Beyond the Pixel-Wise Loss for Topology-Aware Delineation

Tong Zhao & Xiaoqi Xu

Ecole des Ponts ParisTech & ENS Paris-Saclay

May 2, 2019

Outline

1. Introduction
2. Methods
   - Topology-aware loss
   - Iterative refinement
3. Experiments
   - Quantitative results
   - Qualitative results
4. Conclusions

We want to extract automatically the curvilinear structures from numerical pictures obtained from various sources.

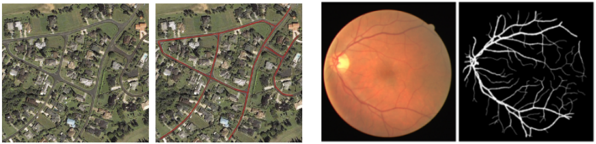


Figure: Delineation of curvilinear structures

Introduction

In recent works, Convolutional Neural Networks (CNN) have been successfully applied in this field. But most methods use the binary cross-entropy (BCE) as the loss function, which neglects global topological structures.

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

Motivation

Filters in convolutional layers are sensitive to curvilinear structures as shown in figure below.

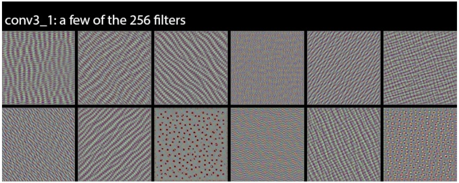


Figure: Some filters of VGG16

Motivation

Filters in convolutional layers are sensitive to curvilinear structures as shown in figure below.

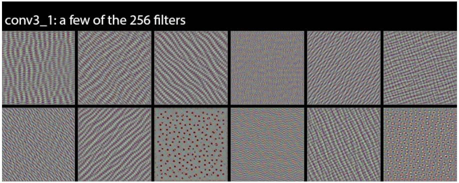


Figure: Some filters of VGG16

> Design a neural network which takes into account at the same time the details and global topology.

https://blog.keras.io/how-convolutional-neural-networks-see-the-world.html

Topology-aware loss

Inspired by the previous observation, the authors use the difference between the feature maps of ground truth and predictions of U-Net to define a loss term that takes account of global topological structures.

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

The final loss function is a combination of BCE loss and topological loss:

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

Iterative refinement

The iterative refinement procedure is based on the following property: the correct delineation $\mathbf{y}$ satisfies $\mathbf{y} = f^k(\mathbf{x} \oplus \mathbf{y})$, where $\oplus$ denotes channel concatenation.

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.

Pipeline

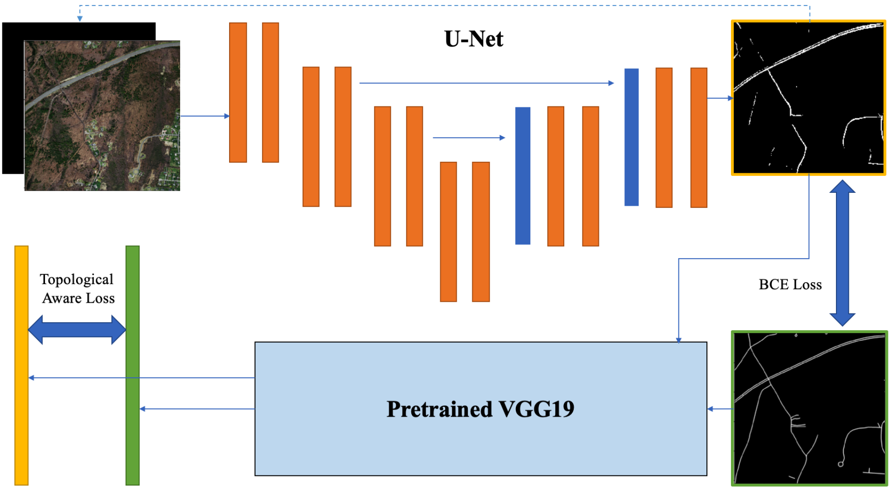


Figure: Pipeline

Phase

1. Intialization

In order to get a good initialization, we train from scratch a network with $K = 1$.

2. Finetune

To furthermore get a good model with iterative strategy, we finetune the network with $K = 3$ on the same dataset.

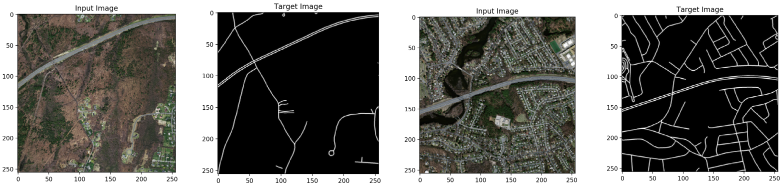
Parameters

Parameter	Value (K=1)	Value (K=3)
Learning Rate	0.01	0.001
Epoch	100	300
Optimizer	ADAM	
VGG	VGG19 with BN	
Batch size	8	
Batch Normalisation	True	
Activation	P-Relu with r = 0.2	
Dropout	False	
Regularisation Coeff	0.1	
Layer Index	conv1_2, conv2_2, conv3_4, conv4_4	

Experiments

Datasets: Massachusetts Roads Dataset.

Dataset	Training	Validation	Test
Number	1108	14	49



Criterion

- Completeness: the percentage of the groundtruth which is explained by the prediction

$$f_1(\text{label}, \text{pred}) = \frac{TP}{TP + FN}$$

- Correctness: the percentage of correctly extracted road data

$$f_2(\text{label}, \text{pred}) = \frac{TP}{TP + FP}$$

- Quality: a general measure of the final result combining completeness and correctness into a single measure

$$f_3(\text{label}, \text{pred}) = \frac{TP}{TP + FP + FN}$$

Quantitative results on testset

	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: Quantitative results on test dataset

Qualitative results

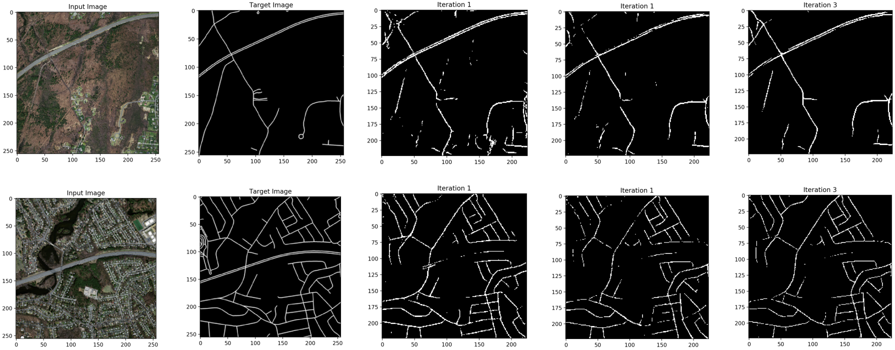


Figure: Qualitative result: from left to right, the original image, ground truth, U-Net without topological loss, U-Net with topological loss but without refinement, U-Net with topological loss and refinement

Qualitative results

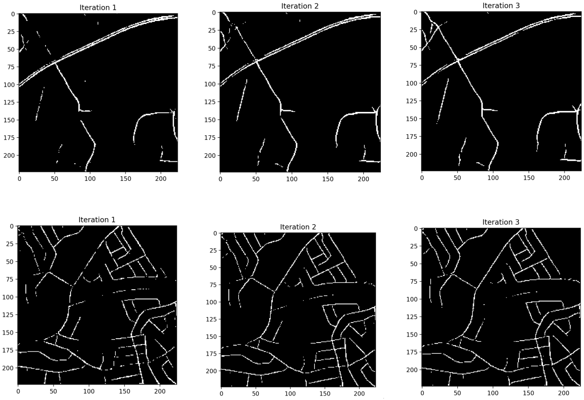


Figure: Iterative refinement fills small gaps

Conclusions

Pros:

- Improvement on both statistical and topological metrics
- Less topological mistakes
- Flexible framework

Cons:

- Computational expensive
- Need massive data
- No clear theoretical explanation or guarantee
- Sensitive to hyper-parameters

Future Work

- Do more experiments on a powerful device
- Try more structures, like GAN based model, etc.
- Design sophisticated loss function using the criterion above

Code available on github:
https://github.com/Tong-ZHAO/topology_aware_delineation

Thanks for your attention !

Questions ?