Note: Rethinking Semantic Segmentation: A Prototype View[1]

sgc

April 19, 2022

1 Abstract

Prevalent works on semantic segmentation tasks mostly base on a learnable prototype per class and due to that these methods are limited. So this paper proposed a method that uses multiple of unlearnable prototypes for each class. The authors show that when applying on different models and backbones, this nonparametric framework yields great results on several datasets.
The authors claim that there 3 main limitations:
1, Only 1 prototype per class is insufficient to describe the intra class variance.
2, The parameters number is large.
3, with the cross-entropy loss, only the relative relations between intra-class and inter-class distances are optimized; the actual distances between pixels and prototypes are ignored.

2 Method: Non-Learnable Prototype based Nonparametric Semantic Segmentation

Non-Learnable Prototype based Pixel Classification Each class $c \in 1, ..., C$ is represented by a total of K prototypes $p_{c,k} \in R^D{}_{c,k=1}^{C,K}$, and prototype $p_{c,k}$ is determined as the center of k-th sub-cluster of training pixel samples belonging to class c, in the embedding space ϕ.
The prediction is achieved through:

$$\hat{c}_i = c*, \text{ with } (c*, k*) = argmin< i, p_{c,k} >_{c,k=1}^{C,K})$$

L_{CE} is the cross entropy of probability of pixels over C classes and the GT.

Within-Class Online Clustering The object:

Formally, given pixels $\mathcal{I}^c = \{i_n\}_{n=1}^N$ in a training batch that belong to class c (*i.e.*, $c_{i_n} = c$), our goal is to map the pixels $\mathcal{I}^c$ to the K prototypes $\{\boldsymbol{p}_{c,k}\}_{k=1}^K$ of class c. We denote this pixel-to-prototype mapping as $\boldsymbol{L}^c = [\boldsymbol{l}_{i_n}]_{n=1}^N \in \{0,1\}^{K\times N}$, where $\boldsymbol{l}_{i_n} = [l_{i_n,k}]_{k=1}^K \in \{0,1\}^K$ is the one-hot assignment vector of pixel i_n over the K prototypes. The optimization of $\boldsymbol{L}^c$ is achieved by maximizing the similarity between pixel embeddings, *i.e.*, $\boldsymbol{X}^c = [\boldsymbol{i}_n]_{n=1}^N \in \mathbb{R}^{D\times N}$, and the prototypes, *i.e.*, $\boldsymbol{P}^c = [\boldsymbol{p}_{c,k}]_{k=1}^K \in \mathbb{R}^{D\times K}$:

$$\begin{aligned} &\max_{\boldsymbol{L}^c} \mathtt{Tr}(\boldsymbol{L}^{c\top}\boldsymbol{P}^{c\top}\boldsymbol{X}^c), \\ &\textit{s.t.} \quad \boldsymbol{L}^c \in \{0,1\}^{K\times N}, \boldsymbol{L}^{c\top}\mathbf{1}^K = \mathbf{1}^N, \boldsymbol{L}^c\mathbf{1}^N = \frac{N}{K}\mathbf{1}^K, \end{aligned} \tag{8}$$

where $\mathbf{1}^K$ denotes the vector of all ones of K dimensions. The unique assignment constraint, *i.e.*, $\boldsymbol{L}^{c\top}\mathbf{1}^K = \mathbf{1}^N$, ensures that each pixel is assigned to one and only one prototype. The equipartition constraint, *i.e.*, $\boldsymbol{L}^c\mathbf{1}^N = \frac{N}{K}\mathbf{1}^K$, enforces that on average each prototype is selected at least $\frac{N}{K}$ times in the batch [13]. This prevents the trivial solution: all pixel samples are assigned to a single prototype.

To solve this problem, this paper relaxes L^c

$$\begin{aligned} &\max_{\boldsymbol{L}^c} \mathtt{Tr}(\boldsymbol{L}^{c\top}\boldsymbol{P}^{c\top}\boldsymbol{X}^c) + \kappa h(\boldsymbol{L}^c), \\ &\textit{s.t.} \quad \boldsymbol{L}^c \in \mathbb{R}_+^{K\times N}, \boldsymbol{L}^{c\top}\mathbf{1}^K = \mathbf{1}^N, \boldsymbol{L}^c\mathbf{1}^N = \frac{N}{K}\mathbf{1}^K, \end{aligned}$$

where $h(L^c)$ is the entropy of its elements.

And the solution is:

$$\boldsymbol{L}^c = \mathtt{diag}(\boldsymbol{u})\exp\left(\frac{\boldsymbol{P}^{c\top}\boldsymbol{X}^c}{\kappa}\right)\mathtt{diag}(\boldsymbol{v}),$$

Pixel-Prototype Contrastive Learning Loss:

$$\mathcal{L}_{\text{PPC}} = -\log \frac{\exp(\boldsymbol{i}^\top \boldsymbol{p}_{c_i,k_i}/\tau)}{\exp(\boldsymbol{i}^\top \boldsymbol{p}_{c_i,k_i}/\tau) + \sum_{\boldsymbol{p}^-\in\mathcal{P}^-}\exp(\boldsymbol{i}^\top \boldsymbol{p}^-/\tau)}, \tag{11}$$

Pixel-Prototype Distance Optimization The compactness-aware loss:

$$L_{PPD} = (1 - i^T p_{c_i,k_i})^2$$

Network Learning and Prototype Update The combined loss:

$$L_{SEG} = L_{CE} + \lambda_1 L_{PPC} + \lambda_2 L_{PPD}$$

Prototype evolution is achieved continuously accounting for online clustering results:

$$p_{c,k} = \mu p_{c,k} + (1 - \mu)\bar{i}_{c,k}$$

i represents the embedding pixels assigned to prototype.

3 Summary

This paper proposed a new view to think about prototype in segmentation tasks and inspired us to using more prototypes per class to get more information(like detailed intra class distance). This method yields outstanding performance.

References

[1] T. Zhou, W. Wang, E. Konukoglu, and L. Van Gool, "Rethinking semantic segmentation: A prototype view," *arXiv preprint arXiv:2203.15102*, 2022. (document)