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

Simone Rossetti$^{1,2}$, Dam Zappia$^1$, Marta Sanzari$^2$, Marco Schaerf$^{2,1}$, Fiora 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 $D=\{\langle X_{in}, t \rangle \}$. 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 $C=\{0, 1, \ldots, K\}$. The output of $f$ is a tensor $\hat{Y} \in C^{h \times w}$ 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 ($\mathcal{L}_{MCE}$). 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 ($\text{BCE}$):

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

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):

\[ y_k = \text{GMP}(Z^k) = \max(Z^k) = Z^k_i, \text{ for some } i \in \{0, 1, \ldots, s\} \quad (2) \]

where,

\[ Z^k = \text{softmax}(A^k) = \left( \frac{\exp(A^k_1)}{\sum_{c \in C} \exp(A^k_c)}, \ldots, \frac{\exp(A^k_s)}{\sum_{c \in C} \exp(A^k_c)} \right) \]

and \( A^k_j = F_j W^k \) \( (3) \)

As defined above, \( F_j \) 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 \( k \in \{0, 1, \ldots, K\} \), where the \( k=0 \) class is predicted within the optimization.

The relative error back propagation of \( L_{MCE} \) w.r.t weights \( W \) is given by:

\[ \frac{\partial L_{MCE}}{\partial W} = \sum_{k=0}^{K} \frac{\partial BCE(t_k, y_k)}{\partial y_k} \frac{\partial y_k}{\partial \max(Z^k)} \frac{\partial \max(Z^k)}{\partial A^q} \frac{\partial A^q}{\partial W} \quad (5) \]

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

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

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

By Eq. (7) we instantiate \( \max(Z^k) \) as \( Z^k_1 \). Rewriting \( \frac{\partial \max(Z^k)}{\partial A^q} \) as \( \frac{\partial Z^k_i}{\partial A^q} \) we get the Jacobian of \( \frac{\partial Z^k_i}{\partial A^q} \) as:

\[ \frac{\partial Z^k_i}{\partial A^q} = \left( \frac{\partial Z^k_1}{\partial A^q}, \ldots, \frac{\partial Z^k_i}{\partial A^q}, \ldots, \frac{\partial Z^k_s}{\partial A^q} \right) \quad (8) \]

hence for \( j \in \{0, 1, \ldots, s\} \):

\[ \frac{\partial Z^k_i}{\partial A^q_j} = \begin{cases} 0 & \text{if } i \neq j \\ \frac{\exp(A^k_i)}{\sum_{c \in C} \exp(A^k_c)} - \left( \frac{\exp(A^k_j)}{\sum_{c \in C} \exp(A^k_c)} \right)^2 & \text{if } i = j \text{ and } q = k \\ \frac{\exp(A^k_i)}{\sum_{c \in C} \exp(A^k_c)} - \left( \frac{\exp(A^k_j)}{\sum_{c \in C} \exp(A^k_c)} \right)^2 & \text{if } i = j \text{ and } q \neq k \\ \end{cases} \quad (9) \]
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:

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

Consider the following matrix:

\[
Z = \begin{pmatrix}
Z_0^0 & \cdots & Z_0^q & \cdots & Z_0^K \\
\vdots & \ddots & \vdots & \ddots & \vdots \\
Z_s^0 & \cdots & Z_s^q & \cdots & Z_s^K 
\end{pmatrix}
\]

\[
\text{subject to } \sum_{k=0}^{K} Z_i^k = 1, \quad i=0, \ldots, s
\]

\[
\text{max}(Z^q) = Z_j^q \\
j=0, \ldots, s, \\
q \in \{0, \ldots, K\}
\]  
(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:

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

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

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

\[
GMP(Z) = (\text{max}(Z^0), \text{max}(Z^1), \ldots, \text{max}(Z^K)).
\]  
(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 \( \text{max}(Z^q) = Z_j^q \), for some \( i=0, \ldots, s \), 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:

\[
\frac{\partial A_i^q}{\partial W^q} = F_i.
\]  
(14)

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

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

\[
\frac{\partial BCE(t_k, y_k)}{\partial W^q} = F_i \cdot \begin{cases} 
    y_k - t_k & q = k \\
    Z_i^q \frac{t_k - y_k}{1 - y_k} & q \neq k 
\end{cases}
\]  
(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 \( 0, \ldots, s \) 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:

\[
\frac{\partial L_{MCE}}{\partial W_q} = \frac{\partial BCE(t_q, y_q)}{\partial W_q} + \sum_{k \in C, k \neq q} \frac{\partial BCE(t_k, y_k)}{\partial W_q}
\]

\[
= -F_{i_q}(t_q - y_q) + \sum_{k \in C, k \neq q} F_{i_k} Z_{i_k} t_k - y_k \frac{y_k}{1 - y_k}
\]

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 \( F_{i_u} \) according to the error value \( (t_q - y_q) \), and in the direction of \( F_{i_k} \) according to the term \( Z_{i_k} t_k - y_k \frac{y_k}{1 - y_k} \), for any category \( k \), with \( k \neq q \).

More specifically, when the term \( (t_k - y_k) = 1 \), 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 \( -(Z_{i_k} y_k) \) 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 \( y_k \to 0 \). 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.

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 BPM 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 $60 \times 60$ 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

| Method                  | bird | bear | bicycle | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | mbk | person | plant | sheep | sofa | train | tv |
|------------------------|------|------|----------|------|--------|-----|-----|-----|-------|-----|------|-----|-------|-----|--------|-------|-------|------|-------|----|
| Ours                   | 82.9 | 35.1 | 86.8 | 59.4 | 53.7  | 39.8| 56.2| 79.3| 70.6  | 82.4| 88.1 | 81.1| 82.6  | 81.1| 68.8  | 81.1 | 84.4 | 85.4 | 85.9 | 89.5 |

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 ⋆), 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 [15] | MCT-Former [16] | Ours | AuxSegNet [15] | MCT-Former [16] | 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           | 29.0 | 40.9 |
| bicycle        | 43.0           | 47.4            | 54.3 | fork           | 5.4            | 13.9 | 31.1 |
| 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            | 66.6           | 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           | 79.1            | 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 |
| friege         | 1.8            | 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 |
| kine           | 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 |

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)

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)

3. Araslanov, N., Roth, S.: Single-stage semantic segmentation from image labels. In: CVPR. pp. 4253–4262 (2020)

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)

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)

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. Fan, J., Zhang, Z., Tan, T., Song, C., Xiao, J.: Cian: Cross-image affinity net for weakly supervised semantic segmentation. In: AAAI (2020)

8. Krähenbühl, P., Koltun, V.: Efficient inference in fully connected crfs with gaussian edge potentials. Advances in neural information processing systems 24 (2011)

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)

10. Lee, J., Kim, E., Yoon, S.: Anti-adversarially manipulated attributions for weakly and semi-supervised semantic segmentation. In: CVPR. pp. 4071–4080 (2021)

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)

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)

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)

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)

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)

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)

17. Zhang, F., Gu, C., Zhang, C., Dai, Y.: Complementary patch for weakly supervised semantic segmentation. In: ICCV. pp. 7242–7251 (2021)