Max Pooling with Vision Transformers reconciles class and shape in weakly supervised semantic segmentation

Simone Rossetti[1,2], Damiano Zappia[1], Marta Sanzari[2], Marco Schaerf[2,1], Fiora Pirri[1,2]

[1] DeepPlants, @deepplants.com
[2] DIAG, Sapienza @diag.uniroma1.it

Abstract. Weakly Supervised Semantic Segmentation (WSSS) research has explored many directions to improve the typical pipeline CNN plus class activation maps (CAM) plus refinements, given the image-class label as the only supervision. Though the gap with the fully supervised methods is reduced, further abating the spread seems unlikely within this framework. On the other hand, WSSS methods based on Vision Transformers (ViT) have not yet explored valid alternatives to CAM. ViT features have been shown to retain a scene layout, and object boundaries in self-supervised learning. To confirm these findings, we prove that the advantages of transformers in self-supervised methods are further strengthened by Global Max Pooling (GMP), which can leverage patch features to negotiate pixel-label probability with class probability. This work proposes a new WSSS method dubbed ViT-PCM (ViT Patch-Class Mapping), not based on CAM. The end-to-end presented network learns with a single optimization process, refined shape and proper localization for segmentation masks. Our model outperforms the state-of-the-art on baseline pseudo-masks (BPM), where we achieve 69.3% mIoU on PascalVOC 2012 *val* set. We show that our approach has the least set of parameters, though obtaining higher accuracy than all other approaches. In a sentence, quantitative and qualitative results of our method reveal that ViT-PCM is an excellent alternative to CNN-CAM based architectures.

Keywords: weakly-supervised semantic segmentation, Vision Transformers, Global Max Pooling, Image class-labels supervision

1 Introduction

Weakly supervised semantic segmentation (WSSS) is about segmenting object classes with no pixel-label supervision and using the less demanding supervision possible. The most economic supervision is via image-level class labels, out of which a WSSS method computes pseudo-masks for each object class in an image. To test a WSSS method accuracy, a supervised segmentation network, such as DeepLab [7], is trained on the devised pseudo-masks, and the induced accuracy is compared with the fully supervised methods. The segmentation task, supervised

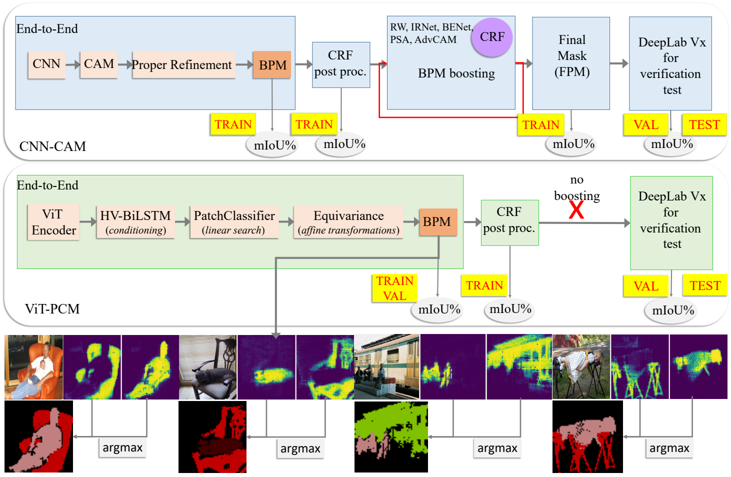

Fig. 1: The figure compares the basic structure of a CNN-CAM method, above in light blue, with our proposed ViT-PCM method, below in light green. ViT-PCM learns to estimate the BPM, shown in the last two strips, with a single optimization. Our BPM are then refined with a CRF (see Figure 4) and, without further processing, are passed to the verification task (DeepLab). Differently from ViT-PCM, a CAM-based method demands a multi-stage optimization. All recent approaches require boosting the BPM, improved by the CRF, before passing them to the verification task.

by the pseudo-mask labels, is a *verification task* aiming at demonstrating the computed pseudo-mask quality. In principle, the verification task adds equal improvement to all methods.

So far, methods based on image-level class labels generate pseudo-mask using class activation maps (CAM) [62]. CAM are obtained from a multi-label classification network, such as a CNN.

CAM limitations in estimating both shape and localization of the classes of interest [12,20,4,50] induce many researchers to resort to extra refinements between the baseline pseudo-masks (BPM), often called *seeds*, and the final pseudo-masks production for test verification. These refinements mostly often bring into play multi-stage architectures, as noted in PAMR [3]. Several authors resort to saliency maps as subsidiary supervision for good localization [52,32,57,47,63]. Other authors adopt image operations such as region erasing [42,51], or region growing to expand the seed region during training [26,23], and multi-scale map fusion to improve background and foreground [53]. Jang *et Al.* [25] reviewed the feature layers selection for CAMs using attribute propagation methods [35]. Sun *et Al.* [45] estimate the similarity of the foreground features of the same class with two co-attention networks to capture better the context and [19] look into relations across different images.

Yet, the greatest success in refinement strategies has been earned by IRNet [1], PSA [2], and AdvCAM [30]. Also, PAC [44] and BENet [9], have been recently

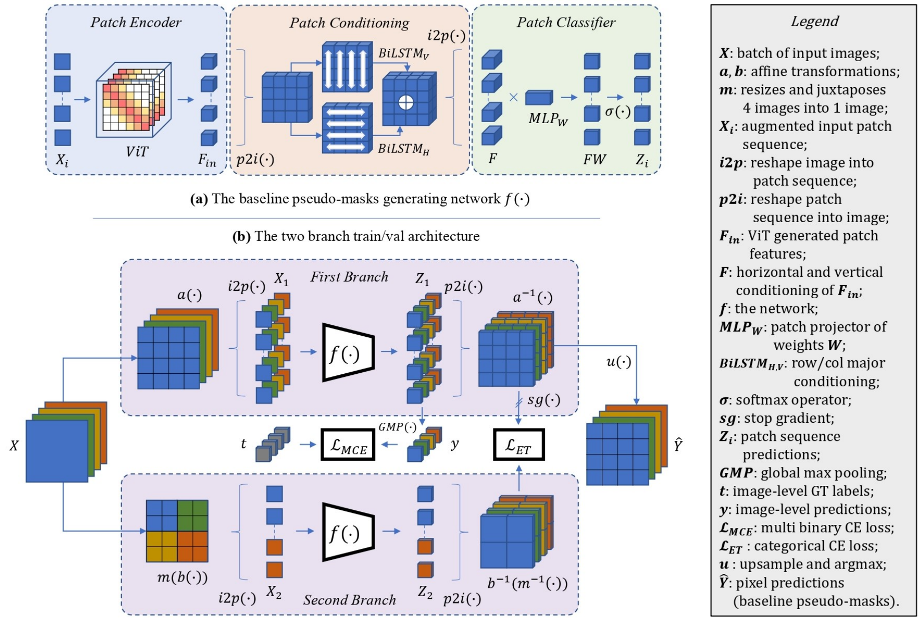
Patch Encoder
X_i
ViT
F_{in}
Patch Conditioning
$i2p(\cdot)$
$BiLSTM_V$
$BiLSTM_H$
$p2i(\cdot)$
Patch Classifier
F
MLP_W
FW
$\sigma(\cdot)$
Z_i
(a) The baseline pseudo-masks generating network $f(\cdot)$
(b) The two branch train/val architecture
First Branch
$a(\cdot)$
$i2p(\cdot)$
X_1
$f(\cdot)$
Z_1
$p2i(\cdot)$
$a^{-1}(\cdot)$
$u(\cdot)$
$sg(\cdot)$
X
t
$\mathcal{L}_{MCE}$
$GMP(\cdot)$
y
$\mathcal{L}_{ET}$
$\hat{Y}$
$m(b(\cdot))$
$i2p(\cdot)$
X_2
Second Branch
Z_2
$p2i(\cdot)$
$b^{-1}(m^{-1}(\cdot))$
Legend
X: batch of input images;
a, b: affine transformations;
m: resizes and juxtaposes 4 images into 1 image;
X_i: augmented input patch sequence;
$i2p$: reshape image into patch sequence;
$p2i$: reshape patch sequence into image;
F_{in}: ViT generated patch features;
F: horizontal and vertical conditioning of F_{in};
f: the network;
MLP_W: patch projector of weights W;
$BiLSTM_{H,V}$: row/col major conditioning;
σ: softmax operator;
sg: stop gradient;
Z_i: patch sequence predictions;
GMP: global max pooling;
t: image-level GT labels;
y: image-level predictions;
$\mathcal{L}_{MCE}$: multi binary CE loss;
$\mathcal{L}_{ET}$: categorical CE loss;
u : upsample and argmax;
$\hat{Y}$: pixel predictions (baseline pseudo-masks).

Fig. 2: The above schema shows the end-to-end ViT-PCM, a semantic segmentation method supervised by image-level class labels t. The plate in (a) shows the core network $f(\cdot)$ implementing the *linear search method*, which maps the image-level class labels to patch-labels. The plate (b) shows the two-branches architecture, including $f(\cdot)$ in both branches.

used. For example, SEAM [50], Chang *et Al.* [6] and [41] use PSA; CONTA [14] and ReCAM [12] use IRNet while [30] using both. AFA [39] use PAC [44].

CRF[27] are trained on PascalVOC, fully supervised, and introduced in WSSS by [26]. CRF used as post-processing out of a training loop, improve the BPM, on average, 3-4% mIoU, on Pascal VOC 2012. On the other hand, multi-stage methods, refining BPM with IRNet [1], PSA [2], and AdvCAM [30] use dense CRF in the training loop, which gives a substantial boost in accuracy. Using dense CRF, optimized on PascalVOC, likewise using saliency (e.g. [22], which operates dense CRF too) in the refinement loop to obtain the final pseudo-mask, beside being resource intensive, fails to generalize a method beyond the PascalVOC dataset. This lack of generalization power is common to any WSSS approach using biased methods in a refinement training loop.

The challenge is to raise the bar of the baseline pseudo-mask accuracy so that the only supervision truly sticks to the image-level label. To this end, we introduce a new model for computing pseudo-masks, which bypasses the CAM bottleneck. The main contribution of the paper are the followings:

- We introduce a novel model for weakly supervised semantic segmentation (WSSS) based on ViT [15]. The model, dubbed ViT-PCM, is represented in Figure 2.
- We propose a new pseudo-mask computation method *Explicit Search* without resorting to CAM. The method leverages the locality properties of ViT to come close to an effective mapping between multi-label classification and semantic segmentation. We use the Global Max Pooling (GMP) to fetch the relevance of each patch, given the patches' categorical distribution over the classes of interest. This way, we project the patch features to class predictions (PCM) using a multi-label BCE loss (MCE). We ensure equivariance to translation and scaling transformations defining two branches, see Figure 2.
- The proposed pseudo-mask computation outperforms all state-of-the-art methods: we obtain BPM accuracy of 67.7 mIoU% on Pascal VOC 2012 *train* set which improves the current best BPM ([39]) of 3.91% . On average, we improve more than 5% mIoU than all the other competitors. On MS-COCO 2014 we obtain 45.03% mIoU on *val* set.
- For the verification task, using DeepLab as a segmentation method, we do not need to boost our BPM to obtain masks more suitable for DeepLab, yet we obtain comparable validation and test scores.
- We also prove the advantages of our method in terms of computational effort. In particular, we obtain the final segmentation with 89.4 M of parameter size, the minimal cost amid competitors.

Beyond the novelty of our contribution, which is the first proposal to compute pseudo-mask baselines bypassing CAM, we show that both quantitative and qualitative results prove that exploring new methods for baseline pseudo-masks can be rewarding. We establish a new state of art on baseline pseudo-mask computation, using image-level class labels without refinement.

2 Related Works

Current WSSS methods mostly operate with image-level class labels as the cheapest supervision. Approaches using image-level class labels are based so far on CAM [62] methods using a plain multi-label classification network. The class activation maps are obtained via the global average pooling (GAP) averaging the feature maps of the last layer, further concatenated into a weights vector. This last is connected with the class prediction, using a BCE prediction loss. More recently, Vision Transformers [15] are emerging as an alternative to generate CAM [58,39]. Our method is the first one using only ViT without CAM to generate baseline pseudo-masks.

CNN plus CAM. These methods contribute to two complementary research directions: *Baseline Pseudo-Mask generation*, to control and expand the activation of CAM regions, and *Pseudo-Mask refinement* to obtain the full mask of objects.

Baseline Pseudo-Mask generation extends CAM by revising the loss, or by augmenting the dataset, or by perturbing CAM devised regions, or using pretrained saliency maps. In ReCAM [12] the authors propose softmax cross-entropy (SCE) as a valid solution for CAM, since it bypasses the non-exclusive class problem of BCE. In OoD [31] the authors propose an out of distribution dataset taken from OpenImages [28], to better capture background semantics. Other methods to expand CAM perturb the generated regions to capture new areas [29,43,30], by either erasing or masking. Since [52], pretrained methods for saliency detection and saliency maps have been adopted in [36,61,32,59,54,57], and in [25,24]. The latter propose an online attention accumulation (OAA) strategy based on attribute propagation methods. Pseudo-mask generation is contaminated by self-supervised learning in [50], via downstream tasks and transformations ensuring CAM features equivariance, or via contrastive representation learning, as in RCA [63], C2AM [56] and PPC [16].

Pseudo-Mask refinement. In recent works, all CAM-based approaches explore refinement strategies, ensuring some control on pixel-level labelling. The most common strategies are PSA [2], AdvCAM [30] and IRNet [1]. PSA refines the baseline masks by propagating pixel semantic values to their neighbours, collecting confidence for the target classes. AdvCAM [30] uses iterative adversarial climbing performed on an image to iteratively involve its features in the classification to increase CAM confidence in activated regions. IRNet [1] explores class equivalence relations of pixels and refines pixel-labels by evaluating the displacement w.r.t. computed centroids. Recently BENet [9] has been used for pseudo-mask baseline refinement, too; it refines object boundaries, together with foreground and background. We observed in the introduction that all these strategies use in the training loop dense CRF of [27], which is trained on PascalVOC2012.

Transformers. ViT have so far gathered a significant success with self-supervised learning [15], as witnesses Dino [5], [11,33], and recently SDMP [37]. Dino [5] downstream task segments foreground from background for single class images, differing from WSSS. Only recently ViT contributed to WSSS with MCTformer [58] and AFA [39], though both resort to CAM. MCTformer exploits ViT attention mechanism to obtain localization maps. To generate pseudo-masks, they resort to PSA [2]. AFA uses ViT multi-head-self-attention (MSA) to capture global dependencies and develop an affinity-based attention module to propagate the initial pseudo-masks, namely the obtained CAM. Refinement of the initial pseudo-mask is attained by affinity propagation with RAWK [49], in turn, pretrained on a scribble dataset.

Differently from the above approaches, we use ViT as a backbone for building our explicit search method. Indeed, we devise an end-to-end internal refinement to obtain a baseline pseudo-mask (BMP) without resorting to external strategies.

3 Motivations of using ViT and bypass CAM

At the core of semantic segmentation, supervised by image-level labels, is the mapping between multilabel image classification and pixel-level classification. This mapping requires linking the abstract image feature space, encoding classes into an index vector, to a completely different space in which features encode

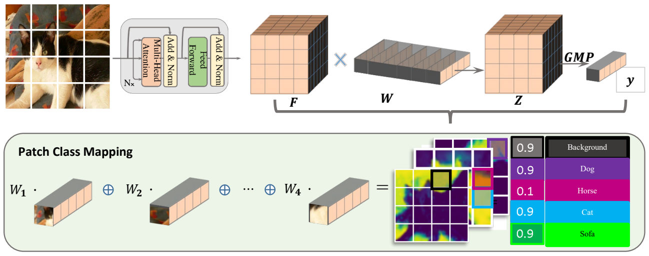

Fig. 3: Patch Class Mapping.

classes into a fine grid structure. How could this be possible? CNN have an inductive bias on the image features local structure because of convolution kernels, which CAM leverages. The inductive bias of CNN entitles CAM to indicate the pixels which mainly contribute to the specific class prediction. The produced map is appealing though misleading: it does not induce a mapping between image features and pixels.

On the other hand, ViT [15] have much less bias because images are split into flattened patches and encoded. Thus, the spatial relations are learned from scratch using attention and position embedding. This learning from a *tabula rasa* generates a number of basis functions for each patch, specifying their internal structure. These basis functions account implicitly for the class a patch belongs to. On these grounds, the mapping problem amounts to unravelling the implicit class representation brought on by the patch principal components. Our proposed *explicit search method* models this mapping.

We describe here the intuition. Let us assume that patches are pixels, the classes (categories) are denoted by $\mathcal{C}$, having cardinality K and $X\in\mathbb{R}^{h\times w\times 3}$ is an image. Let also assume that the ViT inferring the image multiclass labels is the function $f(\cdot|\varphi)$ with parameters φ, mapping an image X to a vector of values in $(0, 1)$ for each category in $\mathcal{C}$. On the other hand, let us represent the basis functions specifying the patches' internal structure, implicitly accounting for the patch classes, by a tensor Z. We shall see below how Z is computed. Z has height and width as the image X, and it also has a third axis for the categories $\mathcal{C}$. We make Z a stochastic tensor along the categories axis: summing up along that axis, we obtain a matrix of ones. Let $f(\cdot|\theta)$, with parameter θ, play the role of the segmentation model; namely, it evaluates the likelihood that a patch of the original image belongs to some precise class in $\mathcal{C}$.

We argue that Global Max Pooling (GMP) relates the two models $f(\cdot|\theta)$ and $f(\cdot|\varphi)$ as follows. Let Z^k be the slice of Z, along the categories axis, which should specify the patches internal structure for the category $k \in \mathcal{C}$. GMP selects the most relevant element of Z^k, namely the element with the highest confidence to belong to the category k, and returns a probability value y^k that it belongs to class $k \in \mathcal{C}$. The selected element Z^k_{ij}, at the same time, is the one in highest consideration to tell whether or not the category k appears in the image. In this way, GMP links image class prediction and patch class prediction.

4 The explicit search method

This section considers the optimization method leading to estimating the map between image classes and patch classes. The end-to-end architecture enclosing the method is described in Figure 2, and in Section 5.

Let us indicate by f the network taking inputs from a dataset $\mathcal{D}{=}\{\langle X_{in}, t\rangle\}$. Here $X_{in}{\in}\mathbb{R}^{h\times w\times 3}$ indicates an input images, possibly obtained from an augmented and transformed set, $t{\in}\{0,1\}^K$ are the ground truth binary labels, and K is the number of classes defined by the category set $\mathcal{C}{=}\{0,1,\ldots,K\}$. The output of f is a tensor $\hat{Y} \in \mathcal{C}^{h\times w}$ which is a *baseline pseudo-mask*.

ViT is part of f. We recall that ViT partitions the image X, resized image of the original X_{in}, into s patches of size $(d{\times}d{\times}3)$. In particular, we are interested in the feature maps $F{\in}\mathbb{R}^{s\times e}$, with $s{=}(n/d)^2$, with $n{=}w{=}h$. The feature maps F are the encoded representations of the patches, obtained by ViT. F represent the basis functions specifying the patches internal structure.

Explicit search by Global Max-Pooling Given $F{\in}\mathbb{R}^{s\times e}$, we consider also a weight matrix $W{\in}\mathbb{R}^{e\times K}$ whose weights are taken into account in the optimization method described below. More precisely, we estimate the baseline pseudo-mask $\hat{Y}$, training the weights W with only image-level class labels as supervision, minimizing the multilabel classification error.

The first objective is to minimize the multilabel classification prediction error (MCE). Thus, given the ground truth binary labels t defined above, and recalling that K are the number of classes, we model the multi-label classification using K independent Bernoulli distributions and K binary cross-entropy losses (BCE):

$$\mathcal{L}_{MCE} = \frac{1}{K}\sum_{k\in\mathcal{C}} BCE(t_k, y_k) = -\frac{1}{K}\sum_{k\in\mathcal{C}} t_k \log(y_k) + (1-t_k)\log(1-y_k). \tag{1}$$

Let us consider first how $y{\in}\mathbb{R}^K$ is obtained. Let:

$$A = FW \quad \text{and } Z = softmax(A), \text{ with } F{\in}\mathbb{R}^{s\times e}, W{\in}\mathbb{R}^{e\times K} \text{ hence } Z{\in}\mathbb{R}^{s\times K}. \tag{2}$$

Z represents the semantic segmentation predictions, needing to be projected into class predictions[3]. We do so using Global Max Pooling (GMP):

$$y_k = GMP(Z^k) = \max(Z^k) = Z^k_i, \text{ for some } i{\in}\{1,\ldots,s\}. \tag{3}$$

Here:

$$Z^k = softmax(A^k) \text{ and } A^k_j = F_j W^k \tag{4}$$

The feature maps F are the encoded representation of patches U, and F_j is the feature map of patch U_j, while A^k_j is the logit of patch U_j, $j = 0,\ldots,s$ with respect to class $k \in \{0,1,\ldots,K\}$.

[3] Note that we are representing here Z as a matrix, which is simply a reshaping of the tensor Z discussed in Section 3.

Given the vector y_k, we show how the optimization obtains the terms separating the feature space by the relative error backpropagation of $\mathcal{L}_{MCE}$, with respect to weights W. Computing the gradient of Eq. (1) w.r.t. the weight W, we obtain:

$$\frac{\partial \mathcal{L}_{MCE}}{\partial W} = \sum_{k \in \mathcal{C}} \frac{\partial BCE(t_k, y_k)}{\partial W} \tag{5}$$

Let us analyze the gradient of the weights W, with respect to each column h, of size e, with $h{\in}\{0, 1, \ldots K\}$. Applying the chain rule, w.r.t. the generic class k:

$$\frac{\partial BCE(t_k, y_k)}{\partial W^h} = \frac{\partial BCE(t_k, y_k)}{\partial y_k} \frac{\partial Z_i^k}{\partial A^h} \frac{\partial A^h}{\partial W^h} \tag{6}$$

Here we used the fact that $y_k{=}max(Z^k)$, and $max(Z^k){=}Z_i^k$ from eq. (3). Therefore, the gradient dimension is $\frac{\partial BCE(t_k,y_k)}{\partial W^h}{\in}\mathbb{R}^e$. The derivation of each term is provided in the supplementary.

Let us select, now, the column h of the weights W, this column will be updated by the quantity:

$$\begin{aligned} \frac{\partial \mathcal{L}_{MCE}}{\partial W^h} &= \frac{\partial BCE(t_h, y_h)}{\partial W^h} + \sum_{k \in \mathcal{C}, k \neq h} \frac{\partial BCE(t_k, y_k)}{\partial W^h} \\ &= -F_{i_h}(t_h - y_h) + \sum_{k \in \mathcal{C}, k \neq h} F_{i_k} Z_{i_k}^h \frac{t_k - y_k}{1 - y_k} \end{aligned} \tag{7}$$

Note that here the subscripts i_h, i_k in F and Z^h indicate, respectively, the indexes at which Z_i have maximum value, w.r.t classes h and k, where F_i is obtained by the last two terms of equation 6, r.h.s. We are using these indexes only in the updating rule for the weights; we are not using them in the derivation.

Eq. 7 specifies the linear-search mechanism of the proposed optimization, iteratively selecting the most representative features F_{i_h} of each category h. At each step, the optimization updates the full column rank matrix $W{\in}\mathbb{R}^{s \times e}$ and returns the minimum error norm solution, which separates the feature vector space $\mathbb{R}^e$ into K linear sub-spaces. Considering the optimization manifold, the vector W^h moves in the direction of the best representative feature vector F_{i_u}, with either u being of the same category of the chosen column h, or not. More precisely, at each iteration, W^h moves in the direction of F_{i_h} according to the error value $(t_h - y_h)$, and in the direction F_{i_k} according to the term $Z_{i_k}^h \frac{t_k - y_k}{1 - y_k}$, for any category k, with $k \neq h$.

More specifically, when the term $\frac{(t_k - y_k)}{1 - y_k}{=}1$, and the category $k{\neq}h$ is considered, W^h moves in the direction opposite to the best representative feature vector F_{i_k}. On the other hand, when $t_k = 0$ the term considered is $-(Z_{i_k}^k \frac{y_k}{1 - y_k})$ which is added to W^h, for its updating. Note that, in this case, the update term is increasingly small, since $y_k{\ll}1{-}y_k$ as $y_k{\to}0$. This optimization method, based on iterative learning and stochastic gradient descent, induces a separation in the space of patch features, according to the multilabel classification.

5 ViT-PCM model structure

The model architecture has two branches, as shown in Figure 2. We describe its components in the following.

Augmentation. The batch of input images is augmented as usual in the first branch. In the second branch, images are translated, rotated and scaled. Furthermore, we merge four images from the batch into a single image after scaling them to have a different tiling of the images into patches.

ViT patch encoder. The Vision Transformer encoder takes as input the augmented batch of images and returns the features F_{in} and the n patches described in the *explicit search* method, Section 4.

HV-BiLSTM patch conditioning. Two bidirectional LSTM (BiLSTM) process row-wise and column-wise the features F_{in} transformed to a tensor grid. The two BiLSTM outputs are concatenated into a HV-BiLSTM (for Horizontal and Vertical), and their feature maps F are fed to the Patch Classifier. The HV-BiLSTM improves information amid neighbour patches by conditioning each patch on all other ones in horizontal (H) and vertical directions (V) [48].

Patch Classifier (PC). While ViT and the two BiLSTM encode class information into the patch features, the Patch Classifier implements the BPM generation, as described in the explicit search method, Section 4.

Two branches for Equivariant regularization. ViT are not equivariant to translations because of the absolute positional encoding used for self-attention. Romero *et Al.* [38] show that for self-attention to be equivariant to group transformations, they must act directly on positional encoding. In our ViT-based method, though GMP is independent of the positional encoding and is invariant to transformations, the BPM generation is not. To remedy we resort to typical self-supervised learning tasks, using two branches enabling the network to learn equivariance properties. Equivariance encourages the feature representation to change coherently to the transformation applied to the input [13]. As discussed above, we apply affine transformations to both the network branches in the preprocessing step. After the same processing steps of the main branch, the sibling one applies an inverse merging of the features and upscales them to obtain the n patches feature maps as in the main branch. Finally, inverse affine transformations are applied to both branches.

The outcome is that these transformations cope both with positional encoding and spatial transformations. The loss to be minimized is the cross entropy loss $\mathcal{L}_{ET}$, taking into account the transformations in the two branches:

$$\begin{gathered}\mathcal{L}_{ET} = -\frac{1}{s}\sum_{i=0}^{s}\sum_{X\in\mathcal{X}} \nu_i(X)\log\mu_i(X) \\ \text{with } \mu_i(X) = a^{-1}f(a(X)) \text{ and } \nu_i(X) = c^{-1}f(c(X))\end{gathered} \tag{8}$$

Here, $\mathcal{X}$ is the images domain, $a(\cdot), b(\cdot)$ are affine transformations in the first and second branch, $m(\cdot)$ is the above defined merging operation, and $c = m(b(\cdot))$.

Final loss We have the $\mathcal{L}_{MCE}$ loss, conveying the mapping between image classification and patch classification, and $\mathcal{L}_{ET}$, which ensures equivariance and

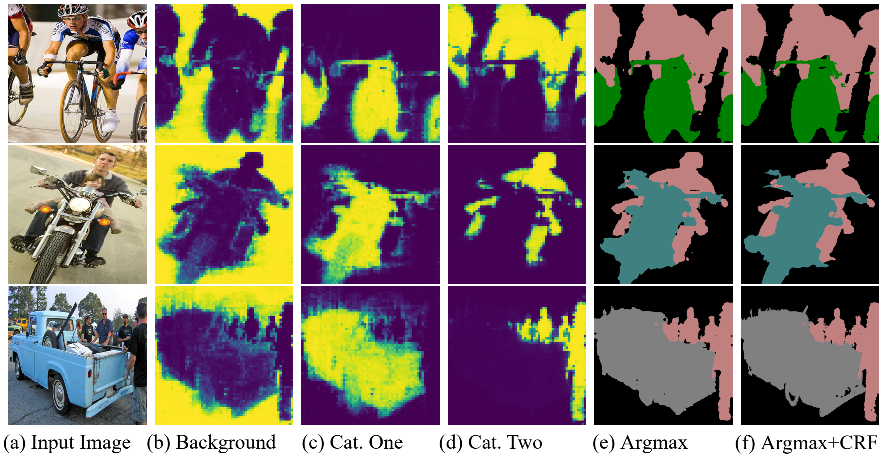

Fig. 4: Columns (b)-(c)-(d) show the BPM inferred by our ViT-PCM, with probabilities highlighted by 60×60 heatmaps: values in yellow indicate the pixels' probability of belonging to the predicted class. Column (e) is the scaled BPM, obtained by selecting from the distribution of each patch the category indices with maximum probability (argmax). Column (f) displays the BPM argmax refined by CRF.

scales the images so that patches get pixel dimension. The final loss is then:

$$\mathcal{L} = \mathcal{L}_{MCE} + \mathcal{L}_{ET} \tag{9}$$

Training the end-to-end network by minimizing this final loss obtains the baseline pseudo-mask.

6 Experiments and results

6.1 Set-Up

Datasets. We conducted our experiments on Pascal VOC 2012 [17] (20 categories) and on MS COCO 2014[34] (80 categories), the additional background class is inferred. The Pascal VOC 2012 Dataset [17] is usually augmented with the SBD dataset [21]. The images in train sets of PASCAL VOC and MS COCO are annotated with image-level labels only. We report mean Intersection-Over-Union (mIoU) as the evaluation criteria.

Networks Configuration. For the ViT transformer backbones [15] we used ViT-S/16 and ViT-B/16 architectures, pre-trained on ImageNet22K and fine-tuned on ImageNet2012 [40]. We designed an MLP layer projecting the patch features into a categorical distribution on the K classes as a baseline model for ablation purposes. For the verification task, we used DeepLab V2[8].

Reproducibility. Images are resized to 384×384 for training and augmented by random colour jitter, random grayscale, 90° rotation, and vertical and horizontal

Table 1: Ablation on our ViT-PCM model for baseline pseudo-mask production, on PASCAL-VOC 2012 values in mIoU%.

Backbone	$\mathcal{L}_{MCE}$	$\mathcal{L}_{ET}$	HV-BiLSTM	CRF	**train**	**val**
ViT-S/16	✓				44.0	43.3
	✓	✓			59.2 +15.2	56.4 +13.1
	✓	✓	✓		63.6 +4.4	61.8+5.4
	✓	✓	✓	✓	67.1 +3.5	64.9+3.1
ViT-B/16	✓				45.6	44.1
	✓	✓			65.1 +19.5	62.4 +18.3
	✓	✓	✓		67.7 +2.6	66.0+3.6
	✓	✓	✓	✓	71.4 +3.7	69.3+3.3

flip. Initially, we freeze the backbone and ignore the output feature for the $[cls]$ token. At the same time, we preserve the $24{\cdot}24$ encoded patch features as input to the BiLSTM conditioning, whose outcome features are passed to the Patch Classifier. We initialize the MLP layer with standard Gaussian distribution and use L2 regularization with coefficient l_2=10^{-1}. We ran our training sessions iterating over the entire dataset, each epoch measuring the mIoU(%) progresses on the PascalVOC 2012 and MS COCO2014 validation sets. We keep the input resolution to 384×384 to hasten the evaluation on a 4 NVIDIA Titan V GPUs with 12GB RAM each, a deliberately limited resources setup. We use Adam optimization and schedule the learning rate as follows: 10^{-3} learning rate for the first two epochs with a frozen backbone; then, we unfreeze the last four backbone layers and keep training until convergence with 10^{-4} learning rate. At inference time, we scale the input image to 960×960 to get pseudo-label segmentation maps of shape 60×60. As expected, we noticed an increase in performance of about $2{-}3\%$ mIoU scores for validation in the training session, confirming that ViTs scales very well on larger input size.

6.2 Ablation studies

In Table 1 we evaluate ViT-PCM computation both with backbone ViT-S/16 and ViT-B/16, considering each component of the end-to-end network. We adopted a patch size of 16 since the memory requirements grow quadratically with the number of patches. The low scores of the ($\mathcal{L}_{MCE}$) in Table 1 are due to the difficulty in encoding the background without equivariance. We observe that with the equivariance, $\mathcal{L}_{ET}$ there is an improvement of 15.2 mIoU% on the *train* set and 13.1 mIoU% on the *val* set for PascalVOC 2012. A further improvement of 4.4 on the *train* set and 5.4 mIoU% on the *val* set is obtained by conditioning the patches with HV-BiLSTM. Finally, we add the dCRF[27] as post-processing obtaining an improvement of 3.5 mIoU% on *train* set.

Figure 4 shows the BPM heat-maps for each class in the second, third and fourth columns, inferred by our end-to-end network, including the background. The BPM heat map highlights each pixel's likelihood of belonging to a specific

Table 2: mIoU(%) of BPM on PascalVOC 2012 *val* set. w/wo CRF

Method	bkg	plane	bike	bird	boat	btl	bus	car	cat	chair	cow	table	dog	horse	mbk	person	plant	sheep	sofa	train	tv	mIoU(%)
pseudo-masks w/o CRF	87.2	66.4	36.9	61.0	61.1	63.0	86.8	76.0	76.9	41.1	80.7	39.0	82.3	77.4	75.7	55.9	50.6	85.0	50.9.6	78.9	54.7	66.0
pseudo-masks w/ CRF	88.8	78.2	39.1	69.2	67.2	67.2	88.0	77.7	78.5	42.5	83.9	39.2	85.2	82.8	79.8	56.2	51.0	91.3	51.0	81.9	57.0	69.3

category. Column (e) shows the pseudo-masks obtained by selecting the indices of the classes with maximum probability. Column (f) shows the pseudo-masks improved by CRF. We use these last masks for the verification task as input to DeepLab [7].

In Table 2 we report the BPM mIoU% on Pascal VOC val set for each category, w and w/o CRF.

6.3 Comparisons with state-of-the-art

Comparison on baseline pseudo-masks. We compare the mIoU(%) accuracy of our ViT-PCM method with other methods, which compute BPM and post-process them with CRF [27] similarly. Some methods such as CIAN [19] and EDAM [54] also incorporate saliency.

Results are reported in Table A. Here we can observe that CRF, used as BPM post-processing, improves the BPM, on average, by 3.97%, with a standard deviation of 1.87. The statistics show that CRF out of a training loop behaves similarly on all methods. Observe that we improved BPM state-of-the-art by 3.91 mIou% points and BPM+CRF by 5.4 mIoU%, both w.r.t. AFA[39], owning so far the best accuracy on both.

Table A: mIoU(%) on PascalVOC2012 train set.

Method	Backbone	BPM	BPM+CRF
ICD [18] CVPR'20	VGG16	57.00	62.20
SCE[6] CVPR'20	ResNet38	50.90	-
SEAM [50] CVPR'20	ResNet38	55.41	56.83
CIAN[19] AAAI'20	ResNet101	58.10	62.50
ECSNet[46] ICCV'20	ResNet38	56.60	58.60
PAMR[3] CVPR'20	ResNet38	59.7	62.7
AdvCAM[30] CVPR'21	ResNet50	55.60	62.10
CPN[60] ICCV'21	ResNet38	57.43	-
CSE[29] ICCV'21	ResNet38	56.0	62.8
EDAM[54] CVPR'21	ResNet101	52.83	58.18
MCTformer[58] CVPR'22	DeiT-S	61.70	-
PPC[16] CVPR'22	Resnet38	61.50	64.00
CLIMS[55] CVPR'22	Resnet50	56.60	-
SIPE[10] CVPR'22	Resnet50	58.60	64.70
AFA[39] CVPR'22	MiT-B1	63.80	66.00
IRN+W-OoD[31] CVPR'22	Resnet50	53.30	58.40
ViT-PCM Ours	ViT-B/16	**67.71**	**71.4**

Table C: mIoU(%) on MS-COCO 2014 val set.

Method	Backbone	Val
MCTformer[58] CVPR'22	Resnet38	42.0
SIPE[10] CVPR'22	Resnet38	43.6
ViT-PCM Ours	ViT-B/16	**45.0**

Table B: mIoU(%) on PascalVOC2012 val and test set.

Method	Backbone	Val	Test
IRNet[1] CVPR'19	ResNet50	63.5	64.8
SCE[6] CVPR'20	ReseNet101	66.1	65.9
SEAM[50] CVPR'20	ResNet38	64.5	65.7
CIAN[19] AAAI'20	ResNet101	64.3	65.3
ECSNet[46] ICCV'20	ResNet38	66.6	67.6
CONTA[14] NUERIPS'20	ResNet101	66.1	66.7
BES[9] ECCV'20	ResNet101	65.7	66.6
AdvCAM[30] CVPR'21	ResNet50	68.1	68.0
CPN[60] ICCV'21	ResNet38	67.8	68.5
EDAM[54] CVPR'21	ResNet101	52.83	58.18
CSE[29] ICCV'21	ResNet38	68.4	68.2
MCTformer[58] CVPR'22	Resnet38	71.9	71.6
CLIMS[55] CVPR'22	Resnet50	70.4	70.0
SIPE[10] CVPR'22	Resnet101	68.8	69.7
AdvCAM+W-OoD[31] CVPR'22	Resnet38	70.7	70.1
PAMR[3] CVPR'20	ResNet38	62.7	64.3
MCIS[45] ECCV'20	ResNet101	66.2	66.9
ICD [18] CVPR'20	Resnet101	64.1	64.3
AFA[39] CVPR'22	MiT-B1	66.0	66.3
MCTformer⋆[58] CVPR'22	Resnet38	68.2	68.4
ViT-PCM Ours	ResNet 101	**70.3**	**70.9**

Table D: mIoU(%) on PascalVOC2012 val set.

Method	ViT-S/8	ViT-S/16	ViT-B/16
DINO	44.7	45.9	-
ViT-PCM Ours	-	**74.55**	**77.25**

Semantic Segmentation Verification Tasks The verification task of the WSSS methods on PascalVOC 2012 tests the final pseudo-mask (FPM), and the results are reported in Table B. We divide the methods into two: those which are boosted (or, according to the definition in PAMR [3] are multi-stage) and

those which are end-to-end, highlighted in grey. For the methods considered, the boosted ones improve the mIoU% w.r.t. the BPM on average of 9.8%, while the end-to-end methods improve on average 4.2%. Our ViT-PCM not being boosted improves by 2.39% on the val set and decreases on the test set. Our ViT-PCM has the best accuracy among the end-to-end methods, with 70.3% and 70.9% on *val* and *test* sets. Our method is second to MCT-Former[58] on the test set w.r.t. all methods (boosted and end-to-end). However, MCT-Former end-to-end version is second to ViT-PCM, on both the val and test sets.
In Table C we also evaluate our method on MS-COCO 2014 dataset [34]. Our ViT-PCM achieves 45.03 mIoU% on *val* set. We reported only the last methods (2022) with the highest performance. Table D compares our foreground maps with DINO [5] maps on the PascalVOC 2012 val set. Figure 5 shows the ratio between the parameters consumed to obtain the BPM and the final segmentation mask, against the mIoU% on the val set of PascalVOC2012. A ⋆ marker specifies the BPM, and a □ marker specifies the final segmentation mask, ours in red and the others in blue. Our ViT-PCM, with backbone ViT-S/16, is green-dashed, and ViT-B/16 is green-continuous. We can observe that most of the shown methods are multi-stage (see also [3,39]), and boosting the BPM asks for a significant increase of parameters. Table 3 shows the accuracy between CAM and PCM on different backbones and the amount of parameters required. We made this table to understand whether it would be profitable to use CAM with ViT. As shown in the table, we can see that the combination ViT and PCM is the best solution. Figure 6 compares our qualitative results on Pascal VOC 2012 val set with other approaches whose implementation we have used to generate the images; therefore, they might be biased.

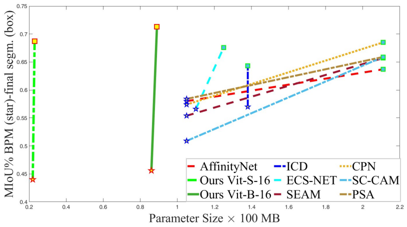

Fig. 5: Networks parameters consumed from the BPM to the final-segmentation in ours and other methods, against mIoU% on PascalVoc2012 val. set.

Backbone	Params (M)	Localization	mIoU (%)	pixAcc (%)
Resnet50v2	25	CAM	**27.8**	72.7
		PCM	25.2	**76.0**
Xception	23	CAM	**37.8**	76.5
		PCM	36.5	**79.5**
ViT-S/16	**22**	CAM	29.3	55.0
		PCM	**43.3**	**80.1**

Table 3: Comparison between CAM [62] and PCM (our Patch Class Mapping) on PascalVOC2012 *val* set. The Table reports the best results obtained with Multi-Label BCE loss and L2 regularization loss in all experiments, for both CAM and PCM.

6.4 Limitations

We observed that ViT-PCM is biased on the most discriminative features. Many approaches to WSSS highlight the improvements due to processing pixel relations, boundaries, and neighbourhoods. We have used only the conditioning from

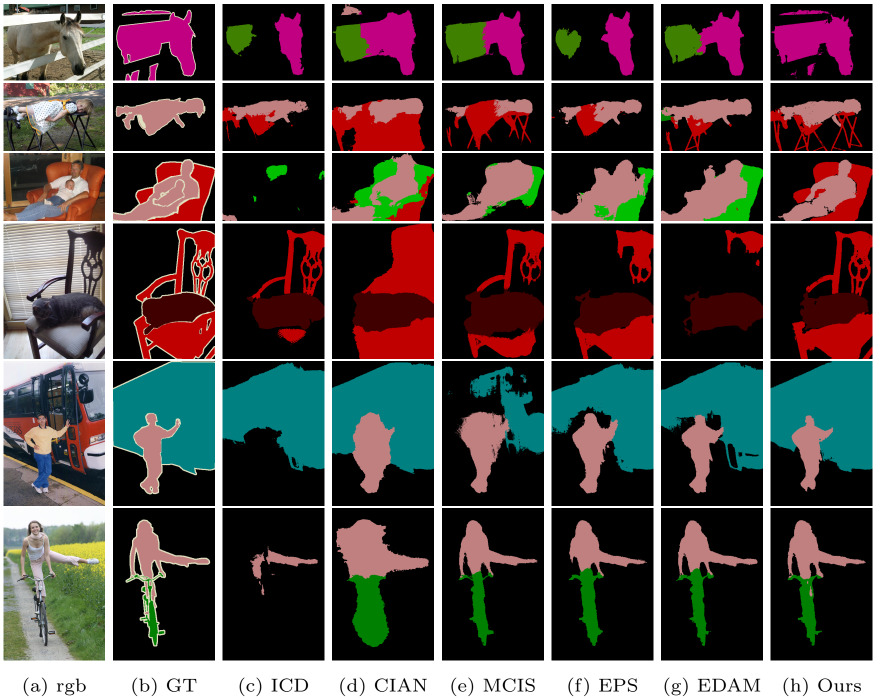

Fig. 6: Qualitative comparison on Pascal VOC 2012 validation set.

HV-BiLSTM, which might not be the best solution. On the other hand, some recent approaches have explored contrastive loss for foreground-background learning with no image-level supervision. Since the background is our Achille's heel, we could have explored this idea. Another bottleneck of our approach is the final scaling to map patches to pixels, where we perform a rough scaling to keep the resources limited.

7 Conclusions

We presented an innovative, simple and end-to-end method, ViT-PCM, based on ViT for generating baseline pseudo-masks (BPM) with precise localization and higher quality than those obtained from the more involved CAM CNN-based architectures. We obtained new state-of-the-art in BPM generation with 67.7 % mIoU on PascalVOC 2012 *train set* and 71.4% mIoU using CRF only in post-processing. These results demonstrate this work's high contribution to the field of WSSS. Therefore, we hope that others will continue in this direction. In the supplementary files, we report more analysis and results. The code is available at https://github.com/deepplants/ViT-PCM.

References

1. Ahn, J., Cho, S., Kwak, S.: Weakly supervised learning of instance segmentation with inter-pixel relations. In: ICCV. pp. 2209–2218 (2019) 2, 3, 5, 12
2. Ahn, J., Kwak, S.: Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In: CVPR. pp. 4981–4990 (2018) 2, 3, 5
3. Araslanov, N., Roth, S.: Single-stage semantic segmentation from image labels. In: CVPR. pp. 4253–4262 (2020) 2, 12, 13
4. Bae, W., Noh, J., Kim, G.: Rethinking class activation mapping for weakly supervised object localization. In: ECCV. pp. 618–634 (2020) 2
5. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: ICCV. pp. 9650–9660 (2021) 5, 13
6. Chang, Y.T., Wang, Q., Hung, W.C., Piramuthu, R., Tsai, Y.H., Yang, M.H.: Weakly-supervised semantic segmentation via sub-category exploration. In: CVPR. pp. 8991–9000 (2020) 3, 12
7. Chen, L.C., Papandreou, G., Kokkinos, I., Murphy, K.P., Yuille, A.L.: Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE TPAMI **40**, 834–848 (2018) 1, 12
8. Chen, L.C., Zhu, Y., Papandreou, G., Schroff, F., Adam, H.: Encoder-decoder with atrous separable convolution for semantic image segmentation. In: Proceedings of the European conference on computer vision (ECCV). pp. 801–818 (2018) 10
9. Chen, L., Wu, W., Fu, C., Han, X., Zhang, Y.: Weakly supervised semantic segmentation with boundary exploration. In: European Conference on Computer Vision. pp. 347–362. Springer (2020) 2, 5, 12
10. Chen, Q., Yang, L., Lai, J.H., Xie, X.: Self-supervised image-specific prototype exploration for weakly supervised semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4288–4298 (2022) 12
11. Chen, X., Xie, S., He, K.: An empirical study of training self-supervised vision transformers. In: ICCV. pp. 9640–9649 (2021) 5
12. Chen, Z., Wang, T., Wu, X., Hua, X.S., Zhang, H., Sun, Q.: Class re-activation maps for weakly-supervised semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 969–978 (2022) 2, 3, 5
13. Dangovski, R., Jing, L., Loh, C., Han, S., Srivastava, A., Cheung, B., Agrawal, P., Soljacic, M.: Equivariant self-supervised learning: Encouraging equivariance in representations. In: ICLR (2021) 9
14. Dong, Z., Hanwang, Z., Jinhui, T., Xiansheng, H., Qianru, S.: Causal intervention for weakly supervised semantic segmentation. In: Neurips (2020) 3, 12
15. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2021) 4, 5, 6, 10
16. Du, Y., Fu, Z., Liu, Q., Wang, Y.: Weakly supervised semantic segmentation by pixel-to-prototype contrast. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4320–4329 (2022) 5, 12
17. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. Int. J. of computer vision **88**(2), 303–338 (2010) 10

18. Fan, J., Zhang, Z., Song, C., Tan, T.: Learning integral objects with intra-class discriminator for weakly-supervised semantic segmentation. In: CVPR (2020) 12
19. Fan, J., Zhang, Z., Tan, T., Song, C., Xiao, J.: Cian: Cross-image affinity net for weakly supervised semantic segmentation. In: AAAI (2020) 2, 12
20. Guo, H., Zheng, K., Fan, X., Yu, H., Wang, S.: Visual attention consistency under image transforms for multi-label image classification. In: ICCV. pp. 729–739 (2019) 2
21. Hariharan, B., Arbeláez, P., Bourdev, L., Maji, S., Malik, J.: Semantic contours from inverse detectors. In: 2011 international conference on computer vision. pp. 991–998. IEEE (2011) 10
22. Hou, Q., Cheng, M.M., Hu, X., Borji, A., Tu, Z., Torr, P.H.: Deeply supervised salient object detection with short connections. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3203–3212 (2017) 3
23. Huang, Z., Wang, X., Wang, J., Liu, W., Wang, J.: Weakly-supervised semantic segmentation network with deep seeded region growing. In: CVPR. pp. 7014–7023 (2018) 2
24. Jiang, P.T., Han, L.H., Hou, Q., Cheng, M.M., Wei, Y.: Online attention accumulation for weakly supervised semantic segmentation. IEEE TPAMI (2021) 5
25. Jiang, P.T., Hou, Q., Cao, Y., Cheng, M.M., Wei, Y., Xiong, H.K.: Integral object mining via online attention accumulation. In: ICCV. pp. 2070–2079 (2019) 2, 5
26. Kolesnikov, A., Lampert, C.H.: Seed, expand and constrain: Three principles for weakly-supervised image segmentation. In: ECCV. pp. 695–711 (2016) 2, 3
27. Krähenbühl, P., Koltun, V.: Efficient inference in fully connected crfs with gaussian edge potentials. Advances in neural information processing systems **24** (2011) 3, 5, 11, 12
28. Kuznetsova, A., Rom, H., Alldrin, N., Uijlings, J., Krasin, I., Pont-Tuset, J., Kamali, S., Popov, S., Malloci, M., Kolesnikov, A., et al.: The open images dataset v4. International Journal of Computer Vision **128**(7), 1956–1981 (2020) 5
29. Kweon, H., Yoon, S.H., Kim, H., Park, D., Yoon, K.J.: Unlocking the potential of ordinary classifier: Class-specific adversarial erasing framework for weakly supervised semantic segmentation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 6994–7003 (2021) 5, 12
30. Lee, J., Kim, E., Yoon, S.: Anti-adversarially manipulated attributions for weakly and semi-supervised semantic segmentation. In: CVPR. pp. 4071–4080 (2021) 2, 3, 5, 12
31. Lee, J., Oh, S.J., Yun, S., Choe, J., Kim, E., Yoon, S.: Weakly supervised semantic segmentation using out-of-distribution data. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16897–16906 (2022) 5, 12
32. Lee, S., Lee, M., Lee, J., Shim, H.: Railroad is not a train: Saliency as pseudo-pixel supervision for weakly supervised semantic segmentation. In: CVPR. pp. 5495–5505 (2021) 2, 5
33. Li, C., Yang, J., Zhang, P., Gao, M., Xiao, B., Dai, X., Yuan, L., Gao, J.: Efficient self-supervised vision transformers for representation learning. arXiv preprint arXiv:2106.09785 (2021) 5
34. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: ECCV. pp. 740–755 (2014) 10, 13
35. Montavon, G., Lapuschkin, S., Binder, A., Samek, W., Müller, K.R.: Explaining nonlinear classification decisions with deep taylor decomposition. Pattern recognition **65**, 211–222 (2017) 2

36. Oh, S.J., Benenson, R., Khoreva, A., Akata, Z., Fritz, M., Schiele, B.: Exploiting saliency for object segmentation from image level labels. In: CVPR. pp. 5038–5047 (2017) 5
37. Ren, S., Wang, H., Gao, Z., He, S., Yuille, A., Zhou, Y., Xie, C.: A simple data mixing prior for improving self-supervised learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 14595–14604 (2022) 5
38. Romero, D.W., Cordonnier, J.B.: Group equivariant stand-alone self-attention for vision. arXiv preprint arXiv:2010.00977 (2020) 9
39. Ru, L., Zhan, Y., Yu, B., Du, B.: Learning affinity from attention: End-to-end weakly-supervised semantic segmentation with transformers. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 16846–16855 (2022) 3, 4, 5, 12, 13
40. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. Int. J. of computer vision **115**(3), 211–252 (2015) 10
41. Shimoda, W., Yanai, K.: Self-supervised difference detection for weakly-supervised semantic segmentation. In: ICCV. pp. 5208–5217 (2019) 3
42. Singh, K.K., Lee, Y.J.: Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In: ICCV. pp. 3544–3553 (2017) 2
43. Stammes, E., Runia, T.F., Hofmann, M., Ghafoorian, M.: Find it if you can: end-to-end adversarial erasing for weakly-supervised semantic segmentation. In: ICDIP. vol. 11878 (2021) 5
44. Su, H., Jampani, V., Sun, D., Gallo, O., Learned-Miller, E., Kautz, J.: Pixel-adaptive convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11166–11175 (2019) 2, 3
45. Sun, G., Wang, W., Dai, J., Van Gool, L.: Mining cross-image semantics for weakly supervised semantic segmentation. In: ECCV. pp. 347–365 (2020) 2, 12
46. Sun, K., Shi, H., Zhang, Z., Huang, Y.: Ecs-net: Improving weakly supervised semantic segmentation by using connections between class activation maps. In: ICCV. pp. 7283–7292 (2021) 12
47. Sun, W., Zhang, J., Barnes, N.: Inferring the class conditional response map for weakly supervised semantic segmentation. In: WACV. pp. 2878–2887 (2022) 2
48. Van Oord, A., Kalchbrenner, N., Kavukcuoglu, K.: Pixel recurrent neural networks. In: International conference on machine learning. pp. 1747–1756. PMLR (2016) 9
49. Vernaza, P., Chandraker, M.: Learning random-walk label propagation for weakly-supervised semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7158–7166 (2017) 5
50. Wang, Y., Zhang, J., Kan, M., Shan, S., Chen, X.: Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In: CVPR. pp. 12275–12284 (2020) 2, 3, 5, 12
51. Wei, Y., Feng, J., Liang, X., Cheng, M.M., Zhao, Y., Yan, S.: Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In: CVPR. pp. 1568–1576 (2017) 2
52. Wei, Y., Liang, X., Chen, Y., Shen, X., Cheng, M.M., Feng, J., Zhao, Y., Yan, S.: Stc: A simple to complex framework for weakly-supervised semantic segmentation. IEEE TPAMI **39**(11), 2314–2320 (2016) 2, 5

53. Wei, Y., Xiao, H., Shi, H., Jie, Z., Feng, J., Huang, T.S.: Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation. In: CVPR. pp. 7268–7277 (2018) 2
54. Wu, T., Huang, J., Gao, G., Wei, X., Wei, X., Luo, X., Liu, C.H.: Embedded discriminative attention mechanism for weakly supervised semantic segmentation. In: CVPR. pp. 16765–16774 (2021) 5, 12
55. Xie, J., Hou, X., Ye, K., Shen, L.: Clims: Cross language image matching for weakly supervised semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4483–4492 (2022) 12
56. Xie, J., Xiang, J., Chen, J., Hou, X., Zhao, X., Shen, L.: C2am: Contrastive learning of class-agnostic activation map for weakly supervised object localization and semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 989–998 (2022) 5
57. Xu, L., Ouyang, W., Bennamoun, M., Boussaid, F., Sohel, F., Xu, D.: Leveraging auxiliary tasks with affinity learning for weakly supervised semantic segmentation. In: ICCV. pp. 6984–6993 (2021) 2, 5
58. Xu, L., Ouyang, W., Bennamoun, M., Boussaid, F., Xu, D.: Multi-class token transformer for weakly supervised semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4310–4319 (2022) 4, 5, 12, 13
59. Yao, Y., Chen, T., Xie, G.S., Zhang, C., Shen, F., Wu, Q., Tang, Z., Zhang, J.: Non-salient region object mining for weakly supervised semantic segmentation. In: CVPR. pp. 2623–2632 (2021) 5
60. Zhang, F., Gu, C., Zhang, C., Dai, Y.: Complementary patch for weakly supervised semantic segmentation. In: ICCV. pp. 7242–7251 (2021) 12
61. Zhang, T., Lin, G., Liu, W., Cai, J., Kot, A.: Splitting vs. merging: Mining object regions with discrepancy and intersection loss for weakly supervised semantic segmentation. In: ECCV. pp. 663–679 (2020) 5
62. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: CVPR. pp. 2921–2929 (2016) 2, 4, 13
63. Zhou, T., Zhang, M., Zhao, F., Li, J.: Regional semantic contrast and aggregation for weakly supervised semantic segmentation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4299–4309 (2022) 2, 5