method for rapid measurement of canal water table depth using airborne lidar, with application to drained peatlands in indonesia," *Water*, vol. 12, no. 5, p. 1486, 2020.

[20] Y. Ishii, K. Koizumi, H. Fukami, K. Yamamoto, H. Takahashi, S. Limin, K. Kusin, A. Usup, and G. Susilo, "Groundwater in peatland," in *Tropical Peatland Ecosystems*, 01 2016, pp. 265–279.

[21] J. Clough, I. Oksuz, N. Byrne, J. Schnabel, and A. King, "Explicit Topological Priors for Deep-Learning Based Image Segmentation Using Persistent Homology," in *Information Processing in Medical Imaging*, 2019.

[22] X. Hu, F. Li, D. Samaras, and C. Chen, "Topology-Preserving Deep Image Segmentation," *CoRR*, vol. abs/1906.05404, 2019.

[23] A. Sironi, V. Lepetit, and P. Fua, "Multiscale Centerline Detection by Learning a Scale-Space Distance Transform," in *Conference on Computer Vision and Pattern Recognition*, 2014.

[24] 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," in *Conference on Computer Vision and Pattern Recognition*, June 2018.

[25] P. Team, "Planet application program interface: In space for life on earth," https://api.planet.com, 2017. [Online]. Available: https://api.planet.com

[26] F. Bastani, "Roadtracer web page," https://mapster.csail.mit.edu/roadtracer.html, 2018. [Online]. Available: https://mapster.csail.mit.edu/roadtracer.html

[27] A. Chaurasia and E. Culurciello, "Linknet: Exploiting Encoder Representations for Efficient Semantic Segmentation," *CoRR*, vol. abs/1707.03718, 2017.

[28] W. Wang, K. Yu, J. Hugonot, P. Fua, and M. Salzmann, "Recurrent U-Net for Resource-Constrained Segmentation," in *International Conference on Computer Vision*, 2019.

[29] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimisation," in *International Conference on Learning Representations*, 2015.

[30] A. V. Etten, "Spacenet Road Detection and Routing Challenge Part II — APLS Implementation."

[31] J. Wegner, J. Montoya-Zegarra, and K. Schindler, "A Higher-Order CRF Model for Road Network Extraction," in *Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1698–1705.

[32] J. Biagioni and J. Eriksson, "Inferring Road Maps from Global Positioning System Traces: Survey and Comparative Evaluation," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2291, pp. 61–71, 12 2012.

[33] C. Wiedemann, C. Heipke, H. Mayer, and O. Jamet, "Empirical Evaluation of Automatically Extracted Road Axes," in *Empirical Evaluation Techniques in Computer Vision*, 1998, pp. 172–187.