



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

Supplementary Material

Simone Rossetti^{1,2}, Dam Zappia¹, Marta Sanzari², Marco Schaerf^{2,1},
Fiara Pirri^{1,2}

¹ DeepPlants, @deepplants.com

² DIAG, Sapienza @diag.uniroma1.it

1 Derivation of the \mathcal{L}_{MCE} optimization

Here to keep this supplementary section self contained we reintroduce the whole paragraph comprehensive of the parts we omitted in the main paper.

Let us indicate by f the network taking inputs from a dataset $\mathcal{D}=\{(X_{in}, t)\}$. 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 $\mathcal{C}=\{0, 1, \dots, K\}$. 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):

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

Here, $y\in\mathbb{R}^K$ is obtained as described in the following. Let $Z=\text{softmax}(A)$ with $A=FW$, hence both A and Z are in $\mathbb{R}^{s\times K}$, since $W\in\mathbb{R}^{e\times K}$. Z are the semantic

segmentation predictions, needing to be projected into class predictions. We do so using global max pooling (GMP):

$$y_k = GMP(Z^k) = \max(Z^k) = Z_i^k, \text{ for some } i \in \{0, 1, \dots, s\} \quad (2)$$

where,

$$Z^k = \text{softmax}(A^k) = \begin{pmatrix} \frac{\exp(A_1^k)}{\sum_{c \in \mathcal{C}} \exp(A_c^k)} \\ \vdots \\ \frac{\exp(A_s^k)}{\sum_{c \in \mathcal{C}} \exp(A_c^k)} \end{pmatrix} \text{ and } A_j^k = F_j W^k \quad (3)$$

As defined above, F_j is the feature map of patch U_j , while A_j^k is the logit of patch U_j , $j=0, \dots, s$ with respect to class $k \in \{0, 1, \dots, K\}$, where the $k=0$ class is predicted within the optimization.

The relative error back propagation of \mathcal{L}_{MCE} w.r.t weights W is given by:

$$\frac{\partial \mathcal{L}_{MCE}}{\partial W} = \sum_{k=0}^K \frac{\partial BCE(t_k, y_k)}{\partial W} \quad (4)$$

To simplify we analyze the gradient with respect to each column q of the weights W , with $q=0, 1, \dots, K$. Applying the chain rule:

$$\frac{\partial BCE(t_k, y_k)}{\partial W^q} = \frac{\partial BCE(t_k, y_k)}{\partial y_k} \frac{\partial y_k}{\partial \max(Z^k)} \frac{\partial \max(Z^k)}{\partial A^q} \frac{\partial A^q}{\partial W^q} \quad (5)$$

Each term of Eq. (5) is derived in the following:

$$\frac{\partial BCE(t_k, y_k)}{\partial y_k} = -\frac{t_k}{y_k} + \frac{1-t_k}{1-y_k} = \frac{y_k - t_k}{y_k(1-y_k)} \quad (6)$$

$$\frac{\partial y_k}{\partial \max(Z^k)} = 1, \text{ since Eq. (2)} \quad (7)$$

By Eq. (7) we instantiate $\max(Z^k)$ as Z_i^k . Rewriting $\frac{\partial \max(Z^k)}{\partial A^q}$ as $\frac{\partial Z_i^k}{\partial A^q}$ we get the Jacobian of $\frac{\partial Z_i^k}{\partial A^q}$ as:

$$\frac{\partial Z_i^k}{\partial A^q} = \begin{pmatrix} \frac{\partial Z_i^k}{\partial A_0^q} & \cdots & \frac{\partial Z_i^k}{\partial A_s^q} \\ \vdots & \ddots & \vdots \\ \frac{\partial Z_i^k}{\partial A_0^q} & \cdots & \frac{\partial Z_i^k}{\partial A_s^q} \end{pmatrix} \quad (8)$$

hence for $j \in \{0, 1, \dots, s\}$:

$$\frac{\partial Z_i^k}{\partial A_j^q} = \begin{cases} 0 & \text{if } i \neq j \\ \frac{\exp(A_j^k)}{\sum_{c \in \mathcal{C}} \exp(A_c^k)} - \left(\frac{\exp(A_j^k)}{\sum_{c \in \mathcal{C}} \exp(A_c^k)} \right)^2 & \text{if } i = j \text{ and } q = k \\ -\frac{\exp(A_j^k) \exp(A_j^q)}{(\sum_{c \in \mathcal{C}} \exp(A_c^q))^2} & \text{if } i = j \text{ and } q \neq k \end{cases} \quad (9)$$

From Eq. (7) $y_k=Z_i^k$, and from the above Eq. (9) we instantiate A^q with A_i^q , then from Eq. (9) it follows that:

$$\frac{\partial Z_i^k}{\partial A_i^q} = \begin{cases} y_k(1 - y_k) & q = k \\ -y_k Z_i^q & q \neq k \end{cases} \quad (10)$$

Consider the following matrix:

$$Z = \underbrace{\begin{pmatrix} Z_0^0 & \dots & Z_0^q & \dots & Z_0^K \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ Z_s^0 & \dots & Z_s^q & \dots & Z_s^K \end{pmatrix}}_{\substack{\max(Z^q) = Z_j^q \\ j=0, \dots, s, \\ q \in \{0, \dots, K\}}} \left. \vphantom{\begin{pmatrix} Z_0^0 & \dots & Z_0^q & \dots & Z_0^K \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ Z_s^0 & \dots & Z_s^q & \dots & Z_s^K \end{pmatrix}} \right\} \sum_{k=0}^K Z_i^k = 1, \quad i=0, \dots, s \quad (11)$$

From the above setting, which is like the vector of Eq. (3) repeated up to K , we can see that for each row we have a categorical distribution, and it sums to one. More precisely, consider the above matrix $Z \in \mathbb{R}^{s \times K}$, we have that each Z_i^q is the softmax of A_i^q , and indicates the probability that the corresponding patch U_i is of class q . Namely:

$$p(U_i^0, \dots, U_i^K) = \prod_{k=0}^K Z_i^{[q=k]} \quad (12)$$

Which is, indeed, the joint mass function for patch U_i of a categorical distribution with probabilities Z_i^0, \dots, Z_i^K .

On the other hand, along each column, taking the maximum for each of them, we obtain

$$GMP(Z) = (\max(Z^0), \max(Z^1), \dots, \max(Z^K)). \quad (13)$$

$GMP(Z)$ then gives the probability for the q -th class to appear in the image X_{in} , it specifies, indeed, a multi-label classification. Note that, since $\max(Z^q) = Z_i^q$, for some $i=0, \dots, s$, we also know the location of the category, with respect to the patch U_i^q .

From the last two terms of the r.h.s. of eq.(5) and the definition of A we obtain that:

$$\frac{\partial A_i^q}{\partial W^q} = F_i. \quad (14)$$

We can see that $i = 0, \dots, s$ depends on the choice of A , which in turns depends on the index of $\max(Z^k)$.

Finally, we get the error backpropagation with respect to the network weights:

$$\frac{\partial BCE(t_k, y_k)}{\partial W^q} = F_i \cdot \begin{cases} y_k - t_k & q = k \\ Z_i^q \frac{t_k - y_k}{1 - y_k} & q \neq k \end{cases} \quad (15)$$

The gradients have size $\frac{\partial y_k}{\partial Z^k} \in \mathbb{R}^s$, $\frac{\partial Z^k}{\partial A^q} \in \mathbb{R}^{s \times s}$, $\frac{\partial A^q}{\partial W^q} \in \mathbb{R}^{s \times e}$, and $\frac{\partial BCE(t_k, y_k)}{\partial W^q} \in \mathbb{R}^e$. Note that in equation (15), according to equation (2), the subscript i , varying in $0, \dots, s$ concerns the GMP computed with respect to a specific class k , and either the choice of the column q for W is equal to such a k or it is not. We consider both cases, relative to the index where Z^k is maximum.

To keep track of the index i w.r.t. the specific class, for notational purpose we indicate by i_a the location at which the value Z^a is maximum, which we use improperly as a subscript also for the feature vectors F . Let us consider again the column q of the weights W , this column will be updated by the quantity:

$$\begin{aligned} \frac{\partial \mathcal{L}_{MCE}}{\partial W^q} &= \frac{\partial BCE(t_q, y_q)}{\partial W^q} + \sum_{\substack{k \in C \\ k \neq q}} \frac{\partial BCE(t_k, y_k)}{\partial W^q} \\ &= -F_{i_q}(t_q - y_q) + \sum_{\substack{k \in C \\ k \neq q}} F_{i_k} Z_{i_k}^q \frac{t_k - y_k}{1 - y_k} \end{aligned} \quad (16)$$

Eq. 16 specifies the linear-search mechanism of the proposed optimization, iteratively selecting the most representative features F_{i_q} of each category q . 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^q 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 q , or not. More precisely, at each iteration, W^q moves in the direction of F_{i_q} according to the error value $(t_q - y_q)$, and in the direction F_{i_k} according to the term $Z_{i_k}^q \frac{t_k - y_k}{1 - y_k}$, for any category k , with $k \neq q$.

More specifically, when the term $\frac{(t_k - y_k)}{1 - y_k} = 1$, and the category $k \neq q$ is considered, W^q 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^q , for its updating. Note that, in this case, the update term is increasingly small, since $y_k \ll 1 - y_k$ as $y_k \rightarrow 0$. 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.

2 Further Experiments and Results

2.1 Qualitative Results on Pascal VOC 2012

In Figure S1, we show more examples of the BMP inferred by ViT-PCM supervised by image-class labels. There are two or three original images per row, and the BMP for each class appearing in the image is on its right. We can note that the shapes are pretty accurate, but for the noise on the shape contour. In Figure S3, we show the best quality final pseudo-masks obtained by BPM plus CRF post-processing, verified by DeepLabV2 trained on our BPM. The CRF of [8], used as post-processing of the BPM, removes much of the noise, though it cannot improve a non-accurate shape.

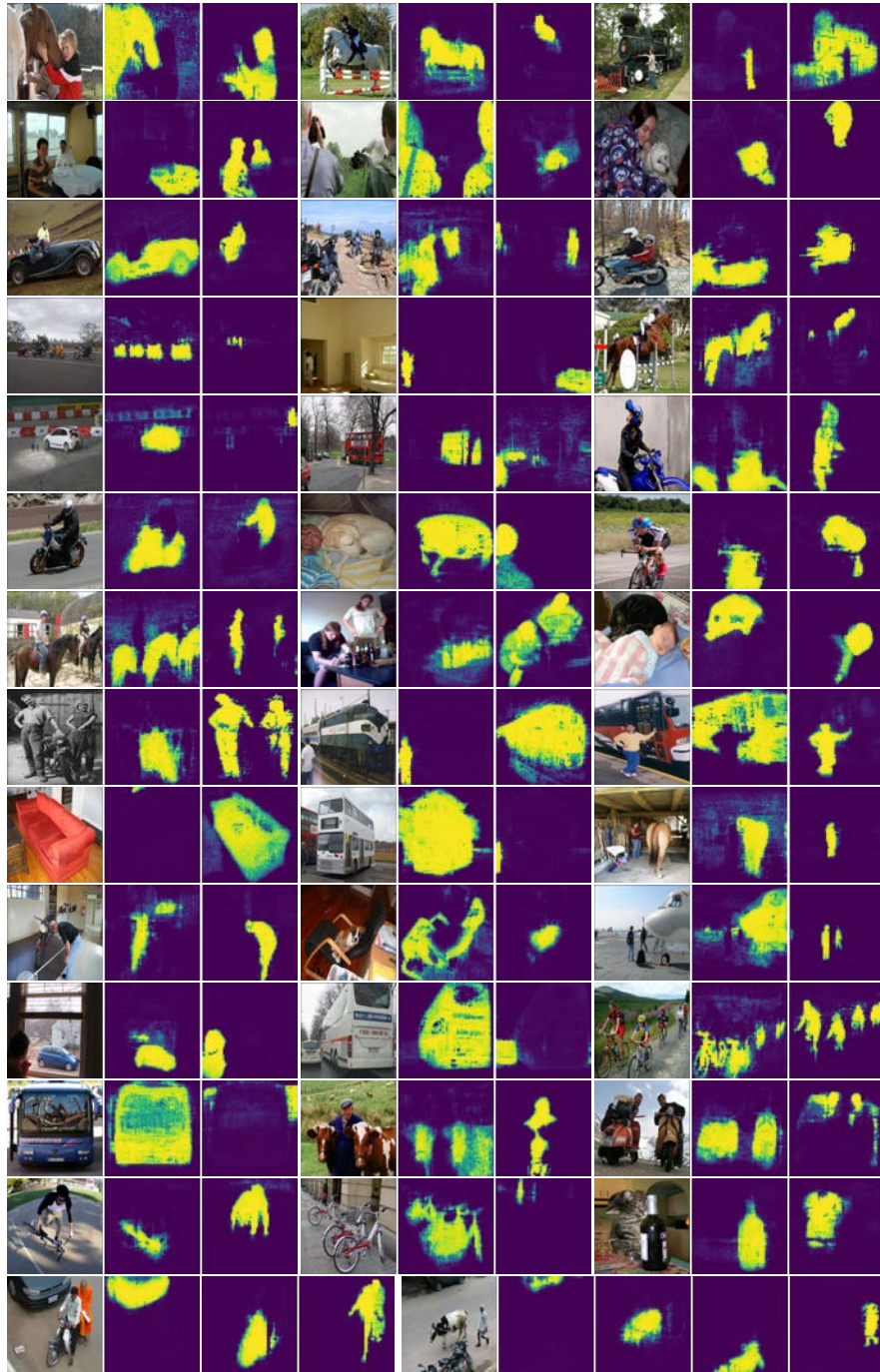


Fig. S1: Each row shows the original images (two or three per row) and the BPM for each class appearing in the image on its right. The BPM of size 60×60 are resized to the original image size.

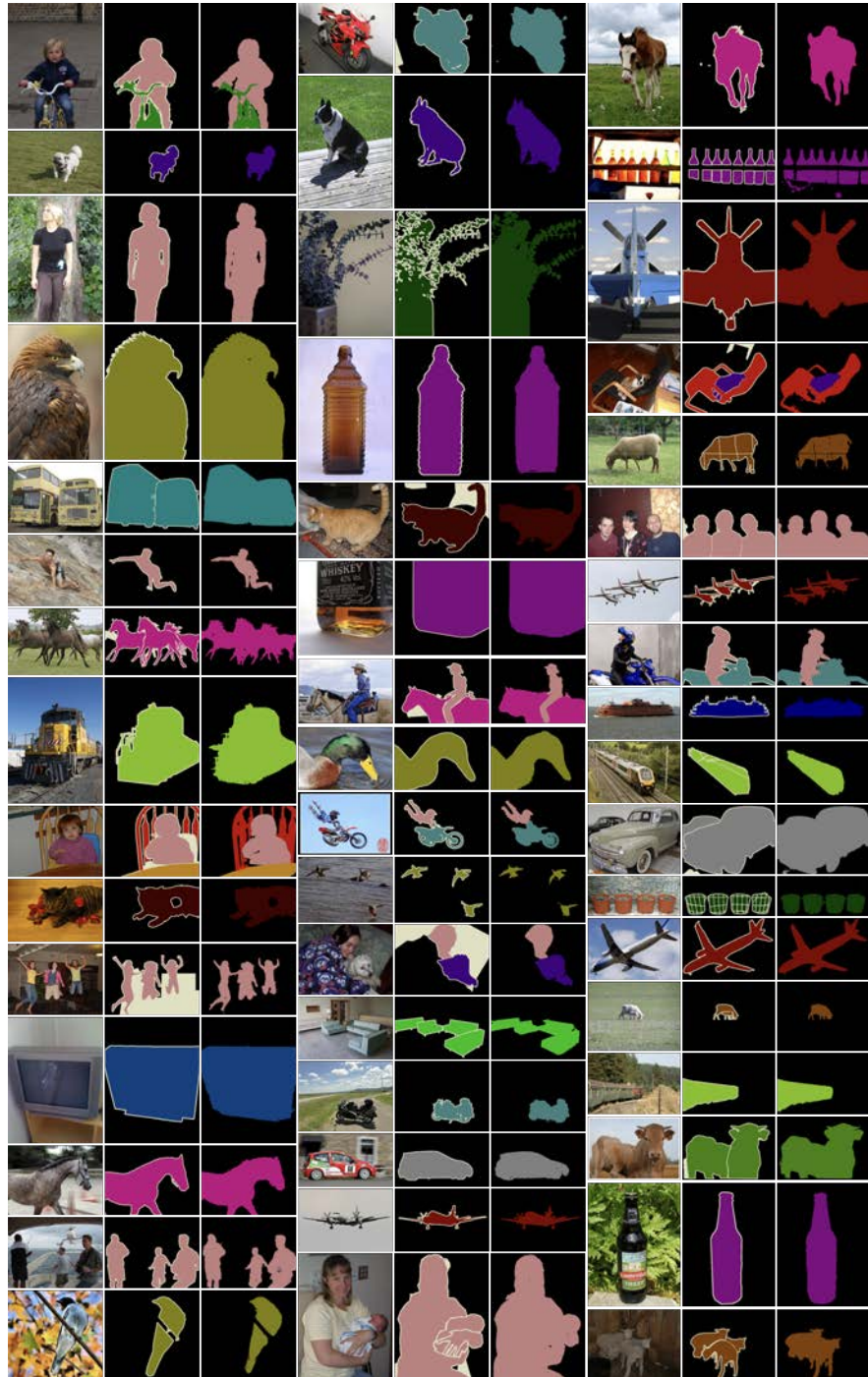


Fig. S2: Original image, ground-truth and final segmentation mask verified with DeepLab V2 [4] trained on our BPM.

3 Per-Class Comparisons with state-of-the-art on the Verification Task in Pascal VOC 2012

Table S1: Per-class performance comparison with the state-of-the-art WSSS methods on the verification task with final-segmentation masks, in terms of mIoU% on PASCAL VOC val set and test set.

Pascal VOC2012 val set																							
Method		bgk	aero	bike	bird	boat	btl	bus	car	cat	chair	cow	table	dog	horse	mbk	person	plant	sheep	sofa	train	tv	mIoU
CIAN[7] _{AAAI20}		83.6	59.4	35.4	53.7	39.8	56.2	79.3	73.0	79.0	28.9	67.5	54.8	74.9	68.7	74.1	58.7	49.5	72.3	33.5	57.3	50.4	59.5
SEAM[14] _{CVPR20}		88.8	68.5	33.3	85.7	40.4	67.3	78.9	76.3	81.9	29.1	75.5	48.1	79.9	73.8	71.4	75.2	48.9	79.8	40.9	58.2	53.0	64.5
BES[5] _{ICCV20}		88.9	74.1	29.8	81.3	53.3	69.9	89.4	79.8	84.2	27.9	76.9	46.6	78.8	75.9	72.2	70.4	50.8	79.4	39.9	65.3	44.8	65.7
ECSNet[13] _{ICCV20}		89.8	68.4	33.4	85.6	48.6	72.2	87.4	78.1	86.8	33.0	77.5	41.6	81.7	76.9	75.4	75.6	46.2	80.7	43.9	59.8	56.3	66.6
AdvCAM[10] _{CVPR21}		89.5	76.9	33.5	80.3	63.7	68.6	89.7	77.9	87.6	31.6	77.2	36.2	82.6	78.7	73.5	69.8	51.9	81.9	43.8	70.9	52.6	67.5
CPN[17] _{ICCV21}		89.9	75.1	32.9	87.8	60.9	69.5	87.7	79.5	89.0	28.0	80.9	34.8	83.4	79.7	74.7	66.9	56.5	82.7	44.9	73.1	45.7	67.8
CSE[9] _{ICCV21}		90.2	82.9	35.1	86.8	59.4	70.6	82.5	78.1	87.4	30.1	79.4	45.9	83.1	83.4	75.7	73.4	48.1	89.3	42.7	60.4	52.3	68.4
W-OoD[11] _{CVPR22}		91.2	80.1	34.0	82.5	68.5	72.9	90.3	80.8	89.3	32.3	78.9	31.1	83.6	79.2	75.4	74.4	58.0	81.9	45.2	81.3	54.8	69.8
MCT-Former[16] _{CVPR22}		91.9	78.3	39.5	89.9	55.9	76.7	81.8	79.0	90.7	32.6	87.1	57.2	87.0	84.6	77.4	79.2	55.1	89.2	47.2	70.4	58.8	71.9
End-to-end methods																							
PAMR[3] _{CVPR20}		88.7	70.4	35.1	75.7	51.9	65.8	71.9	64.2	81.1	30.8	73.3	28.1	81.6	69.1	62.6	74.8	48.6	71.0	40.1	68.5	64.3	62.7
ICD[6] _{CVPR20}		82.4	67.6	46.1	63.5	51.9	53.2	76.1	68.6	74.6	24.4	71.2	31.4	62.1	70.6	73.0	10.5	49.1	74.6	31.6	69.0	33.4	56.4
AFA[12] _{CVPR22}		89.9	79.5	31.2	80.7	67.2	61.9	81.4	65.4	82.3	28.7	83.4	41.6	82.2	75.9	70.2	69.4	53.0	85.9	44.1	64.2	50.9	66.0
MCT-Former*[16] _{CVPR22}		90.6	71.8	37.5	85.1	52.9	68.8	78.8	78.7	87.1	28.4	78.9	53.0	83.9	78.2	76.8	76.4	54.1	80.1	46.0	71.6	54.3	68.2
ViT-PCM Ours		91.2	86.0	37.8	83.7	67.1	70.2	90.4	85.0	90.2	29.5	82.1	57.3	84.1	78.3	77.7	83.5	53.0	78.7	22.7	82.6	44.8	70.3
Pascal VOC2012 test set																							
CIAN[7]		82.1	57.6	28.5	49.2	36.5	58.9	84.6	72.4	76.6	23.3	68.4	47.0	72.1	66.8	70.6	61.2	39.4	64.1	34.6	55.8	47.4	57.0
AdvCam[14] _{CVPR20}		90.1	81.2	33.6	80.4	52.4	66.6	87.1	80.5	87.2	28.9	80.1	38.5	84.0	83.0	79.5	71.9	47.5	80.8	59.1	65.4	49.7	68.0
CPN[17] _{ICCV21}		90.4	79.8	32.9	85.8	52.9	66.4	87.2	81.4	87.6	28.2	79.7	50.2	82.9	80.4	78.9	70.6	51.2	83.4	55.4	68.5	44.6	68.5
W-OoD[11] _{CVPR22}		91.4	85.3	32.8	79.8	59.0	68.4	88.1	82.2	88.3	27.4	76.7	38.7	84.3	81.1	80.3	72.8	57.8	82.4	59.5	79.5	52.6	69.9
MCT-Former[16] _{CVPR22}		92.3	84.4	37.2	82.8	60.0	72.8	78.0	79.0	89.4	31.7	84.5	59.1	85.3	83.8	79.2	81.0	53.9	85.3	60.5	65.7	57.7	71.6
End-to-end methods																							
PAMR[3] _{CVPR20}		89.2	73.4	37.3	68.3	45.8	68.0	72.7	64.1	74.1	32.9	74.9	39.2	81.3	74.6	72.6	75.4	58.1	71.0	48.7	67.7	60.1	64.3
ICD[6]		83.7	75.3	31.4	68.8	56.1	63.4	87.6	77.2	76.6	25.0	72.4	37.2	67.4	73.0	70.1	7.6	46.0	79.8	31.2	75.0	33.3	59.0
MCT-Former*[16] _{CVPR22}		90.9	76.0	37.2	79.1	54.1	69.0	78.1	78.0	86.1	30.3	79.5	58.3	81.7	81.1	77.0	76.4	49.2	80.0	55.1	65.4	54.5	68.4
ViT-PCM Ours		91.1	88.9	39.0	87.0	58.8	69.4	89.4	85.4	89.9	30.7	82.6	62.2	85.7	83.6	79.7	81.6	52.1	82.0	26.5	80.3	42.4	70.9

In Table S2 we report the per-class comparison with other WSSS methods on val and test set of Pascal VOC 2012. Not all methods report both the val and the test set. We divide the methods into two sets: those using boosting and those end-to-end. An explanation of the computational effort of boosting w.r.t. the end-to-end networks is also given in their paper supplements in PAMR [3]. It is interesting to note that, according to the PAMR’s authors, methods such as PSA[2], and IRNet[1], have three stages and additionally train a standalone segmentation network. The end-to-end methods are highlighted in grey. Thanks to MCT-Former, operated in the two versions (boosted and end-to-end, this last indicated by a \star), we can appreciate the difference between the two approaches. Among all methods, our ViT-PCM is second to MCT-Former, boosted with PSA[2]. W.r.t the end-to-end methods ViT-PCM advances the state-of-the-art on all categories and improves the results of 2.1% on the val set and 2.5% on the test set.

4 Qualitative results on MS-COCO val set

In Figure S3, we show some results on the final-segmentation masks for MS-COCO 2014 val set. All the shown images are chosen among those with mIoU% greater than 95% which are about 1500 images. We can note that some results are even better than the ground-truth annotations, such as, for example, the

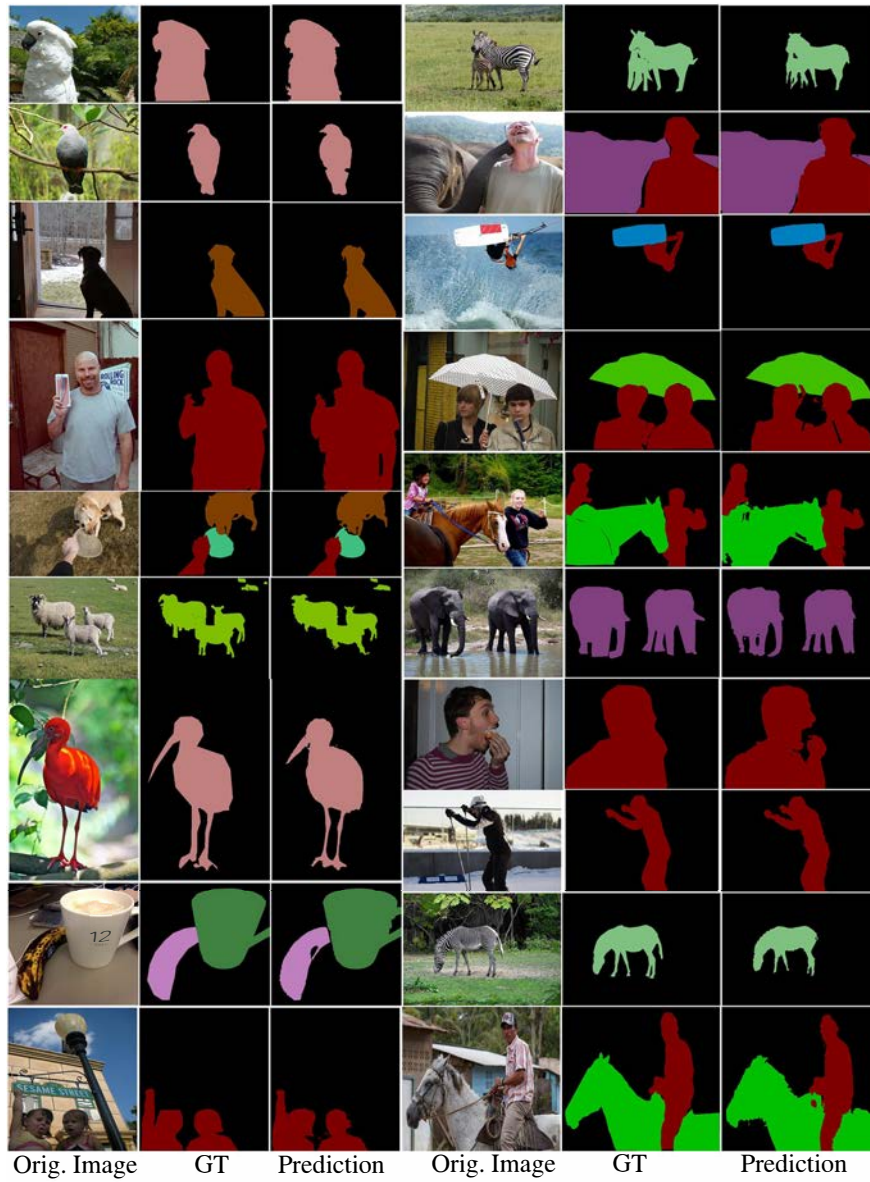


Fig. S3: Qualitative results on MS-COCO 2014 val set.

man-eating or the two elephants, where the annotation does not consider the space inside the shapes, which instead ViT-PCM can recognize.

5 Per-Class Comparisons with state-of-the-art on Verification Task in MS-COCO 2014

Table S2: Per-class performance comparison with the state-of-the-art WSSS methods on the verification task with final-segmentation masks, in terms of mIoU% on MS-COCO 2014 *val* set.

Class	AuxSegNet _{ICCV'21} [15]	MCT-Former _{CVPR'22} [16]	ViT-PCM Ours	Class	AuxSegNet _{ICCV'21} [15]	MCT-Former _{CVPR'22} [16]	ViT-PCM Ours
background	82.0	82.4	81.9	wine-glass	32.1	27.0	38.2
person	65.4	62.6	62.4	cup	29.3	29.	40.9
bicycle	43.0	47.4	54.3	fork	5.4	13.9	33.3
car	34.5	47.2	49.2	knife	1.4	12.0	31.0
motorcycle	66.2	63.7	70.3	spoon	1.4	6.6	21.4
airplane	60.3	64.7	74.5	bowl	19.5	22.4	36.2
bus	63.1	64.5	76.0	banana	46.9	63.2	58.6
train	57.3	64.5	61.2	apple	40.4	44.4	52.1
truck	38.9	44.8	45.3	sandwich	39.4	39.7	57.1
boat	30.1	42.3	47.8	orange	52.9	63.0	55.8
traffic-light	40.4	49.9	22.2	broccoli	36.0	51.2	53.5
fire-hydrant	72.7	73.2	78.8	carrot	13.9	40.0	45.0
stop-sign	40.3	76.6	11.0	hot-dog	46.1	53.0	41.4
parking-meter	59.8	64.4	65.5	pizza	62.0	62.2	77.6
bench	16.0	32.8	42.6	donut	43.9	55.7	39.4
bird	61.0	62.6	67.0	cake	30.6	47.9	63.0
cat	68.6	78.2	20.4	chair	11.4	22.8	35.6
dog	66.9	68.2	71.7	couch	14.5	35.0	41.7
horse	55.6	65.8	68.6	potted-plant	2.1	13.5	37.9
sheep	61.4	70.1	67.2	bed	20.5	48.6	53.2
cow	60.7	68.3	70.4	dining-table	9.5	12.9	29.4
elephant	76.1	81.6	83.3	toilet	57.8	63.1	67.3
bear	73.0	80.1	74.2	tv	36.0	47.9	38.7
zebra	80.8	83.0	72.6	laptop	35.2	49.5	51.7
giraffe	71.6	76.9	67.3	mouse	13.4	13.4	13.9
backpack	11.3	14.6	24.3	remote	23.6	41.9	34.2
umbrella	35.0	61.7	67.7	keyboard	17.9	49.8	65.0
handbag	2.2	4.5	19.4	cellphone	49.9	54.1	56.8
tie	14.7	25.2	19.0	microwave	28.7	38.0	50.2
suitcase	31.7	46.8	47.6	oven	13.3	29.9	35.8
frisbee	1.0	43.8	38.1	toaster	0.0	0.0	13.8
skis	8.1	12.8	20.3	sink	21.0	28.0	14.3
snowboard	7.6	31.4	41.6	refrigerator	16.6	40.1	44.9
sports-ball	28.8	9.2	7.1	book	8.7	32.2	40.6
kite	27.3	26.3	41.5	clock	34.4	43.2	51.3
baseball-bat	2.2	0.9	2.3	vase	25.9	22.6	25.0
baseball-glove	1.3	0.7	5.0	scissors	16.6	32.9	48.1
skateboard	15.2	7.8	10.3	teddy-bear	47.3	61.9	53.9
surfboard	17.8	46.5	45.9	hair-drier	0.0	0.0	13.4
tennis-racket	47.1	1.4	16.1	toothbrush	1.4	12.2	33.1
bottle	33.2	31.1	41.5	mIoU%	33.9	42.0	45.0

In Table S1 we expose the per-class comparison on MS-COCO 2014 val set. We compare our results with AugSegNet [15] and with MCT-Former [16]. ViT-PCM accuracy measured in mIoU% outperforms the other two, though the results per class are highly variable. For example, on the class *stop-sign* we have an accuracy of 11.0% while MCT-Former obtains 76.6% and AugSegNet 40.3%. On the other hand, for tiny objects such as *fork*, *knife* and *spoon*, we obtain

resp. 33.3%, 31.0% and 21.4 against a much lower accuracy obtained by the two competitors.

References

1. Ahn, J., Cho, S., Kwak, S.: Weakly supervised learning of instance segmentation with inter-pixel relations. In: ICCV. pp. 2209–2218 (2019) [7](#)
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) [7](#)
3. Araslanov, N., Roth, S.: Single-stage semantic segmentation from image labels. In: CVPR. pp. 4253–4262 (2020) [7](#)
4. 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) [6](#)
5. 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) [7](#)
6. Fan, J., Zhang, Z., Song, C., Tan, T.: Learning integral objects with intra-class discriminator for weakly-supervised semantic segmentation. In: CVPR (2020) [7](#)
7. Fan, J., Zhang, Z., Tan, T., Song, C., Xiao, J.: Cian: Cross-image affinity net for weakly supervised semantic segmentation. In: AAAI (2020) [7](#)
8. Krähenbühl, P., Koltun, V.: Efficient inference in fully connected crfs with gaussian edge potentials. *Advances in neural information processing systems* **24** (2011) [4](#)
9. 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) [7](#)
10. Lee, J., Kim, E., Yoon, S.: Anti-adversarially manipulated attributions for weakly and semi-supervised semantic segmentation. In: CVPR. pp. 4071–4080 (2021) [7](#)
11. 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) [7](#)
12. 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) [7](#)
13. 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) [7](#)
14. 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) [7](#)
15. 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) [9](#)
16. 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) [7](#), [9](#)
17. Zhang, F., Gu, C., Zhang, C., Dai, Y.: Complementary patch for weakly supervised semantic segmentation. In: ICCV. pp. 7242–7251 (2021) [7](#)