Note: Learning Affinity from Attention: End-to-End Weakly-Supervised Semantic Segmentation with Transformers

Sinkoo

April 10, 2022

1 Introduction

Convolution CNN failed to use global information and thus limits the performance. This paper proposes an Affinity from the multi-head attention(AFA) to learn semantic affinity from pseudo labels. The learned affinity is then leveraged to refine the initial pseudo labels for segmentation.
To derive highly-confident pseudo affinity labels for AFA and ensure the local consistency of the propagated pseudo labels, we further propose a Pixel-Adaptive-Refinement module (PAR). PAR can incorporate the RGB and position information for label refinement.
Remarkably, this is the first works to implement Transformers to WSSS.

2 Method

2.1 Affinity from Attention

Even though MHSA can provide semantic affinity somehow, but directly implement it does not work really well. So this paper proposed AFA.
Assuming MHSA in a transformer block is denoted as $S \in R^{hw*hw*n}$, where hw is the flattened spatial size and n is the number of attention heads. In AFA, the predicted semantic affinity matrix is denoted as: $A = MLP(S + S^T)$.

Pseudo Affinity Label Generation The key step to learn the best A is to derive a reliable pseudo affinity label Y_{aff} as supervision.
Given GAMs, the pseudo labels are constructed as:

$$Y_p^{i,j} = \begin{cases} \mathtt{argmax}(M^{i,j,:}), & \text{if } \mathtt{max}(M^{i,j,:}) \geq \beta_h, \\ 0, & \text{if } \mathtt{max}(M^{i,j,:}) \leq \beta_l, \\ 255, & \text{otherwise}, \end{cases}$$

where 0 and 255 denote the background class and ignored regions. Use β_h and β_l to filter the refined pseudo labels to reliable foreground, background, and uncertain regions.
The pseudo affinity label Y_{aff} is derived from Y_p. for Y_p, if the pixel (i, j) and (k, l) share the same semantic, we set their affinity as positive; otherwise, their affinity is set as negative. And we only care about pixels in the same local window, disregard pixels in distance.

Affinity Loss A affinity loss term L_{aff} is constructed as :

$$\mathcal{L}_{aff} = \frac{1}{N^-} \sum_{(ij,kl)\in\mathcal{R}^-} (1 - \texttt{sigmoid}(A^{ij,kl})) + \frac{1}{N^+} \sum_{(ij,kl)\in\mathcal{R}^+} \texttt{sigmoid}(A^{ij,kl}),$$

where R^+ and R^- denote the set of positive and negative samples in Y_{aff}, N is the number of R.

Propagation with Affinity The semantic transition matrix T is derived as: $T = D^{-1}A^{\alpha}$, with $D^{ii} = \sum_k A^{jk^{\alpha}}$
where α is a hyper parameter to ignore trivial affinity values in A, and D is a diagonal matrix to normalize A rowwise.
The random walk propagation for the initial CAM $M \in R^{h*w*C}$ is accomplished as:

$$M_{aff} = T * vec(M)$$

vec(M) means M is vectorized.
Applying this propagation, the class activation is refine using the affinity of the pixel and the whole image, with pixels with high affinity contributes more to class activation.

2.2 Pixel-Adaptive Refinement

PAR: the RGB and spatial pairwise terms are defined as:

$$\kappa_{rgb}^{ij,kl} = -\left(\frac{|I_{ij} - I_{kl}|}{w_1 \sigma_{rgb}^{ij}}\right)^2, \quad \kappa_{pos}^{ij,kl} = -\left(\frac{|P_{ij} - P_{kl}|}{w_2 \sigma_{pos}^{ij}}\right)^2,$$

The affinity kernel for PAR is :

$$\kappa^{ij,kl} = \frac{\exp(\kappa_{rgb}^{ij,kl})}{\sum_{(x,y)} \exp(\kappa_{rgb}^{ij,xy})} + w_3 \frac{\exp(\kappa_{pos}^{ij,kl})}{\sum_{(x,y)} \exp(\kappa_{pos}^{ij,xy})},$$

where (x, y) is sampled from the neighbor set of (i, j).
The refinement is conducted for multiple iterations. For CAM $M \in R^{h*w*C}$, in iteration t:

$$M_t^{i,j,c} = \sum_{(k,l)\in\mathcal{N}(i,j)} \kappa^{ij,kl} M_{t-1}^{k,l,c}.$$

where N represents the neighbor field.

2.3 Training Network

Total Loss : $L = L_{cls} + \lambda_1 L_{seg} + \lambda_2 L_{aff} + \lambda_3 L_{reg}$
classification loss:

$$\mathcal{L}_{cls} = \frac{1}{C}\sum_{c=1}^{C}(y^c \log(p_{cls}^c) + (1-y^c)\log(1-p_{cls}^c)),$$

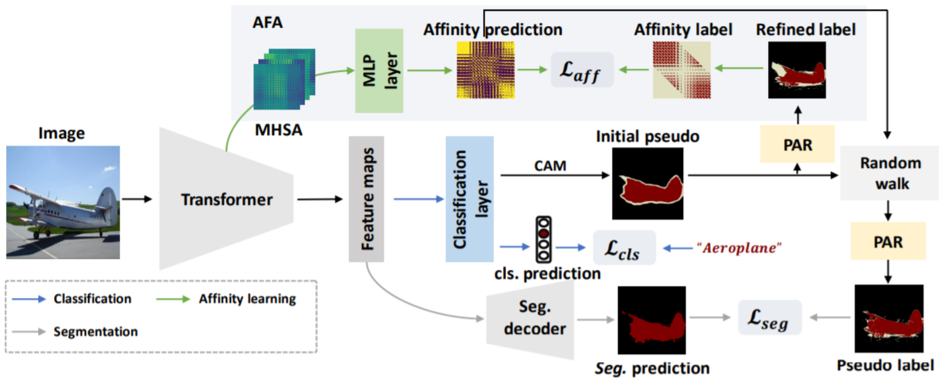

Figure 3. The proposed end-to-end framework for WSSS. We use a Transformer backbone as the encoder to extract feature maps. The initial pseudo labels are generated with CAM [59] and then refined with the proposed PAR. In the AFA module, we derive the semantic affinity from MHSA in Transformer blocks. AFA is supervised with the pseudo affinity labels derived from the refined label. Next, we employ the learned affinity to revise the pseudo labels via random walk propagation [2, 1]. The propagated labels are finally refined with PAR as the pseudo labels for the segmentation branch.

3 Summary

This paper proposed AFA, which is the first method to implement Transformer to WSSS by using its MHSA to refine pseudo labels Generating. It also proposed PAR to refine the local appearance or making pixels 'smoother', enforcing better alignment with the low level boundaries.

4 Questions

1, In section 3.3, function (5), (j, k) and (k, l) mean pixel but for $A \in R^{hw*hw}$,(h, w) is used to describe patches. how dose the function work?