Note: Class-Balanced Pixel-Level Self-Labeling for Domain Adaptive Semantic Segmentation

sgc

June 19, 2022

1 Introduction

Domain adaptive semantic segmentation often produces pseudo labels containing much noise because the model is biased to source domain. This paper proposed to explore the intrinsic pixel distributions of target domain data, instead of heavily rely on the source domain. So the authors similtaneously cluster pixels and rectify pseudo labels with the obtained cluster assignments.
The proposed method: Class-balanced Pixel-level Self Labeling (CPSL).

2 Method

2.1 Overall Framework

The proposed model is described bellow:

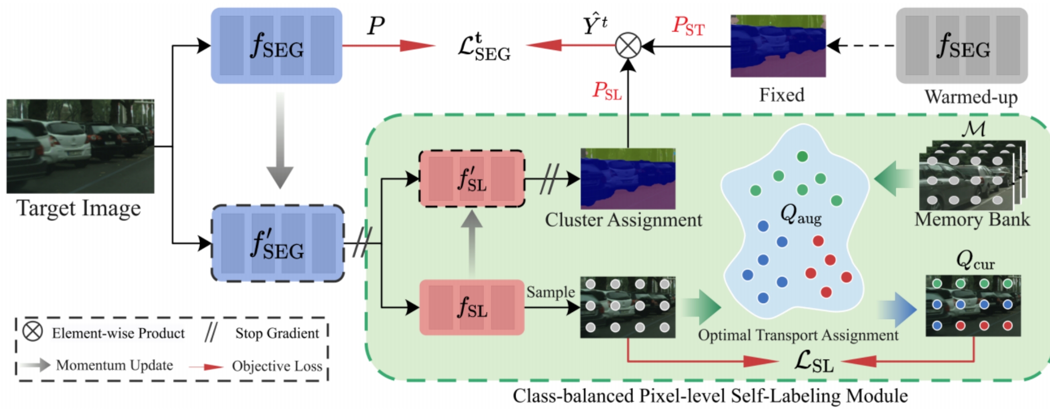

Figure 1. The framework of Class-balanced Pixel-level Self-Labeling (CPSL). The model contains a main segmentation network f_{SEG} and its momentum-updated version f'_{SEG}. The f'_{SEG} is followed by a self-labeling head f_{SL} and its momentum version f'_{SL}, which projects pixel-wise feature embedding into a class probability vector. The pixel-level self-labeling module produces soft cluster assignment P_{SL} to gradually rectify soft pseudo label P_{ST}. Then the segmentation loss $\mathcal{L}^t_{\text{SEG}}$ is computed between the prediction map P and the rectified pseudo label $\hat{Y}^t$. To train the self-labeling head, we randomly sample pixels from each image, and use the memory bank $\mathcal{M}$, which contains previous batches of pixel features, to augment the current batch. Then we compute the optimal transport assignment Q_{aug} over the augmented data by enforcing class balance, and use the assignment of current batch Q_{cur} to compute the self-labeling loss $\mathcal{L}_{\text{SL}}$.

The proposed pixel-level self-labeling module is designed to explore the intrinsic pixel wise distributions of the target domain data via clustering. First the model generate the soft pseudo label map $P_{ST} \in R^{H\times W\times C}$($C$ is the number of categories), then rectify it with the soft clustering assignment.

$$\hat{Y}^{t,(c)}_{n,i} = \begin{cases} 1, & \text{if } c = \underset{c*}{argmax}(P^{(c*)}_{\text{SL},n,i} \cdot P^{(c*)}_{\text{ST},n,i}) \\ 0, & \text{otherwise} \end{cases},$$

Then use the rectified pseudo labels to supervise the segmentation model through loss function:
On target domain:

$$\mathcal{L}^t_{\text{SEG}} = -\sum_{n=1}^{N_T}\sum_{i=1}^{H\times W}\sum_{c=1}^{C} \hat{Y}^{t,(c)}_{n,i} \log P^{(c)}_{n,i}.$$

On source domain:

$$\mathcal{L}^s_{\text{SEG}} = -\sum_{n=1}^{N_S} \sum_{i=1}^{H\times W} \sum_{c=1}^{C} Y^{s,(c)}_{n,i} \log P^{(c)}_{n,i}.$$

The total loss is the sum of 2 loss aforementioned.

2.2 Online Pixel level Self Labeling

Pixel-Level Self-Labeling The proposed method is clustering on target domain and use the obtained cluster assignments to rectify the pseudo labels.

Weight Initialization The prototypes are computed below:

$$\bar{\mathbf{z}}_c = \frac{1}{|\Gamma_c|} \sum_{n=1}^{N_T} \sum_{i=1}^{H\times W} Y^{(c)}_{\text{ST},n,i} \cdot z_{n,i},$$

Where Y_{ST} is the hard version of P_{ST}.

Online Cluster Assignments This model maintains a memory banks to store a queue of 65536 pixel features from previous batches. In each iteration, compute the optimal transport assignment on the augmented data $Zaug$, denoted by $Qaug$, but only the assignment of current batch, denoted by $Qcur$, is used to compute the self-labeling loss.

2.3 Class-Balanced Self-Labeling

This part consists of 2 techniques: class-balanced sampling and distribution alignment.

Class-Balanced Sampling First compute the class distribution:

$$\delta^{(c)}_n = \frac{1}{H\times W} \sum_{i}^{H\times W} \hat{Y}^{t,(c)}_{n,i},$$

Then the number of sample is generated by:

$$M_c = \left\lfloor M \times \delta^{(c)}_n \right\rfloor.$$

Distribution Alignment This method is used to align the distribution of cluster assignments to ground truth class distribution σ_{gt}. Since GT is unknown, use the moving average to approximate σ_{gt}. First initialize the moving average as:

$$\delta_{pseudo}^{(c)}|_0 = \frac{1}{N_T \times H \times W} \sum_{n}^{N_T} \sum_{i}^{H \times W} Y_{\mathrm{ST},n,i}^{t,(c)}.$$

Over the course of training the moving average is updated by:

$$\delta_{pseudo}^{(c)}|_k = \alpha \delta_{pseudo}^{(c)}|_{k-1} + (1-\alpha)\delta_{n}^{(c)}.$$

The usage of the distribution is detailed in the paper. They are used in Pixel-Level Self-Labeling.

2.4 Loss Function

Specifically, we generate a weakly-augmented image Xw and a strongly-augmented image X_s from the same input image X, and pass X_w through the momentum segmentation network f'_{SEG} to generate a probability map Pw, which is used to supervise the output Ps of strongly augmented image X_s from f_{SEG}. Then enforce P_w and P_s to be consistent via:

$$\mathcal{L}_{\mathrm{REG}} = \sum_{n=1}^{N_T} \sum_{i=1}^{H \times W} \left(\ell_{\mathrm{KL}}\left(P_{w,n,i}, P_{s,n,i}\right) + \ell_{\mathrm{KL}}\left(P_{s,n,i}, P_{w,n,i}\right)\right), \tag{13}$$

The total loss is: $\mathcal{L}_{TOTAL} = \mathcal{L}_{SEG} + \lambda_1 \mathcal{L}_{SL} + \lambda_2 \mathcal{L}_{TOTAL}$

3 Summary

CPSL is a plug-and-play module that can be seamlessly incorporated into self-training pipelines to improve the domain adaptive semantic segmentation performance. This model use the pixels clustering to rectify the pseudo labels. This model achieved SOTA.

References