Multi-class Token Transformer for Weakly Supervised Semantic Segmentation

Lian Xu[1], Wanli Ouyang[2], Mohammed Bennamoun[1], Farid Boussaid[1], and Dan Xu[3]
[1]The University of Western Australia [3]Hong Kong University of Science and Technology
[2]The University of Sydney, SenseTime Computer Vision Group, Australia
{lian.xu,mohammed.bennamoun,farid.boussaid}@uwa.edu.au,
wanli.ouyang@sydney.edu.au, danxu@cse.ust.hk

Abstract

This paper proposes a new transformer-based framework to learn class-specific object localization maps as pseudo labels for weakly supervised semantic segmentation (WSSS). Inspired by the fact that the attended regions of the one-class token in the standard vision transformer can be leveraged to form a class-agnostic localization map, we investigate if the transformer model can also effectively capture class-specific attention for more discriminative object localization by learning multiple class tokens within the transformer. To this end, we propose a Multi-class Token Transformer, termed as ***MCTformer****, which uses multiple class tokens to learn interactions between the class tokens and the patch tokens. The proposed MCTformer can successfully produce class-discriminative object localization maps from the class-to-patch attentions corresponding to different class tokens. We also propose to use a patch-level pairwise affinity, which is extracted from the patch-to-patch transformer attention, to further refine the localization maps. Moreover, the proposed framework is shown to fully complement the Class Activation Mapping (CAM) method, leading to remarkably superior WSSS results on the PASCAL VOC and MS COCO datasets. These results underline the importance of the class token for WSSS.* [1]

1. Introduction

Weakly supervised semantic segmentation (WSSS) aims to alleviate the reliance on pixel-level ground-truth labels by using weak supervision. A critical step for this task is to generate high-quality pseudo segmentation ground-truth labels by using weak labels. Image-level labels can provide simple weak labels which only indicate the presence or absence of certain classes without any ground-truth localization information. Previous WSSS methods generally rely on Class Activation Mapping (CAM) [51] to extract object localization maps from Convolutional Neural Networks (CNNs). Despite using complex CAM expansion strategies or multiple training steps, existing methods still exhibit limited performance in terms of both completeness of the localized objects and accuracy.

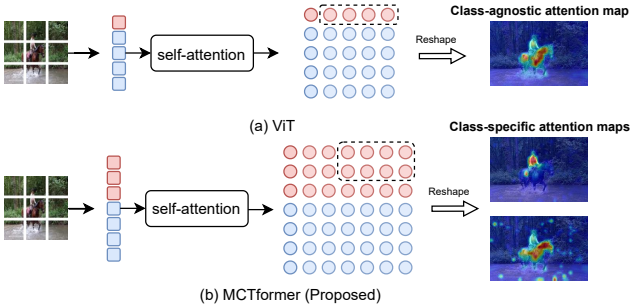

Figure 1. (a) In previous vision transformers [10], only one class token (red square) is used to aggregate information from patch tokens (blue square). The learned patch attentions corresponding to the class token generate a class-agnostic localization map. (b) In contrast, the proposed MCTformer uses multiple class tokens to learn interactions between class tokens and patch tokens. The learned class-to-patch attentions of different class tokens can produce class-specific object localization maps.

Vision Transformer (ViT) [10], as the first transformer model specifically designed for computer vision, has recently achieved performance breakthroughs on multiple vision tasks [18]. Particularly, ViT has achieved state-of-the-art performance for large-scale image recognition, thanks to its strong capability to model long-range contexts. ViT splits the input image into non-overlapping patches and transforms them into a sequence of vectors. ViT also uses *one* extra class token to aggregate information from the entire sequence of the patch tokens. Although the class token has been removed in a number of recent transformer methods [7, 8, 29], this work will underline its importance for weakly supervised semantic segmentation.

A recent work, DINO [3], revealed that there was explicit information about the semantic segmentation of an image in self-supervised ViT features. More specifically, it was ob-

[1] https://github.com/xulianuwa/MCTformer

served that a semantic scene layout can be discovered from the attention maps of the class token. These attention maps lead to promising results in the unsupervised segmentation task. Although it was demonstrated that different heads in the transformer attention can attend to different semantic regions of an image, it remains unclear how to associate a head to a correct semantic class. That is, these attention maps are still class-agnostic (see Figure 1).

It is challenging to exploit class-specific attention from transformers. We argue that existing transformer-based works have a common issue, *i.e.*, using only *one* class token, which makes the accurate localization of different objects on a single image challenging. There are two main reasons for this. **First**, a one-class-token design essentially inevitably captures context information from other object categories and the background. In other words, it naturally learns both class-specific and generic representations for different object classes as only one class token is considered, 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. The model capacity is consequently not adequate enough to achieve the targeted discriminative localization performance.

To tackle these issues, a straightforward idea is to leverage multiple class tokens, which will be responsible for learning representations for different object classes. To this end, we propose a Multi-class Token Transformer (MCTformer), in which *multiple* class-specific tokens are employed to exploit class-specific transformer attention. Our goal of having class-specific tokens cannot be achieved by simply increasing the number of class tokens in ViT, because these class tokens still do not have specific meanings. To ensure that each class token can effectively learn high-level discriminative representations of a specific object class, we propose a class-aware training strategy for multiple class tokens. More specifically, we apply average pooling on the output class tokens from the transformer encoder along the embedding dimension, to generate class scores, 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. Through this design, one significant advantage is that the learned class-to-patch attention of different classes can be directly used as class-specific localization maps.

It is worth noting that the learned patch-to-patch attention, as a byproduct of training without additional computation, can serve as a patch-level pairwise affinity. This can be used to further refine the class-specific transformer attention maps, dramatically improving the localization performance. Moreover, we also show that the proposed transformer framework fully complements the CAM method when applied on patch tokens (by simultaneously learning to classify with class-token and patch-token based representations). This leads to high consistency between class tokens and patch tokens, thus considerably enhancing the discriminative ability of their derived object localization maps.

In summary, the main contribution is three-fold:

- We propose to exploit class-specific transformer attentions for weakly supervised semantic segmentation.
- We propose an effective transformer framework, which includes a novel multi-class token transformer (MCTformer) coupled with a class-aware training strategy, to learn class-specific localization maps from the class-to-patch attention of different class tokens.
- We propose to use the patch-to-patch transformer attentions as a patch-level pairwise affinity, which can significantly refine the class-specific transformer attentions. Furthermore, the proposed MCTformer can fully complement the CAM mechanism, leading to high-quality object localization maps.

The proposed method can generate high-quality class-specific multi-label localization maps for WSSS, establishing new state-of-the-art results on PASCAL VOC (mIoU of 71.6% on the test set) and MS COCO (mIoU of 42.0%).

2. Related works

2.1. Weakly supervised semantic segmentation

Most existing WSSS approaches rely on Class Activation Mapping [51] to extract object localization maps from CNNs. The raw CAM maps are incomplete with coarse boundaries and thus unable to provide sufficient supervision to the learning of semantic segmentation networks. To tackle this problem, specific segmentation losses have been proposed to cater for deficient segmentation supervision, including SEC loss [19], CRF loss [35, 46] and contrastive loss [17]. In addition, a number of studies have focused on improving the pseudo segmentation labels obtained from CAM maps. These methods can be categorized as follows:

Generating high-quality CAM maps. A few methods developed heuristic strategies, such as "Hide & Seek" [31] and Erasing [40], applied either on images [24, 48] or feature maps [16, 21] to drive the network to learn novel object patterns. Prior works also exploited sub-categories [4] and cross-image semantics [13, 25, 33] to localize more accurate object regions. To address the limitation of the standard classification objective loss function, regularization losses [39, 49] have been proposed to guide the network to discover more object regions. Moreover, several other works [41] addressed the problem of the limited receptive field of standard image classification CNNs by introducing dilated convolutions, to encourage the propagation of discriminative activations to their surroundings.

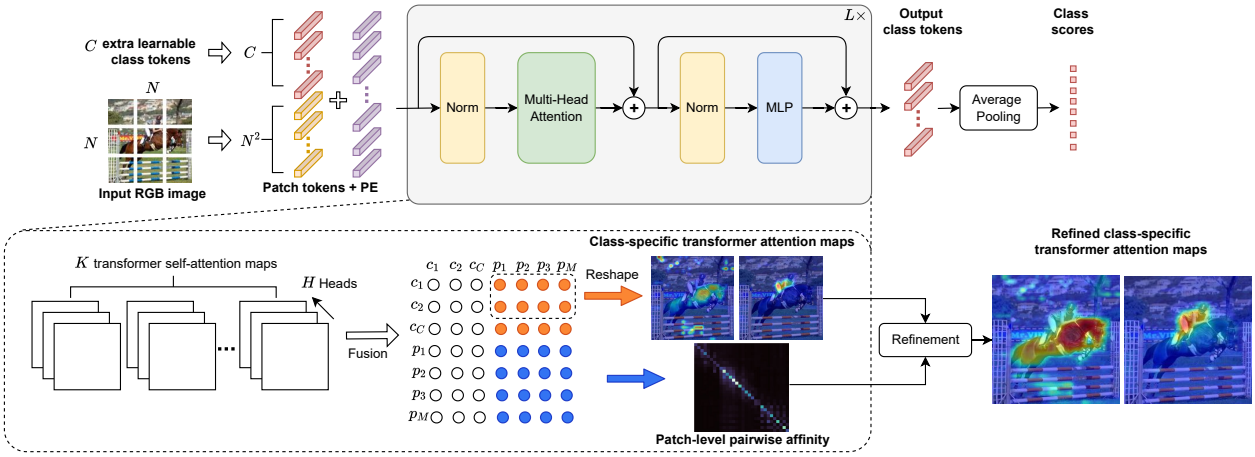

Figure 2. An overview of the proposed multi-class token transformer (MCTformer-V1). It first splits and transforms an input RGB image into a sequence of patch tokens. We propose to learn C extra class tokens, where C is the number of classes. The C class tokens are concatenated with patch tokens, with 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.

Refining CAM maps with affinity learning. Several works focused on learning pairwise semantic affinities to refine the CAM maps. Ahn *et al*. [1] proposed AffinityNet to learn the affinities between adjacent pixels from the reliable seeds of the raw CAM maps. The learned AffinityNet can predict an affinity matrix to propagate the CAM maps via random walk. Similarly, Wang *et al*. [38] also learned a pairwise affinity network using the confident pixels from the segmentation results. In [39, 48], the affinity is directly learned from the feature maps of the classification network to refine the CAM maps. In addition, Xu *et al*. [44] proposed a cross-task affinity, which is learned from the saliency and segmentation representations in a weakly supervised multi-task framework.

In contrast to previous WSSS methods, which are all based on CNNs, we propose a transformer based model to extract class-specific object localization maps. We exploit the transformer attention map from the self-attention mechanism to generate object localization maps.

2.2. Transformers for visual tasks

Transformers [37], were originally designed to model long-range dependencies of long sequences in the field of NLP. Recently, transformer models have been adapted to accommodate a wide variety of vision tasks [18], such as image classification [10], saliency detection [27] and semantic segmentation [30], achieving promising performances. The first transformer based vision model, ViT [10], splits an image into patches and transforms them into a sequence of tokens. These tokens are then forwarded into multiple stacked self-attention [37] based layers, enabling each patch to have a global receptive field.

Caron *et al*. [3] adapted self-supervised methods to ViT and observed that the attentions of the class token on patches contain information about the semantic layout of the scenes. However, the one-to-one mapping between the attention and the class was not established in [3]. Besides, their findings on the transformer attention have not been extended to weakly supervised learning.

Another related work, TS-CAM [14], adapts a CAM module to ViT. However, TS-CAM only leverages the class-agnostic attention maps of ViT, while the proposed method exploits class-specific localization maps from the transformer attention. Moreover, the proposed multi-class token transformer framework is shown to better complement the CAM mechanism than the original ViT, generating better object localization maps than TS-CAM (see Table 5).

3. Multi-class Token Transformer

3.1. Overview

We propose a novel purely transformer-based framework (MCTformer-V1) to exploit class-specific object localization maps from the transformer attention. The overall architecture of MCTformer-V1 is shown in Figure 2. An input RGB image is first split into non-overlapping patches, which are then transformed into a sequence of patch tokens. In contrast to conventional transformers, which only use one class token, we propose to use multiple class tokens. These class tokens are concatenated with patch tokens, embedding position information, to form the input tokens of the transformer encoder. Several transformer blocks are used in the transformer encoder to extract features for both

patch tokens and class tokens. We apply average pooling on the output class tokens from the last layer to generate class scores, instead of using a Multi-Layer Perception (MLP) as in conventional transformers, for classification prediction.

At training time, to ensure that different class tokens can learn different class-specific representations, we adopt the class-aware training strategy detailed in Section 3.2. A classification loss is computed between the class scores directly produced by class tokens and the ground-truth class labels. This thus enables a strong connection between each class token and the corresponding class label. At test time, we can extract class-specific localization maps from the class-to-patch attention in the transformer. We further aggregate the attention maps from multiple layers to utilize complementary information learned from different transformer layers. Moreover, a patch-level pairwise affinity can be extracted from the patch-to-patch attentions, to further refine the class-to-patch attentions, leading to significantly improved class-specific localization maps. The class-specific localization maps are used as the seed to generate pseudo labels to supervise the segmentation model.

3.2. Class-Specific Transformer Attention Learning

Multi-class token structure design. Consider an input image, it is split into $N \times N$ patches, which are then transformed into a sequence of patch tokens $\mathbf{T}_p \in \mathbb{R}^{M \times D}$, where D is the embedding dimension, $M = N^2$. We propose to learn C class tokens $\mathbf{T}_c \in \mathbb{R}^{C \times D}$, where C is the number of classes. The C class tokens are concatenated with patch tokens, with added position embeddings to form the input tokens $\mathbf{T}_{in} \in \mathbb{R}^{(C+M) \times D}$ to the transformer encoder. The transformer encoder has L consecutive encoding layers, each of which consists of a Multi-Head Attention (MHA) module, a MLP, and two LayerNorm layers applied before the MHA and the MLP, respectively.

Class-specific multi-class token attention. We use the standard self-attention layer to capture the long-range dependencies between tokens. More specifically, we first normalize the input token sequence and transform it to a triplet of $\mathbf{Q} \in \mathbb{R}^{(C+M) \times D}$, $\mathbf{K} \in \mathbb{R}^{(C+M) \times D}$ and $\mathbf{V} \in \mathbb{R}^{(C+M) \times D}$, through linear layers [10]. We employ the Scaled Dot-Product Attention [37] mechanism to compute the attention values between the queries and keys. Each output token is a weighted sum of all tokens using the attention values as weights, formulated as:

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}(\mathbf{Q}\mathbf{K}^\top / \sqrt{D})\mathbf{V}, \quad (1)$$

where we can obtain a token-to-token attention map $\mathbf{A}_{t2t} \in \mathbb{R}^{(C+M) \times (C+M)}$ and $\mathbf{A}_{t2t} = \text{softmax}(\mathbf{Q}\mathbf{K}^\top / \sqrt{D})$.

From the global pairwise attention map $\mathbf{A}_{t2t}$, we can extract the class attentions to patches $\mathbf{A}_{c2p} \in \mathbb{R}^{C \times M}$, *i.e.*, class-to-patch attention, where $\mathbf{A}_{c2p} = \mathbf{A}_{t2t}[1 : C, C + 1 : C + M]$, as illustrated by the matrix with yellow dots in Figure 2. Each row represents the attention scores of a specific class to all patches. Leveraging these attention vectors, with the original spatial positions of all patches, can produce C class-relevant localization maps. We can extract class-relevant localization maps from each transformer encoding layer. Given that higher layers learn more high-level discriminative representations (while earlier layers capture more general and low-level visual information), we propose to fuse the class-to-patch attentions from the last K transformer encoding layers, to explore a good trade-off between precision and recall on the generated object localization maps. This process is formulated as:

$$\hat{\mathbf{A}}_{mct} = \frac{1}{K} \sum_{l}^{K} \hat{\mathbf{A}}^l_{mct}, \quad (2)$$

where $\hat{\mathbf{A}}^l_{mct}$ is the class-specific transformer attention extracted from the l^{th} transformer encoding layer of the proposed MCTformer-V1. The fused maps $\hat{\mathbf{A}}_{mct}$ are further normalized by the min-max normalization method along the two spatial dimensions, to generate the final class-specific object localization maps $\mathbf{A}_{mct} \in \mathbb{R}^{C \times N \times N}$. Detailed results on how to choose K can be found in Figure 6.

Class-specific attention refinement. The pairwise affinity is often used in prior works [1, 38, 44] to refine object localization maps. It usually requires an additional network or extra layers to learn an affinity map. In contrast, we propose to extract a pairwise affinity map from the patch-to-patch attention of the proposed MCTformer, without additional computations nor supervision. This is achieved by extracting the patch-to-patch attentions $\mathbf{A}_{p2p} \in \mathbb{R}^{M \times M}$, where $\mathbf{A}_{p2p} = \mathbf{A}_{t2t}[C + 1 : C + M, C + 1 : C + M]$, as illustrated by the matrix with blue dots in Figure 2. The patch-to-patch attentions are reshaped to a 4D tensor $\hat{\mathbf{A}}_{p2p} \in \mathbb{R}^{N \times N \times N \times N}$. The extracted affinity is used to further refine the class-specific transformer attention. This process is formulated as:

$$\mathbf{A}_{mct\_ref}(c, i, j) = \sum_{k}^{N} \sum_{l}^{N} \hat{\mathbf{A}}_{p2p}(i, j, k, l) \cdot \mathbf{A}_{mct}(c, k, l), \quad (3)$$

where $\mathbf{A}_{mct\_ref} \in \mathbb{R}^{C \times N \times N}$ is the refined class-specific localization map. As shown in Table 5 and Figure 5, using the patch-level pairwise affinity for refinement leads to better object localization maps with improved appearance continuity. This was not observed in the prior work [14].

Class-aware training. In contrast to conventional transformers which use the single class token from the last layer to perform classification prediction through a MLP, we have multiple class tokens $\mathbf{T}_{cls} \in \mathbb{R}^{C \times D}$, and we need to ensure that different class tokens can learn different class-discriminative information. To this end, we apply average

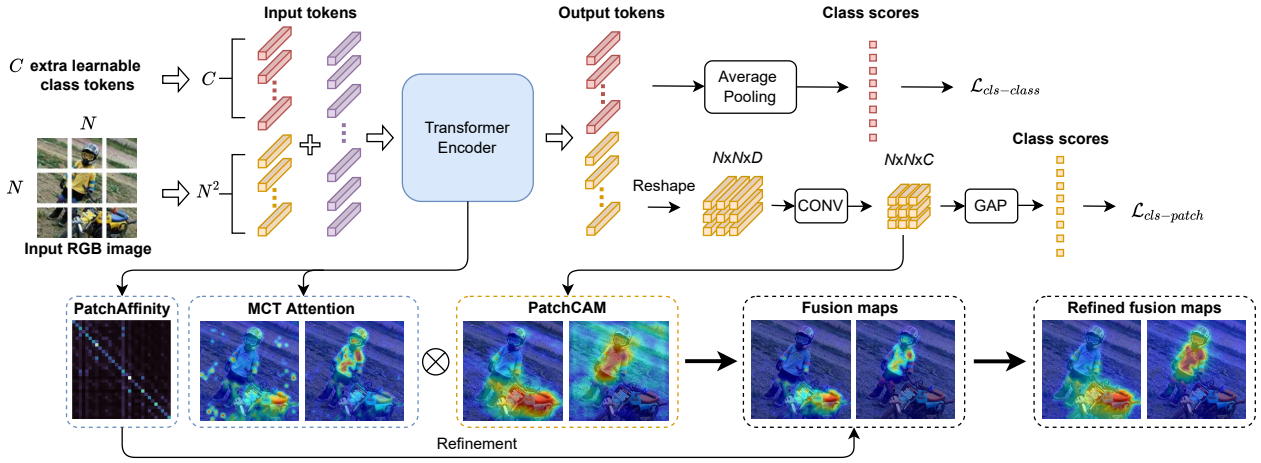

Figure 3. An overview of the proposed MCTformer-V2. We introduce a CAM module into the proposed MCTformer-V1. More specifically, the CAM module is composed of a convolutional layer and a global average pooling (GAP) layer. It takes the reshaped output patch tokens from the last transformer encoding layer 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.

pooling on the output class tokens to produce class scores:

$$\mathbf{y}(c) = \frac{1}{D}\sum_{j}^{D}\mathbf{T}_{cls}(c,j), \tag{4}$$

where $\mathbf{y} \in \mathbb{R}^C$ is the class prediction and $c \in 1, 2, ..., C$. $\mathbf{T}_{cls}(c,j)$ denotes an element in $\mathbf{T}_{cls}$, *i.e.*, the j^{th} feature of the c^{th} class token. We finally compute a multi-label soft margin loss between the class score $\mathbf{y}(c)$ for the class c and its ground-truth label. This provides each class token with strong and direct class-aware supervision, making each class token be able to capture class-specific information.

3.3. Complementarity to Patch-Token CAM

We integrate a CAM module [14, 50, 51] into the proposed multi-class token transformer framework, as shown in Figure 3, constructing an extended model, coined as MCTformer-V2. More specifically, given a sequence of output tokens from the transformer encoder $\mathbf{T}_{out} \in \mathbb{R}^{(C+M)\times D}$, we divide it into the output class tokens $\mathbf{T}_{out\_cls} \in \mathbb{R}^{C\times D}$ and the output patch tokens $\mathbf{T}_{out\_pat} \in \mathbb{R}^{M\times D}$. The patch tokens are then reshaped and forwarded to a convolutional layer with C output channels, producing a 2D feature map $\mathbf{F}_{out\_pat} \in \mathbb{R}^{N\times N\times C}$. $\mathbf{F}_{out\_pat}$ is finally transformed to class predictions through a global average pooling (GAP) layer. In addition, we also use the output class tokens to produce class scores (see Eq. (4)). 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:

$$\mathcal{L}_{total} = \mathcal{L}_{cls-class} + \mathcal{L}_{cls-patch}. \tag{5}$$

Combining PatchCAM and class-specific transformer attention. At test time, patch token-based CAM (thereafter called PatchCAM) maps can be extracted from the last convolutional layer. We extract the PatchCAM maps $\mathbf{A}_{pCAM}$ with $\mathbf{A}_{pCAM} \in \mathbb{R}^{N\times N\times C}$, by applying the min-max normalization on the feature map $\mathbf{F}_{out\_pat}$. The extracted PatchCAM maps are then combined with the proposed class-specific transformer attention maps to produce fused object localization maps $\mathbf{A}$ through an element-wise multiplication operation:

$$\mathbf{A} = \mathbf{A}_{pCAM} \circ \mathbf{A}_{mct}, \tag{6}$$

where $\circ$ denotes the Hadamard product.

Class-specific object localization map refinement. Similar to the attention refinement mechanism proposed in MCTformer-V1 (see Eq. 3), we can also extract the patch-to-patch attention map from MCTformer-V2 as a patch-level pairwise affinity to refine the fused object localization maps as follows:

$$\mathbf{A}_{ref}(c,i,j) = \sum_{k}^{N}\sum_{l}^{N}\hat{\mathbf{A}}_{p2p}(i,j,k,l)\cdot\mathbf{A}(c,k,l). \tag{7}$$

MCTformer-V2 provides an effective transformer-based framework in which the CAM method can flexibly and robustly adapt to multi-label images. 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. The intuition is mainly two-fold. First, this consistency constraint can be regarded as an auxiliary supervision to guide the learning of more effective patch representations. Second, the strong pairwise interaction (*i.e.* message passing) between the patch tokens and the multiple class tokens can also lead to more representative patch tokens, thus pro-

Table 1. Evaluation of the initial seed (Seed) and the corresponding pseudo segmentation ground-truth mask (Mask) in terms of mIoU (%) on the PASCAL VOC *train* set.

Method	Seed	Mask
PSA (CVPR18) [19]	48.0	61.0
Chang *et al*. (CVPR20) [4]	50.9	63.4
SEAM (CVPR20) [39]	55.4	63.6
AdvCAM (CVPR21) [22]	55.6	68.0
CDA (ICCV21) [32]	55.4	63.4
Zhang *et al*. (ICCV21) [48]	57.4	67.8
MCTformer (Ours)	**61.7**	**69.1**

Table 2. Performance comparison of WSSS methods in terms of mIoU (%) on the PASCAL VOC 2012 *val* and *test* sets using different segmentation backbones. Sup.: supervision. I: image-level ground-truth labels. S: off-the-shelf saliency maps.

Method	Backbone	Sup.	Val	Test
CIAN (AAAI20) [13]	ResNet101	I+S	64.3	65.3
ICD (CVPR20) [12]	ResNet101	I+S	67.8	68.0
Zhang *et al*. (ECCV20) [49]	ResNet50	I+S	66.6	66.7
Sun *et al*. (ECCV20) [33]	ResNet101	I+S	66.2	66.9
EDAM (CVPR21) [42]	ResNet101	I+S	70.9	70.6
EPS (CVPR21) [23]	ResNet101	**I+S**	**71.0**	**71.8**
Yao *et al*. (CVPR21) [45]	ResNet101	I+S	68.3	68.5
AuxSegNet (ICCV21) [44]	ResNet38	I+S	69.0	68.6
Zhang *et al*. (AAAI20) [46]	ResNet38	I	62.6	62.9
Luo *et al*. (AAAI20) [28]	ResNet101	I	64.5	64.6
Chang *et al*. (CVPR20) [4]	ResNet101	I	66.1	65.9
Araslanov *et al*. (CVPR20) [2]	ResNet38	I	62.7	64.3
SEAM (CVPR20) [39]	ResNet38	I	64.5	65.7
BES (ECCV20) [5]	ResNet101	I	65.7	66.6
CONTA (NeurIPS20) [47]	ResNet38	I	66.1	66.7
AdvCAM (CVPR21) [22]	ResNet101	I	68.1	68.0
ECS-Net (ICCV21) [34]	ResNet38	I	66.6	67.6
Kweon *et al*. (ICCV21) [20]	ResNet38	I	68.4	68.2
CDA (ICCV21) [32]	ResNet38	I	66.1	66.8
Zhang *et al*. (ICCV21) [48]	ResNet38	I	67.8	68.5
MCTformer (Ours)	ResNet38	**I**	**71.9**	**71.6**

ducing more class-discriminative PatchCAM maps, compared to only using one class token as in TS-CAM [14].

4. Experiments

4.1. Experimental Settings

Datasets. We evaluated the proposed approach on two datasets, *i.e*., PASCAL VOC 2012 [11] and MS COCO 2014 [26]. **PASCAL VOC** has three subsets, *i.e*., training (train), validation (val) and test sets, each containing 1,464, 1,449, and 1,456 images, respectively. It has 20 object classes and one background class for the semantic segmentation task. Following prior works [4, 22, 32, 39, 44, 48], an augmented set of 10,582 images, with additional data from [15], was used for training. **MS COCO** uses 80 object classes and one background class for semantic segmentation. Its training and validation sets contain 80K and 40K images, respectively. Note that we only used *image-level* ground-truth labels from these datasets during training.

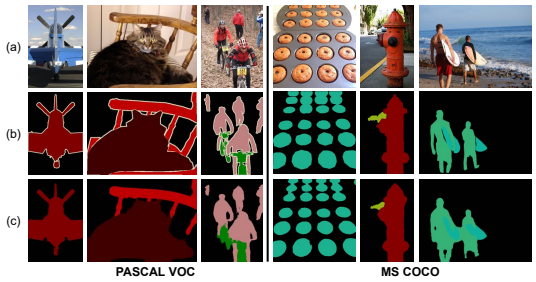

Figure 4. Qualitative segmentation results on the PASCAL VOC and MS COCO *val* sets. (a) Input. (b) Ground-truth. (c) Ours.

Evaluation metrics. In line with previous works [22], we use the mean Intersection-over-Union (mIoU) to evaluate the semantic segmentation performance on the *val* set, of the two benchmarks. We obtained the semantic segmentation results on the PASCAL VOC *test* set from the official PASCAL VOC online evaluation server.

Implementation details. We built the proposed MCTformer using the DeiT-S Backbone [14, 36] pre-trained on ImageNet [9]. More specifically, we used the pre-trained class token in DeiT-S to initialize the proposed multiple class tokens. We followed the data augmentation and default training parameters provided in [14, 36]. Training images are resized to 256×256 and then cropped into 224×224. For semantic segmentation, we followed prior works [1, 44, 46, 48] to use the ResNet38 [43] based Deeplab V1. At test time, we used multi-scale testing and CRFs with the hyper-parameters suggested in [6] for post-processing.

4.2. Comparison with State-of-the-arts

PASCAL VOC. We followed the common practice [4, 22, 32, 39, 48] to apply PSA [1] on the proposed object localization maps (seed) to generate pseudo semantic segmentation ground-truth labels (mask) on the *train* set. As shown in Table 1, the proposed method performs better than existing works by large margins on both the initial seed and the pseudo ground-truth mask, better than the best initial seed [48] by 4.3%. Table 2 shows that the proposed MCTformer achieves segmentation results (mIoUs) of 71.9% and 71.6% on the *val* and *test* sets, respectively. The proposed MCTformer performs significantly better than all the existing methods using only image-level labels. In particular, MCTformer can even achieve comparable or better results compared to the methods using additional saliency maps. Figure 4 (left) shows that the segmentation model trained with our pseudo labels can produce accurate and complete object contours in various challenging scenes.

MS COCO. Table 3 shows that the proposed method achieves a segmentation mIoU of 42.0%, surpassing the recent methods by a large margin. Notably, from Table 3, we observe that several methods using additional saliency maps obtain inferior performance compared to recent meth-

Table 3. Performance comparison of WSSS methods in terms of mIoU(%) on the MS COCO *val* set.

Method	Backbone	Sup.	Val
EPS (CVPR21) [23]	ResNet101	**I+S**	**35.7**
AuxSegNet (ICCV21) [44]	ResNet38	I+S	33.9
Wang *et al*. (IJCV20) [38]	VGG16	I	27.7
Luo *et al*. (AAAI20) [28]	VGG16	I	29.9
SEAM (CVPR20) [39]	ResNet38	I	31.9
CONTA (NeurIPS20) [47]	ResNet38	I	32.8
Kweon *et al*. (ICCV21) [20]	ResNet38	I	36.4
CDA (ICCV21) [32]	ResNet38	I	33.2
MCTformer (Ours)	ResNet38	**I**	**42.0**

Table 4. Complexity of models generating object localization maps. The proposed MCTformer is based on DeiT-S [36].

Model	Image size	#Params (M)	MACs (G)
ResNet38	224×224	104.3	99.8
MCTformer-V1	224×224	21.7	4.6
MCTformer-V2	224×224	21.8	4.7

ods only using image-level labels. This reveals the limitation of relying on pre-trained saliency models, which may perform poorly on challenging datasets. Several qualitative segmentation results are shown in Figure 4 (right).

Model complexity. We compared the model complexity of the proposed MCTformer with a commonly used CNN model for generating object localization maps [1, 39, 48], *i.e*., ResNet38 [43], in terms of the number of parameters and multiply-add calculations (MACs). Table 4 shows that the complexity of the proposed DeiT-S [36] based method is significantly smaller than that of ResNet38 based methods.

4.3. Ablation Studies

Effect of multiple class-specific tokens. In the conventional ViT, the class token attention only indicates a class-agnostic localization map. TS-CAM [14] applies CAM on the output patch tokens of ViT to obtain class-specific localization maps. Following their official implementation, the generated object localization maps by TS-CAM on the PASCAL VOC *train* set obtained an mIoU of 29.9%, as shown in Table 5. We simply added a ReLU layer on their PatchCAM maps (*i.e*., TS-CAM*), yielding a large improvement of 11.4%. In comparison, the proposed baseline method, *i.e.,* the class-specific transformer attention maps of the multiple class-specific tokens in the proposed MCTformer-V1, attains an mIoU of 47.2%, outperforming TS-CAM* by a significant margin of 5.9%. This demonstrates the effectiveness of the proposed transformer attention based class-specific localization maps.

Complementarity of PatchCAM and the proposed class-specific transformer attention. Table 5 shows that the object localization maps generated by MCTformer-V2 with a standard CAM module, obtain an mIoU of 58.2%. This can be further improved to 61.7% by using the patch-level pairwise affinity for refinement. As shown in Figure 5e, the class-specific transformer attention can effectively localize objects while with low responses and noises. In contrast, the PatchCAM maps (Figure 5f) show high responses on object regions, while they also have more background pixels around the objects activated. The fusion of these two leads to clearly improved localization maps which only activate object regions, with significantly reduced background noises (Figure 5g). These class-specific localization maps confirm remarkably superior performance of our proposed model compared to TS-CAM [14] (Figure 5b) that shows sparse and low object responses in most cases.

Table 5. Evaluation of different object localization maps in terms of mIoU(%) on the PASCAL VOC *train* set.

Method	mIoU
TS-CAM [14]	29.9
TS-CAM* [14]	41.3
MCTformer-V1 (Attention)	47.2
MCTformer-V1 (Attention + PatchAffinity)	55.2
MCTformer-V2 (Attention + PatchCAM)	58.2
MCTformer-V2 (Attention + PatchCAM + PatchAffinity)	**61.7**

Table 6. Segmentation results using different object localization maps in terms of mIoU(%) on the PASCAL VOC *val* set.

Method	mIoU
TS-CAM* [14]	49.7
MCTformer-V1 (Attention)	55.6
MCTformer-V1 (Attention + PatchAffinity)	58.8
MCTformer-V2 (Attention + PatchCAM)	61.1
MCTformer-V2 (Attention + PatchCAM + PatchAffinity)	**62.6**

Table 7. Comparison of different methods for class prediction in MCTformer-V1 on the generated class-specific transformer attention in terms of mIoU(%) on the PASCAL VOC *train* set.

	Fully-connected	Max-pooling	Average-pooling
mIoU	41.5	26.8	**47.2**

Effect of patch affinity. As shown in Table 5 and Table 6, by applying the learned patch-to-patch attention as a patch-level pairwise affinity to refine the object localization maps from MCTformer-V1, the pseudo segmentation label maps can be improved by 8%, and accordingly, the segmentation performance is also improved by a gain of 3.2%. MCTformer-V2 yields consistent improvements in terms of the quality of generated pseudo labels and the segmentation performance, compared to the variants that do not use patch affinity. The visualization results in Figure 5 (d) and (h) show that the refined object localization maps appear more complete with smoother object contours. This further demonstrates the great benefits of our method in generating an effective patch affinity without additional computation.

Different class prediction methods. We evaluated the effect of the different strategies used to produce class scores on the generated class-specific transformer attention maps.

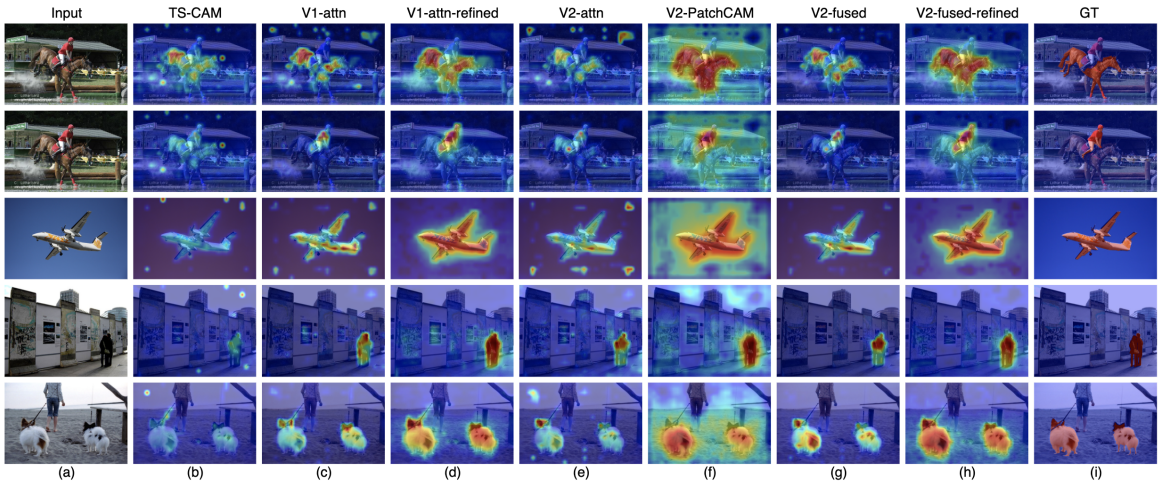

Figure 5. Visualization examples of different object localization maps from different methods: (b) TS-CAM [14]; (c) V1-attn (the class-specific transformer attention from our MCTformer-V1); (d) V1-attn-refined (the refined class-specific transformer attention by the patch affinity from MCTformer-V1); (e) V2-attn (the class-specific transformer attention from MCTformer-V2); (f) V2-PatchCAM (the PatchCAM maps from MCTformer-V2). (g) V2-fused (the fused map of the class-specific transformer attention and the PatchCAM map from MCTformer-V2); (h) V2-fused-refined (the refined fusion map by the patch affinity from MCTformer-V2). (i) Ground-truth.

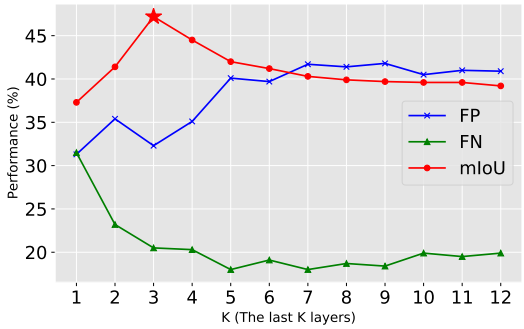

Figure 6. Evaluation of object localization maps generated by fusing the class token attentions from the last K transformer layers using false positive rate (FP), false negative rate (FN) and mIoU.

As shown in Table 7, max pooling has the worst performance for class-specific localization with an mIoU of only 26.8%, while using a fully connected layer for a linear projection yields an improved mIoU of 41.5%. The average pooling produces the best performance with an mIoU of 47.2%. These results confirm our initial design motivation. Specifically, involving extra parameters within a fully-connected layer may increase the complexity of learning the model for discriminative localization. In contrast to the max pooling which only needs to attend to the most relevant patch, the average pooling can encourage class tokens to attend to more relevant patches, which is beneficial to learn better spatial context for localization.

Number of layers for attention fusion. We evaluated the quality of object localization maps by fusing attention maps of different class tokens from multiple transformer encoding layers in the proposed MCTformer-V1. Following [39], we use three evaluation metrics, *i.e.*, false positive rate (FP), false negative rate (FN), and mIoU, in which larger FP and FN values indicate increased over-activated and under-activated regions, respectively. As shown in Figure 6, aggregating information from more layers produces object localization maps which tend to be over-activated. This indicates that the early layers produce more generic low-level representations which may not be very helpful for high-level semantic localization. By decreasing the number of layers, the generated object localization maps become more discriminative at the cost of a lower activation coverage. The results reported in Figure 6 suggest that fusing the attentions from the last three layers can yield the best pseudo segmentation ground-truth labels (mIoU of 47.2%).

5. Conclusions

This paper presents MCTformer, a simple yet effective transformer-based framework to produce class-specific object localization maps, and achieves state-of-the-art results on WSSS. We show that class-to-patch attention of different class tokens can discover class-specific localization information, while patch-to-patch attention can also learn effective pairwise affinities to refine the localization maps. Furthermore, we demonstrate that the proposed framework can seamlessly complement the CAM mechanism, leading to high-quality pseudo ground-truth labels for WSSS. Future work will extend the proposed method to more downstream tasks, such as weakly supervised object detection and instance segmentation.

Acknowledgment. This research is supported in part by Australian Research Council Grant DP210101682, DP210102674, DP200103223, Australian Medical Research Future Fund MRFAI000085, CRC-P Smart Material Recovery Facility (SMRF) - Curby Soft Plastics, the Early Career Scheme of the Research Grants Council (RGC) of the Hong Kong SAR under grant No. 26202321 and HKUST Startup Fund No. R9253.

References

[1] Jiwoon Ahn and Suha Kwak. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *CVPR*, 2018. 3, 4, 6, 7

[2] Nikita Araslanov and Stefan Roth. Single-stage semantic segmentation from image labels. In *CVPR*, 2020. 6

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *ICCV*, 2021. 1, 3

[4] Yu-Ting Chang, Qiaosong Wang, Wei-Chih Hung, Robinson Piramuthu, Yi-Hsuan Tsai, and Ming-Hsuan Yang. Weakly-supervised semantic segmentation via sub-category exploration. In *CVPR*, 2020. 2, 6

[5] Liyi Chen, Weiwei Wu, Chenchen Fu, Xiao Han, and Yuntao Zhang. Weakly supervised semantic segmentation with boundary exploration. In *ECCV*, 2020. 6

[6] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Semantic image segmentation with deep convolutional nets and fully connected crfs. In *ICLR*, 2015. 6

[7] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. In *NeurIPS*, 2021. 1

[8] Xiangxiang Chu, Zhi Tian, Bo Zhang, Xinlong Wang, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Conditional positional encodings for vision transformers. *arXiv preprint arXiv:2102.10882*, 2021. 1

[9] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 6

[10] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1, 3, 4

[11] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010. 6

[12] Junsong Fan, Zhaoxiang Zhang, Chunfeng Song, and Tieniu Tan. Learning integral objects with intra-class discriminator for weakly-supervised semantic segmentation. In *CVPR*, 2020. 6

[13] Junsong Fan, Zhaoxiang Zhang, Tieniu Tan, Chunfeng Song, and Jun Xiao. Cian: Cross-image affinity net for weakly supervised semantic segmentation. In *AAAI*, 2020. 2, 6

[14] Wei Gao, Fang Wan, Xingjia Pan, Zhiliang Peng, Qi Tian, Zhenjun Han, Bolei Zhou, and Qixiang Ye. Ts-cam: Token semantic coupled attention map for weakly supervised object localization. In *ICCV*, 2021. 3, 4, 5, 6, 7, 8

[15] Bharath Hariharan, Pablo Arbeláez, Lubomir Bourdev, Subhransu Maji, and Jitendra Malik. Semantic contours from inverse detectors. In *ICCV*, 2011. 6

[16] Qibin Hou, PengTao Jiang, Yunchao Wei, and Ming-Ming Cheng. Self-erasing network for integral object attention. In *NeurIPS*, 2018. 2

[17] Tsung-Wei Ke, Jyh-Jing Hwang, and Stella X Yu. Universal weakly supervised segmentation by pixel-to-segment contrastive learning. In *ICLR*, 2021. 2

[18] Salman Khan, Muzammal Naseer, Munawar Hayat, Syed Waqas Zamir, Fahad Shahbaz Khan, and Mubarak Shah. Transformers in vision: A survey. *arXiv preprint arXiv:2101.01169*, 2021. 1, 3

[19] Alexander Kolesnikov and Christoph H Lampert. Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In *ECCV*, 2016. 2, 6

[20] Hyeokjun Kweon, Sung-Hoon Yoon, Hyeonseong Kim, Daehee Park, and Kuk-Jin Yoon. Unlocking the potential of ordinary classifier: Class-specific adversarial erasing framework for weakly supervised semantic segmentation. In *ICCV*, 2021. 6, 7

[21] Jungbeom Lee, Eunji Kim, Sungmin Lee, Jangho Lee, and Sungroh Yoon. Ficklenet: Weakly and semi-supervised segmentation using stochastic inference. In *CVPR*, 2019. 2

[22] Jungbeom Lee, Eunji Kim, and Sungroh Yoon. Anti-adversarially manipulated attributions for weakly and semi-supervised semantic segmentation. In *CVPR*, 2021. 6

[23] Seungho Lee, Minhyun Lee, Jongwuk Lee, and Hyunjung Shim. Railroad is not a train: Saliency as pseudo-pixel supervision for weakly supervised semantic segmentation. In *CVPR*, 2021. 6, 7

[24] Kunpeng Li, Ziyan Wu, Kuan-Chuan Peng, Jan Ernst, and Yun Fu. Tell me where to look: Guided attention inference network. In *CVPR*, 2018. 2

[25] Xueyi Li, Tianfei Zhou, Jianwu Li, Yi Zhou, and Zhaoxiang Zhang. Group-wise semantic mining for weakly supervised semantic segmentation. In *AAAI*, 2021. 2

[26] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, 2014. 6

[27] Nian Liu, Ni Zhang, Kaiyuan Wan, Ling Shao, and Junwei Han. Visual saliency transformer. In *ICCV*, 2021. 3

[28] Wenfeng Luo and Meng Yang. Learning saliency-free model with generic features for weakly-supervised semantic segmentation. In *AAAI*, 2020. 6, 7

[29] Zizheng Pan, Bohan Zhuang, Jing Liu, Haoyu He, and Jianfei Cai. Scalable vision transformers with hierarchical pooling. In *ICCV*, 2021. 1

[30] René Ranftl, Alexey Bochkovskiy, and Vladlen Koltun. Vision transformers for dense prediction. In *ICCV*, 2021. 3

[31] Krishna Kumar Singh and Yong Jae Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *ICCV*, 2017. 2

[32] Yukun Su, Ruizhou Sun, Guosheng Lin, and Qingyao Wu. Context decoupling augmentation for weakly supervised semantic segmentation. In *ICCV*, 2021. 6, 7

[33] Guolei Sun, Wenguan Wang, Jifeng Dai, and Luc Van Gool. Mining cross-image semantics for weakly supervised semantic segmentation. In *ECCV*, 2020. 2, 6

[34] Kunyang Sun, Haoqing Shi, Zhengming Zhang, and Yongming Huang. Ecs-net: Improving weakly supervised semantic segmentation by using connections between class activation maps. In *ICCV*, 2021. 6

[35] Meng Tang, Federico Perazzi, Abdelaziz Djelouah, Ismail Ben Ayed, Christopher Schroers, and Yuri Boykov. On regularized losses for weakly-supervised cnn segmentation. In *ECCV*, 2018. 2

[36] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021. 6, 7

[37] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017. 3, 4

[38] Xiang Wang, Sifei Liu, Huimin Ma, and Ming-Hsuan Yang. Weakly-supervised semantic segmentation by iterative affinity learning. *IJCV*, 128(6):1736–1749, 2020. 3, 4, 7

[39] Yude Wang, Jie Zhang, Meina Kan, Shiguang Shan, and Xilin Chen. Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In *CVPR*, 2020. 2, 3, 6, 7, 8

[40] Yunchao Wei, Jiashi Feng, Xiaodan Liang, Ming-Ming Cheng, Yao Zhao, and Shuicheng Yan. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *CVPR*, 2017. 2

[41] Yunchao Wei, Huaxin Xiao, Honghui Shi, Zequn Jie, Jiashi Feng, and Thomas S Huang. Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation. In *CVPR*, 2018. 2

[42] Tong Wu, Junshi Huang, Guangyu Gao, Xiaoming Wei, Xiaolin Wei, Xuan Luo, and Chi Harold Liu. Embedded discriminative attention mechanism for weakly supervised semantic segmentation. In *CVPR*, 2021. 6

[43] Zifeng Wu, Chunhua Shen, and Anton Van Den Hengel. Wider or deeper: Revisiting the resnet model for visual recognition. *Pattern Recognition*, 90:119–133, 2019. 6, 7

[44] Lian Xu, Wanli Ouyang, Mohammed Bennamoun, Farid Boussaid, Ferdous Sohel, and Dan Xu. Leveraging auxiliary tasks with affinity learning for weakly supervised semantic segmentation. In *ICCV*, 2021. 3, 4, 6, 7

[45] Yazhou Yao, Tao Chen, Guo-Sen Xie, Chuanyi Zhang, Fumin Shen, Qi Wu, Zhenmin Tang, and Jian Zhang. Non-salient region object mining for weakly supervised semantic segmentation. In *CVPR*, 2021. 6

[46] Bingfeng Zhang, Jimin Xiao, Yunchao Wei, Mingjie Sun, and Kaizhu Huang. Reliability does matter: An end-to-end weakly supervised semantic segmentation approach. In *AAAI*, 2020. 2, 6

[47] Dong Zhang, Hanwang Zhang, Jinhui Tang, Xiansheng Hua, and Qianru Sun. Causal intervention for weakly-supervised semantic segmentation. In *NeurIPS*, 2020. 6, 7

[48] Fei Zhang, Chaochen Gu, Chenyue Zhang, and Yuchao Dai. Complementary patch for weakly supervised semantic segmentation. In *ICCV*, 2021. 2, 3, 6, 7

[49] Tianyi Zhang, Guosheng Lin, Weide Liu, Jianfei Cai, and Alex Kot. Splitting vs. merging: Mining object regions with discrepancy and intersection loss for weakly supervised semantic segmentation. In *ECCV*, 2020. 2, 6

[50] Xiaolin Zhang, Yunchao Wei, Jiashi Feng, Yi Yang, and Thomas S Huang. Adversarial complementary learning for weakly supervised object localization. In *CVPR*, 2018. 5

[51] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016. 1, 2, 5