Note :Multi-class Token Transformer for Weakly Supervised Semantic Segmentation

Sinkoo

April 3, 2022

1 Abstract

ViT uses only one class token makes it a challenge to to accurately localize different objects on a single image for 2 reasons: First, a one-class-token design essentially inevitably captures context information from other object categories and the background. thus resulting in a rather non-discriminative and noisy object localization. Second, the model uses the only one-class token to learn interactions with patch tokens for a number of distinct object classes in a dataset(? not understand these reasons). This paper proposed a Multi-class Token Transformer(MCTformer) using multiple class tokens to learn interaction between class tokens and the patch tokens.

2 Method

Apply average pooling on the output class tokens(learnable) from the transformer encoder along the embedding dimension to generate class scores(unlike conventional transformer which uses a MLP),which are directly supervised by the ground-truth class labels. This thus builds a one-to-one strong connection between each class token and the corresponding class label.

2.1 Overview

A classification loss is computed between the class scores directly produced by class tokens and the ground-truth class labels and thus builds a strong connection between class token and the corresponding category.

2.2 Class-Specific Transformer Attention Learning

Multi-class token structure design Simply concatenate patch tokens with C class tokens and position embedding to form the input tokens.

Class-specific multi-class token attention Use self-attention operate on input tokens :

$$Attention(Q, K, V) = softmax(QK^T/\sqrt{D})V$$

Then we obtain a token-to-token attention map $A_{t2t} \in R^{(C+M)*(C+M)}$ and $A_{t2t} = softmax(QK^T/\sqrt{D})$ (no V?)
Class to patched attention map: $A_{c2p} = A_{t2t}[1 : C, C + 1 : C + M]$Each row represents the attention scores of a specific class to all patches.
Author proposed to fuse the class-to-patch attentions from the last K transformer encoding layers:

$$\hat{A}_{mct} = \frac{1}{K}\sum_{l}^{K} \hat{A}^l_{mct}$$

$\hat{A}^l_{mct}$ is s the class-specific transformer attention extracted from the l- th transformer encoding layer. And then $\hat{A}_{mct}$ are normalized (by by the min-max normalization method along the two spatial dimensions) to generate the final class-specific object localization maps.

Class-specific attention refinement extracting the patch-to-patch attentions

$$A_{p2p} \in R^{M*M}, A_{p2p} = A_{t2t}[C + 1 : C + M, C + 1 : C + M]$$

and reshape to a 4D tensor to further refine the class-specific transformer attention.

$$A_{mct_ref}(c, i, j) = \sum_{k}^{N}\sum_{l}^{N} \hat{A}_{p2p}(i, j, k, l)A_{mct}(c, k, l)$$

Overall Architecture:

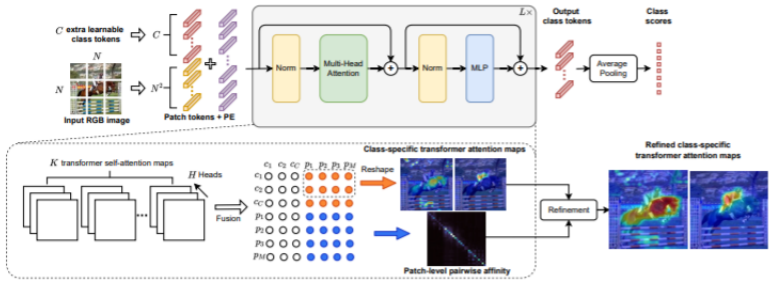

Figure 2. An overview of the proposed multi-class token transformer (MCTformer-V1). It first splits and transforms an input RGB image into a sequence of patch tokens. We propose to learn C extra class tokens, where C is the number of classes. The C class tokens are concatenated with patch tokens, with added position embeddings (**PE**), which then go through consecutive L transformer encoding layers. Finally, the output C class tokens are used to produce class scores via average pooling. We aggregate the transformer attentions from the last K layers and multiple heads to generate a final attention map, from which we can extract class-specific object localization maps and a patch-level pairwise affinity map from the class-to-patch and the patch-to-patch attentions, respectively. The patch-level pairwise affinity can be used to refine the class-specific transformer attention maps to produce improved object localization maps.

Class-aware Training Use pooling on the output class tokens to produce class scores:

$$y(c) = \frac{1}{D}\sum_{j}^{D} T_{cls}(c,j)$$

,then compute a multi-label soft margin loss between the class score y(c) for the class c and its ground-truth label.

2.3 Complementarity to Patch-Token CAM

integrate a CAM and MCTfromer(MCTformerV2):

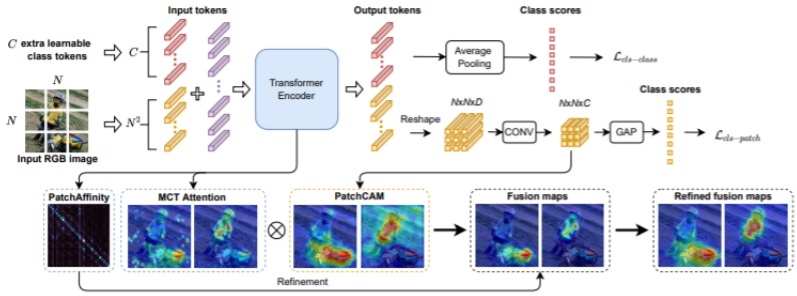

Figure 3. An overview of the proposed MCTformer-V2. We introduce a CAM module into the proposed MCTformer-V1. More specifically, the CAM module is composed of a convolutional layer and a global average pooling (GAP) layer. It takes the reshaped output patch tokens from the last transformer encoding layer as inputs, and outputs class scores. As for MCTformer-V1, we also use the output class tokens to produce class scores. The whole model is thus optimized by two classification losses applied on separately two types of class predictions. At the inference time, we fuse the class-specific transformer attentions (MCT Attention) and the PatchCAM maps. The results are further refined by the patch affinity extracted from the patch-to-patch transformer attentions to produce the final object localization maps.

The total loss is the sum of two multi-label soft margin losses computed between the image-level ground-truth labels and the class predictions respectively from the class tokens and the patch tokens as follows:

$$L_{total} = L_{cls-class} + L_{cls-patch}$$

Combining PatchCAM and class-specific transformer attention Extract feature map $F_{out-pat}$ from the last convolution layer and apply the min-max normalization to form the PatchCAM maps $A_{pCAM} \in R^{N*N*C}$ .Then combined with A_{mct} to produce fused object localization maps A. $A = A_{pCAM} \circ A_{mct}$

Class-specific object localization map refinement

$$A_{ref}(c,i,j) = \sum_{k}^{N}\sum_{l}^{N} \hat{A}_{p2p}(i,j,k,l)A(c,k,l)$$

By applying the classification loss on class predictions from both class tokens and patch tokens, the strong consistency between these two types of tokens can be enforced to improve the model learning.

3 Result

Params of MCTformer significantly smaller than ResNet. And in the experiments, MCTformer achieves a segmentation mIoU of 42.0%, surpassing the recent methods by a large margin.

4 Summary

This paper presents MCTformer, a simple and effective transformer-based framework to produce class-specific object localization maps as a kind of pseudo label to help do semantic segmentation, and achieves state-of-the-art results. By using multi-class-token transformer we can handle more information to effectively capture class-specific attention for more discriminative object localization. And a patchlevel pairwise affinity, combining with CAM greatly improved performance. This idea may derive from Attention map kinda resemble features that can help learning.