Note: Background-Aware Pooling and Noise-Aware Loss for Weakly-Supervised Semantic Segmentation

sgc

May 29, 2022

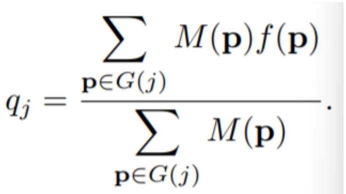
1 Introduction

This paper introduces a new WSSS method using bounding box annotations. In detail, authors propose a background aware pooling(BAP) to discriminate foreground and background inside the bounding boxes, thus generating more accurate CAMs. They also proposed a noise-aware loss(NAL) to train CNNs for semantic segmentation that makes the network less susceptible to incorrect labels. Their experiments show that with BAP this method already outperforms the SOTA. And NAL further boosts the performance.
The main contributions:
1, This paper proposes a new pooling method BAP to take use of box annotations.
2, NAL is introduced to exploit the distances between CNN features for prediction and classifier weights for semantic segmentation, lessening the influence of incorrect labels.
3, This method sets the new SOTA.

2 Method

2.1 Image classification using BAP

First divide the image into N^2 grids. Compute the query as follows:



$$q_j = \frac{\sum\limits_{\mathbf{p}\in G(j)} M(\mathbf{p})f(\mathbf{p})}{\sum\limits_{\mathbf{p}\in G(j)} M(\mathbf{p})}.$$

where M denotes to the mask, f is the feature. $M(p) = 1$ if the position p dose not belong to any bounding boxes.

The attentions map is :

$$A(\mathbf{p}) = \frac{1}{J}\sum_{j} A_j(\mathbf{p}),$$

where:

$$A_j(\mathbf{p}) = \begin{cases} \text{ReLU}\left(\frac{f(\mathbf{p})}{\|f(\mathbf{p})\|} \cdot \frac{q_j}{\|q_j\|}\right) & , \mathbf{p} \in \mathcal{B} \\ 1 & , \mathbf{p} \notin \mathcal{B} \end{cases}.$$

A pixel is more likely to be a background, as the value of the attention map A approaches to one.
In BAP, the foreground features for each bounding box is aggregated using the

$$r_i = \frac{\sum_{\mathbf{p}\in B_i} (1 - A(\mathbf{p}))f(\mathbf{p})}{\sum_{\mathbf{p}\in B_i} (1 - A(\mathbf{p}))},$$

attention map A:
It is a weighted average pooling, the weight is the probability of a pixel being foreground. (remarkably GAP belongs to the situation that $A = 0$)

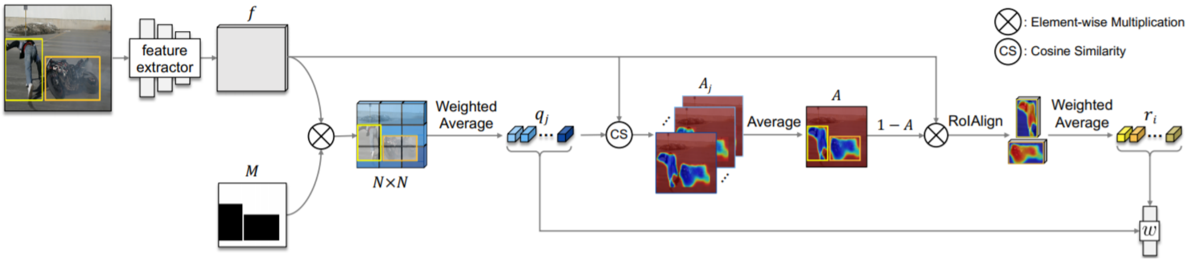


Figure 2: Overview of image classification using BAP. We first extract queries q_j using a feature map f and a binary mask M indicating a definite background. The queries q_j are then used to compute an attention map A describing the likelihood that each pixel belongs to a background. The attention map enables localizing entire foreground regions, leading to better foreground features r_i. Finally, we apply a softmax classifier w to the foreground features r_i for each bounding box together with the queries q_j. The entire network is trained with a cross-entropy loss. See text for details (Sec. 3.1). Best viewed in color.

2.2 Pseudo label generation

First the Y_{crf}:

$$u_c(\mathbf{p}) = \begin{cases} \frac{\mathrm{CAM}_c(\mathbf{p})}{\max_{\mathbf{p}}(\mathrm{CAM}_c(\mathbf{p}))} & , \mathbf{p} \in \mathcal{B}_c \\ 0 & , \mathbf{p} \notin \mathcal{B}_c \end{cases}, \tag{5}$$

$$u_0(\mathbf{p}) = A(\mathbf{p}).$$

Then Y_{ret} is generated from high level features.
First extract a prototype feature foe each class as follow:

$$q_c = \frac{1}{|\mathcal{Q}_c|} \sum_{\mathbf{p} \in \mathcal{Q}_c} f(\mathbf{p}),$$

where Q_c is a set of locations labeled as the class c in Y_{crf} (including the background class). Then the correlation map:

$$C_c(\mathbf{p}) = \frac{f(\mathbf{p})}{\|f(\mathbf{p})\|} \cdot \frac{q_c}{\|q_c\|}.$$

then obtain pseudo segmentation labels Y_{ret} by applying the argmax function over the correlation maps C^c.

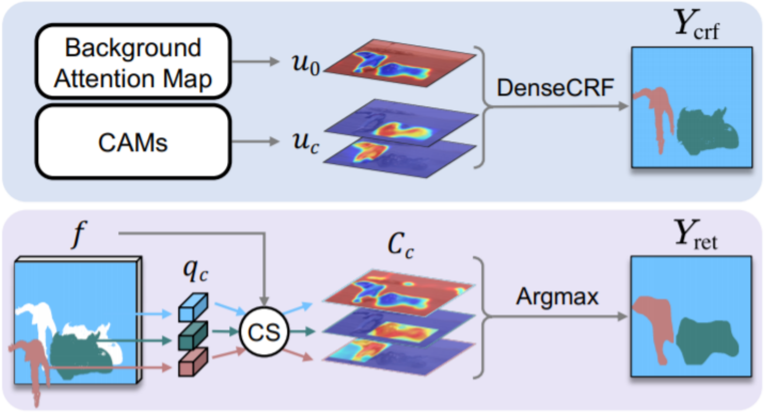


Figure 3: Generating pseudo labels. We compute u_0 and u_c using a background attention map and CAMs, respectively, which are used as a unary term for DenseCRF [29] to obtain pseudo segmentation labels Y_{crf}. We extract prototypical features q_c for each class using the labels Y_{crf}, and use them as queries to retrieve high-level features from the feature map f, from which we obtain additional pseudo labels Y_{ret}. See text for detail (Sec. 3.2). Best viewed in color.

2.3 Semantic segmentation with noisy labels

Extract a feature map ϕ from the penultimate layer, and pass it through a softmax classifier W, resulting in a $(L+1)$-dimensional probability map H. To alleviate the influence of incorrect labels, the regions S where both Y_{crf} and Y_{ret} give the same label, to compute the loss as follows:

$$\mathcal{L}_{\text{ce}} = -\frac{1}{\sum_c |\mathcal{S}_c|} \sum_c \sum_{\mathbf{p} \in \mathcal{S}_c} \log H_c(\mathbf{p}),$$

where c denotes the class.
To take use of the rest regions, a correlation map for each class is computed as follows:

$$D_c(\mathbf{p}) = 1 + \left(\frac{\phi(\mathbf{p})}{\|\phi(\mathbf{p})\|} \cdot \frac{W_c}{\|W_c\|} \right),$$

where W_c denotes the classifier weight for the corresponding class c.
then compute a confidence map:

$$\sigma(\mathbf{p}) = \left(\frac{D_{c^*}(\mathbf{p})}{\max_c(D_c(\mathbf{p}))} \right)^{\gamma},$$

where c is a label obtained by Y_{crf} (i.e., $c = Y_{crf}(p)$)

$$\mathcal{L}_{\text{wce}} = -\frac{1}{\sum_c \sum_{\mathbf{p} \in \sim \mathcal{S}_c} \sigma(\mathbf{p})} \sum_c \sum_{\mathbf{p} \in \sim \mathcal{S}_c} \sigma(\mathbf{p}) \log H_c(\mathbf{p}), \tag{13}$$

The total loss is :

$$\mathcal{L} = \mathcal{L}_{\text{ce}} + \lambda \mathcal{L}_{\text{wce}}.$$

References