Beyond the Pixel-Wise Loss for Topology-Aware Delineation

Tong Zhao and Xiaoqi Xu

May 4, 2019

Contents

1 Introduction

1.1 Problem

The paper "Beyond the Pixel-Wise Loss for Topology-Aware Delineation"[3] tries to exploit recent techniques developed in the field of deep learning to improve the current performance for the problem of delineation of curvilinear structures, i.e. extract linear structures automatically from numerical images.

The delineation problem is an old problem that has been studied since half a century. Former algorithms rely essentially on clever designed features. Therefore they don't need training data but suffer from the problem of poor adaptivity to different applications.

Recent development of deep learning has drawn attention of a wide range of scientific communities to use this cutting-edge technique to solve their problems. There have been some successful examples of applying neuron networks, e.g. VGG and U-Net, to extract blood vessels and do image segmentation.

1.2 Method

The authors of this paper want to remedy the current pipelines by using deep learning techniques to detect curvilinear structures. The former methods focus mostly on finding better features or network architectures while still using a standard cross entropy loss which treats pixels locally, thus it often leads to bad topological structures.

However it leads often to complex pipelines if encoding topology knowledge explicitly. So the authors proposed a simple topology-aware loss term to penalize the topological errors. This term is based on the differences between feature maps of low-level feature maps from a pretrained convolutional neural network which are sensitive to linear structures of the ground truth and the delineation predicted by networks. Then they use a combination of this topological loss and the traditional pixel-wise binary cross-entropy loss to optimize the neural network.

Besides, they have also employed an iterative refinement procedure which improves the quality of the output without adding extra parameters to be optimized, inspired by the recurrent convolutional architecture of Pinheiro and Collobert [4].

1.3 Relation with the course

In the course, we have seen geodesic methods to extract curvilinear structures. More precisely, we consider the grey level of the image as a potential function and by finding a path of minimal energy, we extract the desired curvilinear structure.

The method proposed in this paper solves this problem in another way. By learning parameters from a huge amount of data, it is more robust and less sensitive to noise compared with geodesic methods since it considers the global topology of the curvilinear structure. It also depends less on exterior information and is more adaptive to different situations. For geodesic methods, it is often hard to determine whether the segment is part of the curvilinear structure or the segment is in fact a false segment. Thus we should carefully choose parameters according to different situation in order to get the desired result.

Nevertheless, geodesic method is solved by the fast marching algorithm very efficiently and requires no priors.

1.4 Contribution

The main original contribution of this paper is the introduction of the topology-aware term in the loss function which boosts performance of the network. It is computed from the difference between the VGG descriptors of the ground truth and the ones of predicted delineations.

Another contribution is the use of iterative refinement procedure which improves the prediction quality without introducing more parameters. The improvement can be observed clearly in the following experiments.

1.5 Results

As mentioned above, this paper exploits the feature maps specialized in curvilinear structure of VGG and thus incorporates topology information of the global curvilinear structure. The quantitative results demonstrate that it beats the state of the art under the traditional statistical measures like correctness, completeness and quality, and show a significant improvement under topological measures compared to several recent competing methods. This shows the success of the neural network structure proposed in this paper.

1.6 Drawback

Although this method is very powerful, it is also very computational expensive. Besides, the topology-aware loss uses the l^2 norm to measure the difference between the VGG descriptors of the ground truth and the prediction, which diminishes the topological flavor of this term. Instead, we could try to use l^1 norm or other more sophisticated topology costs.

Similarly as all deep learning methods, the algorithm needs a well-labeled database which is not always available.

2 Summary of the article

2.1 Topology-aware loss

Let $\mathbf{x} \in \mathbb{R}^{H \cdot W \cdot 3}$ be an input color image of size $H \times W \times 3$, and $\mathbf{y} \in \{0,1\}^{H \cdot W}$ be the corresponding ground truth labeling. We denote the network by f and its weights by $\mathbf{w}$. The output of the network $\hat{\mathbf{y}} = f(\mathbf{x}, \mathbf{w}) \in [0,1]^{H \cdot W}$ is an image of same size indicating the probability of each pixel belonging to curvilinear structure.

The traditionally used loss function to train the network is the standard pixel-wise binary cross-entropy (BCE) which is defined as follows:

$$\mathcal{L}_{bce}(\mathbf{x}, \mathbf{y}, \mathbf{w}) = -\sum_i [(1 - \mathbf{y}_i) \cdot \log(1 - f_i(\mathbf{x}, \mathbf{w})) + \mathbf{y}_i \cdot \log f_i(\mathbf{x}, \mathbf{w})] \quad (1)$$

The BCE loss is quite natural and simple, but cannot capture the thin curvilinear structures that have a big topological impact to global prediction while influence slightly the pixel-wise loss.

In order to take account of higher order information, the authors use feature maps of several layers of VGG19 network pretrained on ImageNet as the descriptors of higher order information. This has the advantage of good adaptability compared to any other hand-designed features. More precisely, they defined the topology-aware loss term as the normalization of the squared l^2-distance between the VGG19 features of the ground truth and the ones of the predictions:

$$\mathcal{L}_{top}(\mathbf{x}, \mathbf{y}, \mathbf{w}) = \sum_{n=1}^{N} \frac{1}{M_n W_n H_n} \sum_{m=1}^{M_n} \| l_n^m(\mathbf{y}) - l_n^m(f(\mathbf{x}, \mathbf{w})) \|_2^2 \quad (2)$$

where l_m^n is the m-th feature map of the n-th layer of pretrained VGG19 network, N is the number of layers used and M_n is the number of channels in the n-th layer. The main reason that this term boosts performance is that certain channels of the VGG19 layers are sentitive to linear structures and certain channels are specialized in small connected components: both help to extract the right linear structure that we are interested in.

Finally, the loss function used to train the network is a combination of BCE loss and topology-aware loss with parameter μ balancing the magnitude of the two terms:

$$\mathcal{L}(\mathbf{x}, \mathbf{y}, \mathbf{w}) = \mathcal{L}_{bce}(\mathbf{x}, \mathbf{y}, \mathbf{w}) + \mu \mathcal{L}_{top}(\mathbf{x}, \mathbf{y}, \mathbf{w}) \quad (3)$$

2.2 Iterative refinement

The iterative refinement procedure is based on the following property: the correct delineation $\mathbf{y}$ should be a fixed point of each module f^k, i.e. $\mathbf{y} =$

$f^k(\mathbf{x} \oplus \mathbf{y})$, where $\oplus$ denotes channel concatenation. If the map is contract, the result of iteration procedure will eventually converge to $\mathbf{y}$. At each iteration, we concatenate the prediction of last iteration to the input image and produce a new prediction. Thus we do not increase the number of parameters to learn.

In order to take account of earlier errors, the authors used a weighted sum of partial losses:

$$\mathcal{L}_{ref}(\mathbf{x}, \mathbf{y}, \mathbf{w}) = \frac{1}{Z} \sum_{k=1}^{K} k\mathcal{L}^k(\mathbf{x}, \mathbf{y}, \mathbf{w}) \tag{4}$$

where $Z = \sum_{k=1}^{K} k = \frac{1}{2}K(K+1)$ is the normalization factor.

3 Experiments

In our experiments, we used the Massachusetts Roads Dataset[2] for its rich training data. The training dataset, validation dataset and test dataset contain 1108 images, 14 images and 49 images, respectively. All training images are resized to 256 × 256 and are then randomly cropped to 224 × 224 in order to fit the input size of VGG net. All test images are resized directly to 224 × 224.

We used the pretrained VGG19 (with batch normalization) model and its layers: relu(conv1_2), relu(conv2_2), relu(conv3_4) and relu(conv4_4) are extracted to generate topological features. The main neural network is an UNet-based auto-encoder which contains 4 upsampling blocks and 4 upsampling blocks with batch normalization and prelu activation [1]. We also used Adam as optimizer as the authors did, with a learning rate of 10^{-3}. Batch size is set to 8 due to the device limit. μ is set to 0.1 in the training loss function in Eq. 3. In the iterative refinement step, we have iterated 3 steps as the paper suggested that the results do not improve significantly after $K = 3$.

The pipeline is the same as in the paper (see Fig. 1): first we train an U-Net based model with the loss $\mathcal{L}_{top}$ computed from the responses of VGG and the loss $\mathcal{L}_{bce}$ computed from the prediction and the ground truth; then we iterately finetune the same U-Net to refine the prediction with a weighted sum of partial losses $\mathcal{L}^k$ computed at k-th iteration as loss function.

3.1 Quantitative results

To compare our results with the results in the paper, we have evaluated our results in terms of statistical criterion: correctness, completeness and quality defined in [5].

We take two images as example and show their predictions using 3 networks: the U-Net with BCE loss (U-Net), the U-Net with topological loss of Eq. 3 but

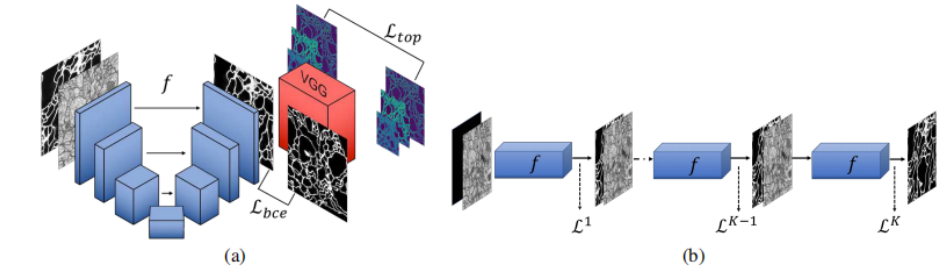

Figure 1: Training pipeline

without refinement ($K = 1$) and the U-Net with topological loss and refinement ($K = 3$). Each network is trained for 100 epochs.

The results are in Table 1 for the first image and Table 2 for the second image. Even though our results are not as good as reported in the paper (especially the completeness), but we still see an improvement in the total quality by successively applying topological loss and iterative refinement into the original U-Net architecture.

	Completeness	Correctness	Quality
U-Net	0.616765	0.713445	0.494312
K=1	0.510445	0.765462	0.441408
K=3	0.566685	0.789847	0.492437

Table 1: Quantitative results for image 20728960

	Completeness	Correctness	Quality
U-Net	0.743153	0.426327	0.371577
K=1	0.602517	0.500307	0.376155
K=3	0.678016	0.516638	0.414855

Table 2: Quantitative results for image 10378780

We have another interesting observation that iterative refinement in test phase decreases the quality of results in all aspects for networks that are not iteratively refined as shown in Table 3. The deterioration is most serious after first iteration, then the predictions get better. Perhaps they just need more iterations to converge to the best results than fine-tuned networks.

	Completeness	Correctness	Quality
iter1	0.743153	0.426327	0.371577
iter2	0.220577	0.207232	0.119631
iter3	0.397483	0.249419	0.180991

Table 3: Iterative refinement deteriorates the prediction results for U-Net without refinement

We then report the results on the whole test dataset, which contains 49 images.

	Completeness	Correctness	Quality
U-Net	0.67584426	0.59423022	0.43436453
K=1	0.58282891	0.7042716	0.44924965
K=3	0.61157189	0.70223687	0.46712003
Paper	0.8057	0.7743	0.6524

Table 4: Quantitative results on test dataset

The poor results may due to the device limit which cannot support a long training procedure, or due to the reduced image size which is limited by the GPU memory. The hyper-parameters are not optimized as well.

3.2 Qualitative results

As shown in Fig. 2, the use of topological loss in training networks helps to eliminate false little segments in predictions. Fig. 3 shows that iterative refinement procedure fills small gaps gradually and makes the predictions much more clean and structured. As we have remarked in the previous section, qualitative results in Fig. 3 show more intuitively that iterative refinement in test phase for networks trained without refinement deteriorates seriously the curvilinear structures.

4 Conclusion

We have summarised the method proposed in paper [3], and compared it with the methods we have seen in the course. We have also implemented this method and analysed our results both quantitatively and qualitatively. If time permitted, we could experiment with a different network architecture than U-Net, for example with GAN, to see whether we could get better results. Or we could try to ameliorate the topological loss to make the computation faster yet still taking account of the global topological structures.

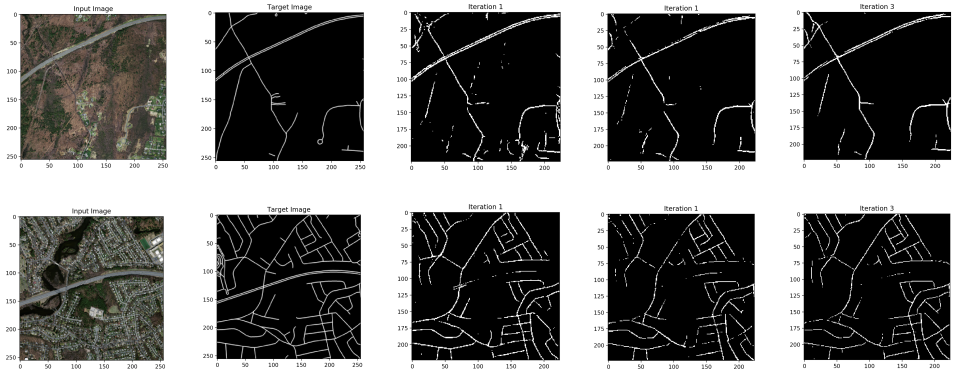

Figure 2: Qualitative result: from left to right, the original image, ground truth, U-Net without topological loss, U-Net with topological loss but without refinement, U-Net with topological loss and refinement

References

[1] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[2] V. Mnih. *Machine Learning for Aerial Image Labeling*. PhD thesis, University of Toronto, 2013.

[3] Agata Mosinska, Pablo Márquez-Neila, Mateusz Koziński, and Pascal Fua. Beyond the pixel-wise loss for topology-aware delineation. In *CVPR*, 2018.

[4] P. Pinheiro and R. Collobert. Recurrent neural networks for scenel labelling. In *International Conference on Machine Learning*, 2014.

[5] C. Wiedemann, C. Heipke, H. Mayer, and O. Jamet. Empirical evaluation of automatically extracted road axes. In *Empirical Evaluation Techniques in Computer Vision*, pages 172–187, 1998.

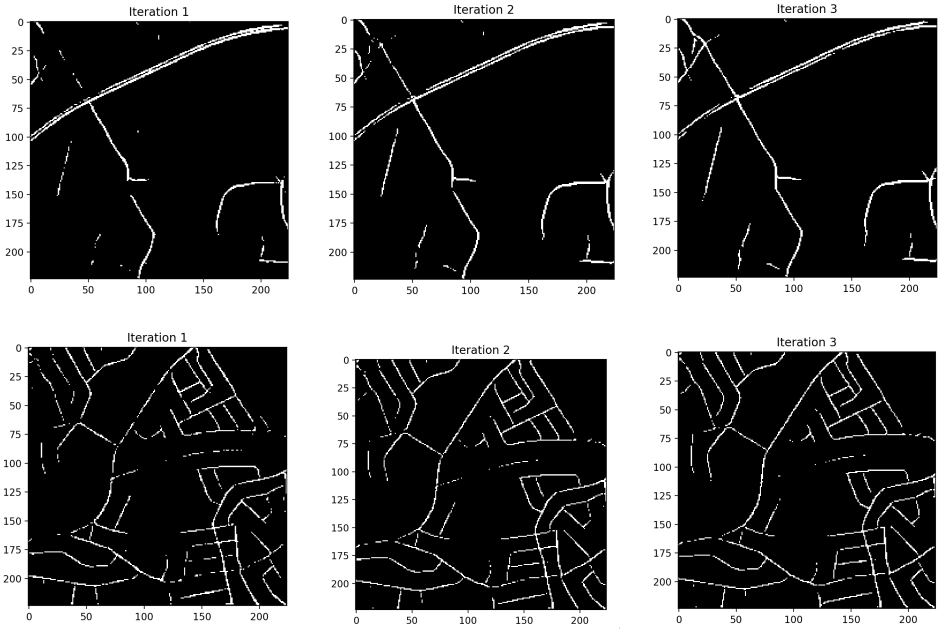

Figure 3: Iterative refinement fills small gaps

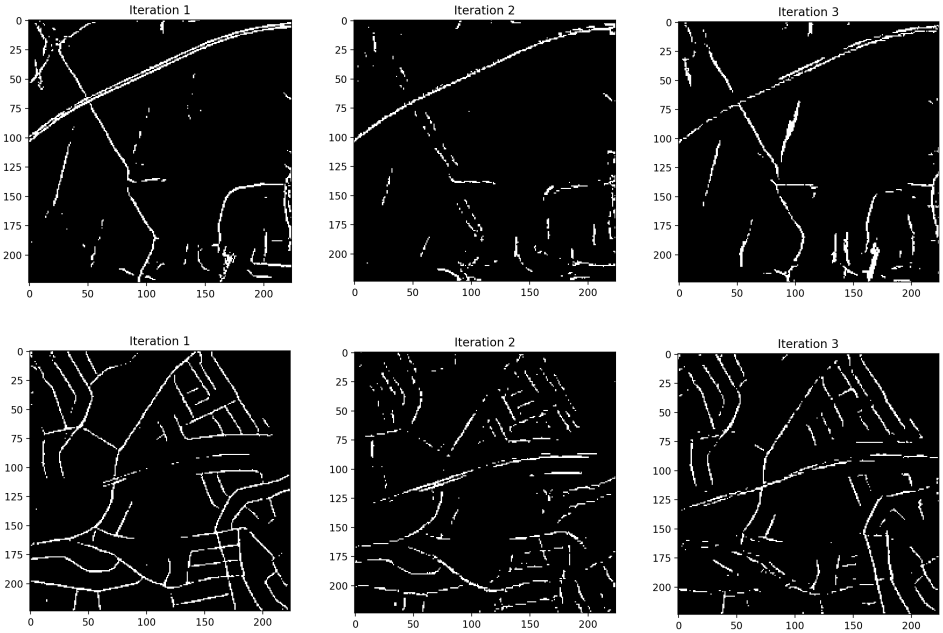

Figure 4: Iterative refinement deteriorates the prediction results for U-Net without refinement