Note:Tree Energy Loss: Towards Sparsely Annotated Semantic Segmentation

Sinkoo

April 10, 2022

1 Introduction

Sparsely annotation semantic segmentation(SASS) aim to train a segmentation networks from partly labeled pixels. This paper proposed a new tree energy loss(TEL) for SASS by providing semantic guidance for unlabeled pixels.
The tree energy loss represents images as minimum spanning trees to model both low-level and high-level pair-wise affinities. In TEL, two minimum spanning trees (MSTs) are built on the low-level color and the high-level semantic features. Each MST is obtained by sequentially eliminating connections between adjacent pixels with large dissimilarity. Then produce pseudo labels by multiply the affinity matrices and predictions in a cascade manner. Combining the TEL with a standard segmentation loss then the network can learn extra information from unlabeled data.

2 Method

2.1 Architecture

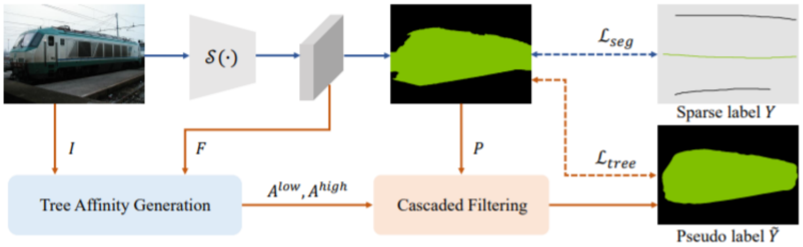


Figure 3. Flowchart of the proposed single-stage SASS method, which is realized by incorporating an auxiliary branch into the traditional segmentation model $\mathcal{S}(\cdot)$. During training, the predicted masks P are split into the labeled and unlabeled parts, which are supervised by the segmentation loss $\mathcal{L}_{seg}$ and the tree energy loss $\mathcal{L}_{tree}$, respectively. To obtain pseudo labels for unlabeled pixels, the Tree Affinity Generation procedure (Eqs. 3-5) first utilizes the color information I and semantic features F to generate the low-level and high-level affinity matrices A^{low}, A^{high}. Then the Cascaded Filtering operation (Eqs. 6-7) converts the network predictions P into soft pseudo labels $\tilde{Y}$. During testing, the auxiliary branch is removed to avoid extra computational costs.

Overall Loss: $L = L_{seg} + \lambda L_{tree}$

$$\mathcal{L}_{seg} = -\frac{1}{|\Omega_L|} \sum_{\forall i \in \Omega_L} Y_i \log(P_i), \qquad (2)$$

where Ω_L represents the labeled data.

2.2 Tree Energy Loss

Tree Affinity Generation

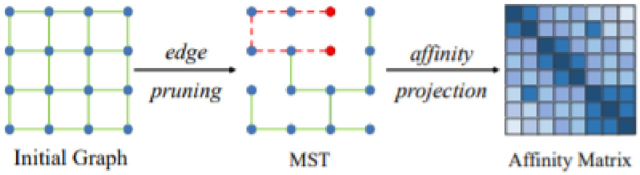


Figure 4. The process of Tree Affinity Generation. An initial graph is first built on the given low-level color or high-level semantic features, then the MST is obtained by the *edge pruning* algorithm [9]. On the MST, the distance between two vertices is calculated by the sum of edge weights along their hyper-edge. An example is illustrated in red dashed lines. Finally, the *affinity projection* is conducted to project the distance map into an affinity matrix.

An image can be represented as an undirected graph $G = (V, E)$
And weights are defined as :

$$\omega^{low}_{i,j} = \omega^{low}_{j,i} = |I(i) - I(j)|^2,$$
$$\omega^{high}_{i,j} = \omega^{high}_{j,i} = |F(i) - F(j)|^2,$$

Calculate Distance between 2 vertices: $D^*_{i,j} = D^*_{j,i} = \sum_{(k,m)\in E^*_{i,j}} \omega^*_{i,j}$
Then project to affinity matrices:

$$A^{low} = exp(-D^{low}/\delta)$$

$$A^{high} = exp(-D^{high})$$

Cascade Filtering Generate pseudo labels $\tilde{Y}$ by:

$$\tilde{Y} = F(F(P, A^{low}), A^{high})$$

And F is :

$$F(P, A^*) = \frac{q}{z_i} \sum_{\forall j \in \Omega} A^*_{i,j} P_j$$

z_i is the normalization term.

Soft Label Assignment $L_{tree} = \delta(P, \tilde{Y})$ Only focus on unlabeled regions.

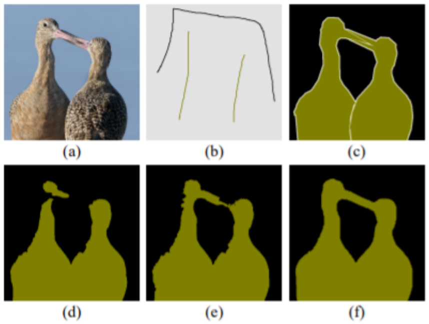


Figure 5. Visualization of the network predictions the corresponding pseudo labels in our training framework. (a) Input image. (b) Sparse annotation. (c) Full annotation. (d) Network prediction. (e) Initial pseudo label generated with the low-level affinity. (f) Final pseudo label generated with multi-level affinities.

3 Results

TEL have great performance when implemented on many backbones.

4 Ablation Study

1, Loss function: Get better performance using L1 distance in L_{tree}.
2, Affinity level: Use image level and feature level affinity improved mIoU respectively, but can get better improvement when use both of them.
3. λ : Around 0.4.

5 Summary

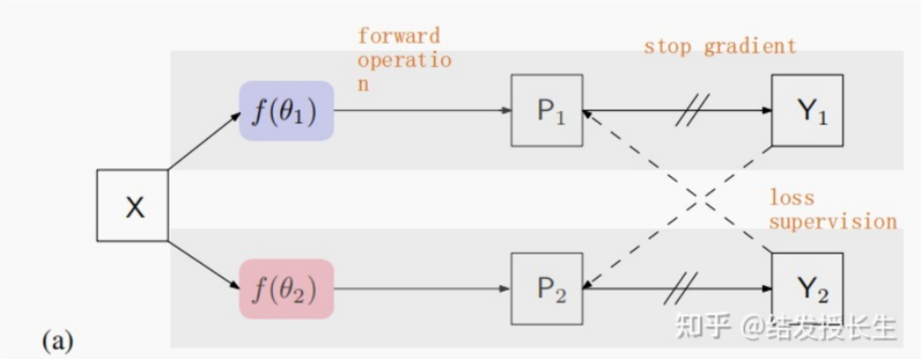
This paper presents a novel TEL as a extra source of information from pixel level affinity to generate pseudo labels to unlabeled pixels in SASS Training. And remarkably it can be easily attached to many existing models with just add an auxiliary branch to the network and a Loss of tree to the normal segmentation Loss. This model outperform the SOTA.

6 Other Knowledge

1, Consistency loss: (SASS,”Semi-Supervised Semantic Segmentation with Cross Pseudo Supervision（CVPR2021）”) Method: Provide 2 network share the same architecture but differ in initialization(parameters):

$$P_1 = f(X; \theta_1)$$

$$P_2 = f(X; \theta_2)$$



Two networks generate 2 pseudo maps Y_1, Y_2
Loss:

$$L_s = \frac{1}{\mathcal{D}^l} \sum_{X \in \mathcal{D}^l} \frac{1}{W \times H} \sum_{i=0}^{W \times H} (l_{ce}(p_{1i}, y^*_{1i}) + l_{ce}(p_{2i}, y^*_{2i}))$$

Use labeled pixels (D^l) to test the performance of the 2 networks.

$$L^u_{cps} = \frac{1}{|\mathcal{D}^u|} \sum_{X \in \mathcal{D}^u} \frac{1}{W \times H} \sum_{i=0}^{W \times H} (l_{ce}(p_{1i}, y_{2i}) + l_{ce}(p_{2i}, y_{1i}))$$

And L^l_{cps} on labeled data. Use Y_1 to supervise P_2. total loss is $L = L_s + \lambda L_{cps}$.