Zoom-CAM: Generating Fine-grained Pixel Annotations from Image Labels

Xiangwei Shi, Seyran Khademi, Yunqiang Li, Jan van Gemert
Computer Vision Lab
Delft University of Technology, The Netherlands

***Abstract*—Current weakly supervised object localization and segmentation rely on class-discriminative visualization techniques to generate pseudo-labels for pixel-level training. Such visualization methods, including class activation mapping (CAM) and Grad-CAM, use only the deepest, lowest resolution convolutional layer, missing all information in intermediate layers. We propose Zoom-CAM: going beyond the last lowest resolution layer by integrating the importance maps over all activations in intermediate layers. Zoom-CAM captures fine-grained small-scale objects for various discriminative class instances, which are commonly missed by the baseline visualization methods. We focus on generating pixel-level pseudo-labels from class labels. The quality of our pseudo-labels evaluated on the ImageNet localization task exhibits more than 2.8% improvement on top-1 error. For weakly supervised semantic segmentation our generated pseudo-labels improve a state of the art model by 1.1%.**

I. Introduction

Visual CNN explanation models allow computer-generated labels (pseudo-labels) to replace laborious human annotations. For example, semantic segmentation [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12] requires expensive pixel-level annotations. Such pixel-level annotations can be generated by CNN visualization methods, with the great advantage of only requiring image-level labels, saving huge annotation costs. Our proposed method focuses on generating fine-grained pseudo-labels from class labels and demonstrated on bounding box labels for object localization and segmentation pixel labels.

Excellent recent visual explanations such as Score-CAM [13], Grad-CAM++ [14] and others [15], [16] focus on decision faithfulness (causality); yet do not give high-precision localization maps, see Figure 2. This is a problem, as weakly supervised learning methods require fine-grained localization maps to generate pseudo-labels from class information [17]. Here, we make the observation that current methods use the last convolutional layer (CL) at the lowest resolution. In fact, small objects are eliminated easily after several pooling layers in most CNN models for classification. Our hypothesis is that by including the visualization maps from intermediate CL, the quality of the pseudo-labels can be improved.

The last CL offers the most semantically comprehensive spatial information with the smallest dimensions. Moreover, the deconvolution is straightforward when computing the rate of change in the class output with respect to the last CL, as commonly there are few (or none) nonlinear layers in between, to impair the mapping. Nevertheless, the resolution is severely compromised once the visualization map from the last CL are projected into the input image. This results in coarse visualization with over-highlighted regions in the background or even missing small-scale objects that are completely removed due to several pooling operations. In Grad-Cam [18], there were unsuccessful attempts to go beyond CL as shown in Figure 4.

In this paper we investigate Zoom-cam: Zoomed-in pseudo-labels for weakly supervised learned using just class labels. We bridge the visualization between the last CL to the input image by visualizing and integrating not only the last but all the feature maps from intermediate CL. Our focus is to generate fine-grained pseudo-labels that highlight the class objects accurately in the original image, as illustrated in Figure 1. We have the following contributions:

- Zoom-CAM generates high-resolution visualization maps, capable of identifying several instances of the same class as well as objects with different scales that are often missed by other methods, see Figure 2.
- We introduce an effective gradient back-propagation scheme to obtain weight masks for a linear combination of intermediate feature maps.
- We demonstrate quantitatively that the best explanation belongs to the last CL, yet combining the visualization maps from intermediate layers reduces the noise corresponding to the locality of the gradient flows.
- On the ImageNet localization task we show 2.8% and 3.7% improvement on top-1 and top-5 errors, respectively, when the objects are localized using Zoom-CAM visual explanations compared to Grad-CAM .
- By plugging our method in a state of the art weakly supervised model [7] it improves by 1.1% where our mIoU for visual explanations outperforms [14], [13], [18].

In the rest of this paper, we refer to the heatmaps that are resized to the input image (See Figure 1) as *visual explanations*. The importance weight matrix for aggregated feature maps in CL are called *weigh masks*. Note that *activation units* and *activation neurons* are used interchangeably.

II. Related work

Automatic Pseudo-labels Generation. Our work focuses on generating high-precision visual explanations from class label information to be used for weakly supervised object localization and segmentation models. Earlier practices to retrieve localization information from class labels utilize the intermediate feature maps. Oquab *et al* [19] show the object

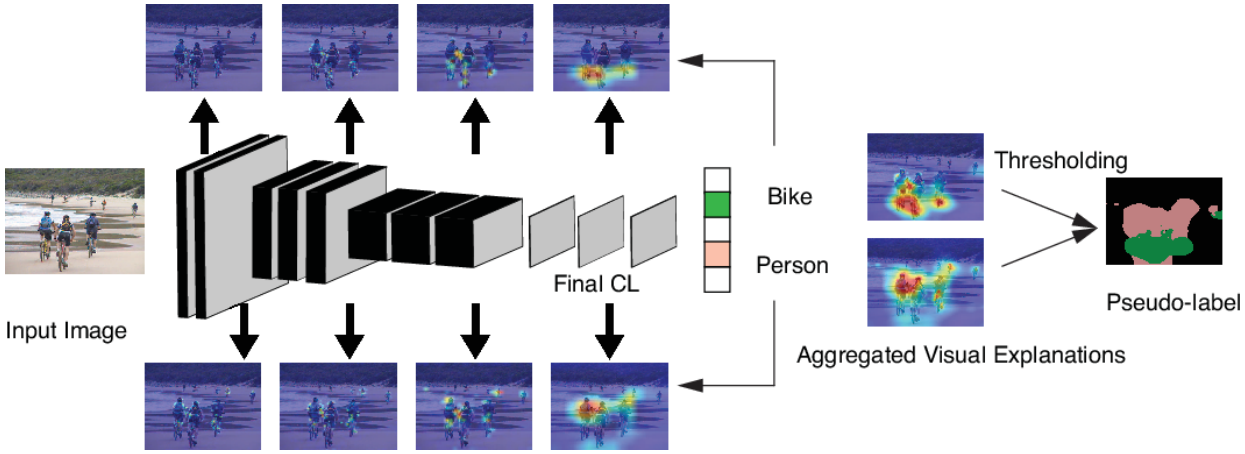


Fig. 1. We generate high-precision pixel-level pseudo-labels for weakly supervised localization and segmentation. We exemplify pseudo-label generating for an example image with "bike" and "person" classes. The pseudo-labels for the intermediate layers are generated by back propagating the gradient of the class score w.r.t. the activations (see Section III) demonstrating localization of the different instances of the class object. Note that the one from only the last convolutional layer over highlights the area around the object and misses the small instances. For clarity, visual explanations are up-sampled to the input size.

localization ability of image classification CNNs by transferring mid-level image representations using a global max pooling layer. Current weakly supervised object segmentation and localization methods [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11], [12], [20], [21] take advantage of visualization techniques such as CAM and Grad-CAM to automatically generate pseudo labels for training purposes. Therefore, the quality of the pseudo labels generated by visualization methods majorly affects the segmentation performance. In this context, a precise and complete visual explanations is of great importance. Wei *et al* [4] aggregated multiple CAM visualizations generated by dilated CNNs with different rates to obtain more accurate regions in visual maps. Lee *et al* [5] randomly selected hidden units from CNNs to generate multiple localization maps that highlight different parts of the class discriminative objects. We evaluate the quality of the generated visualizations by Zoom-Cam, using a weakly supervised segmentation CNN model proposed in [7] as a measure for the precision of visual explanations. the quality of the generated visual explanations is improves in [17] by hiding different parts of the input while generating the visual explanations using CAM. In [17] the final pseudo-labels for object and action localization are generated by aggregating several explanations, in the cost of larger computations.

To the best of our knowledge, all weakly supervised segmentation and localization methods rely on visual explanation techniques to generate the pseudo-labels. We are the first to target completeness in visual explanations for weakly supervised tasks.

Visualizing CNNs. Many explainability work [13], [14], [22], [23], [24] focuses on the faithfulness of the visual explanations to the decision made by a CNN. We noticed that even though these methods are serving their purpose to spot the causal features, they lack the precision and completeness required for accurate pixel-level localization. We discuss related literature on visualizations of CNNs in the following. Inspired by [25], [26], we describe three categories: 1) Deconvolution methods 2) Blind methods and 3) Representation methods.

Deconvolution methods [27], [22], [28] aim at mapping the maximum of activation units in CLs, back to the original images. Zeiler and Fergus [22], in a pioneering work, attempted to interpret the CNN filters by deconvolution to identify what input patterns lead to the maximum activation. Springenberg *et al* [27] proposed guided backpropagation to visualize the feature maps in networks without max-pooling layers (fully convolutional). Even though easily interpretable, Deconvolution methods generate visualizations within several forward and backward passes and thus computationally expensive. The visual explanations generated by Deconvolution are not fine-grained, due to the non-linear nature of CNN models that makes the inverse mapping impossible.

Blind methods follow the black box approach, where the system between input and output is assumed inaccessible. Blind methods [29], [30], [23], [31] generate visual explanations by perturbing (setting pixel intensities to zero, blurring the region or by adding noise) the input in pixel-space to measure the variation in model prediction score. The input regions that increase the classification score (no spatial cues) are reflected in highly activated feature maps with spatial information. [29] covered small-region inputs with patches to identify the highly activated units in the receptive fields. [23] use random binary masks on the entire image. Having infinite options for variations of input space, motivated blind methods to integrate perturbation into loss function [32], [30]. [32] backpropagated the gradients of feature maps to learn a perturbation mask for the input space. Blind methods are known to be reliable and faithful to the underlying model as they capture the CNN response w.r.t. the global changes imposed on the input. Nevertheless, testing different range of variations in the input space impose a great computational burden for generating visual explanations.

Representation methods [33], [34], [18], [14], [35], [13] generate visual explanations based on gradients and/or weights under the assumption that these are accessible. This is in contrast to the blind approach. Commonly, in representation method the gradient flow is used as a local measure that captures the variation of the output w.r.t. the features. A popular technique is the class activation mapping (CAM) proposed by Zhou *et al* [35], which generates visual explanations as a linear weighted combination of activation maps of the last convolutional layer. To do so, the model architecture has to change by replacing the fully connected layers with a global average pooling (AP) layer and subsequent retraining. Selvaraju *et al* [18] proposed gradient-weighted class activation mapping (Grad-CAM), by using the average gradients of target class w.r.t. the last convolutional feature maps to compute the CAM weights without re-training the CNN model. Grad-CAM++ [14], the variant of [18], aims at generating more accurate localization maps by individually weighting the activation units in the last feature map, instead of the global average pooling in [35], [18]. Score-CAM [13] is an input-perturbation-based variant of CAM that measures the class posteriors directly yet relying on the activation units in the ultimate layer. Our Zoom-CAM is the first to exploit intermediate convolutional feature maps for generating pseudo-labels used for weakly supervised learning.

III. Methodology

Our proposed method is inspired by Grad-CAM [18] and its variants, and we use the backward gradients from the class score (before softmax) to weight the neurons in the feature maps. In addition, by extending the gradient flow beyond the LC, Zoom-CAM enables the visualization of any intermediate layers in CNN. This extension is not straightforward and is explained in the following

A. Revisiting CAM and Grad-CAM

Suppose $A_k(i,j)$ is the i,j-th activation in the k-th feature map of the last convolutional layer. Following the definition in CAM and Grad-CAM , the final score for the class c, before softmax, is the weighted sum of the average pooled activation neurons in the last convolutional layer:

$$S^c := \sum_k \alpha_k^c \sum_{i,j} A_k(i,j), \tag{1}$$

where α_k^c are the CAM weights once the network is retrained by replacing the fully connected layers with a global AP layer. Grad-CAM shows that α_k^c can be replaced with the average gradient of the class score w.r.t. the neurons in A_k. Thus, retraining is not required and the gradients can be obtained by single backward pass operation. Accordingly, the visualization map for Grad-CAM is given by:

$$L_{i,j}^c := \text{ReLU}\big(\sum_k \frac{1}{Z}\sum_{i,j} \frac{\partial S^c}{\partial A_k(i,j)} A_k(i,j)\big), \tag{2}$$

where Z is the number of activation units in the feature maps of the last convolutional layer and S^c is the score for class c. The ReLU function in Eq. (2) guarantees that only neurons with positive contribution to the gradient of the class score are considered. Note that by defining $L_{i,j}^c$ from Eq. (1), the sum of the elements in $L_{i,j}^c$ is guaranteed to be equal to the class score S^c. Grad-CAM averages the gradients in the feature maps to get α_k^c as weights for linear combination of feature maps. Instead, we use the back propagated gradients as a *weight mask* (matrix) applied to the feature maps. Careful reformulation of the problem is essential for extracting the weigh masks as explained in the following.

B. Visualizing Intermediate Layers

Here we go beyond Grad-Cam and describe how Zoom-Cam visualizes the intermediate layers. Suppose $B_p(m,n)$ is the m,n-th activation in the p-th feature map of the penultimate convolutional layer. Based on the common operational units in a forward pass of a CNN model, B_p is passed through a non-linear function f such as ReLU, sigmoid, tanh, etc. Then $f(B_p)$ for all $p = 1, 2, ..., P$ in the penultimate convolutional layer is convolved with the k-th filter and summed over p to generate the last convolutional feature map, *i.e.*, A_k. Given that convolution is a linear operation, one can write the sum over the activation units in A_k as a weighted sum of $f(B_p(m,n))$:

$$\sum_{i,j} A_k(i,j) = \sum_{m,n} W_k(m,n) \sum_p f(B_p(m,n)). \tag{3}$$

The elements in matrix W_k are the sum over subset of filter weights for k-th kernel. In turn, the filter weights in each subset is a function of m,n-th position in B_p, the size of the kernel, the stride and the padding of the convolution.

Substituting Eq. (3) in Eq. (1) yields the following

$$S^c := \sum_k \alpha_k^c \sum_{m,n} W_k(m,n) \sum_p f(B_p(m,n)). \tag{4}$$

Similar to Eq. (2), the visualization map of the penultimate convolutions is defined by removing the summation over m,n in Eq. (4), resulting in

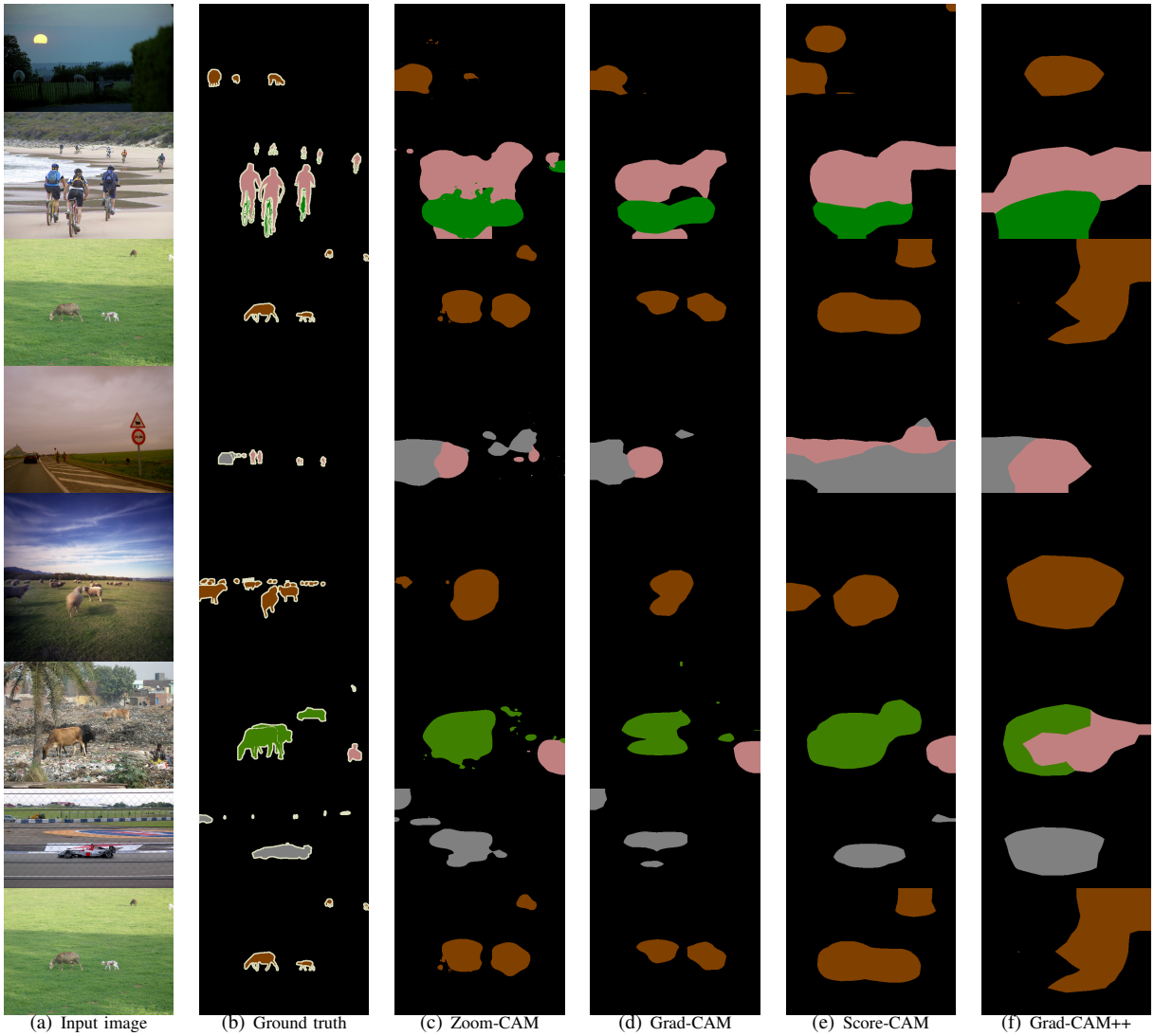
$$L_{m,n}^c := \sum_k \alpha_k^c W_k(m,n) \sum_p f(B_p(m,n)). \tag{5}$$

Note that the nonlinear function f including the ReLU function, batch-norm, or pooling layer, is fixed in the backward pass as we are not in the training process. Thus, $f(B_p)$ can be replaced with a Hadamard product of matrix $N_p \in \mathbb{R}^{m,n}$, representing the nonlinear operation, and $B_p \in \mathbb{R}^{m,n}$. Let us define $F^c \in \mathbb{R}^{m,n}$ as the matrix representation of the penultimate convolutional layer after non-linearity in the backward pass

$$F^c(m,n) := \sum_p N_p(m,n)(B_p(m,n)). \tag{6}$$

From the chain rule, one can write the gradient of the final score for class c w.r.t. the represented penultimate convolutional layer as

$$\frac{\partial S^c}{\partial F^c(m,n)} = \frac{\partial S^c}{\partial \sum_{i,j} A_k(i,j)} \times \frac{\partial \sum_{i,j} A_k(i,j)}{\partial F^c(m,n)}. \tag{7}$$



(a) Input image (b) Ground truth (c) Zoom-CAM (d) Grad-CAM (e) Score-CAM (f) Grad-CAM++

Fig. 2. Examples of pseudo-labels generated by Zoom-CAM visual explanations (ours), Grad-CAM [18], Score-CAM [13] and Grad-CAM++[14]. Zoom-CAM aggregates visual maps through all intermediate layers, which captures objects with different scales and several instances of the same class. The over-highlighted regions relate to false positive. Zoom-CAM can generate fine-grained pseudo-labels by increasing the true positive and reducing the false positive.

The right side of Eq. (7) is carefully decomposed to two terms. The first term is the scaled weights of Grad-CAM layer and the second term can be derived from Eq. (3). Particularly, for a ReLU activation function, the elements of matrix N_p in Eq. (6) are zeros and ones, thus, Eq. (7) boils down to

$$\frac{1}{Z}\frac{\partial S^c}{\partial \sum_p B_p(m',n')} = \alpha_k^c\, W_k(m',n'). \tag{8}$$

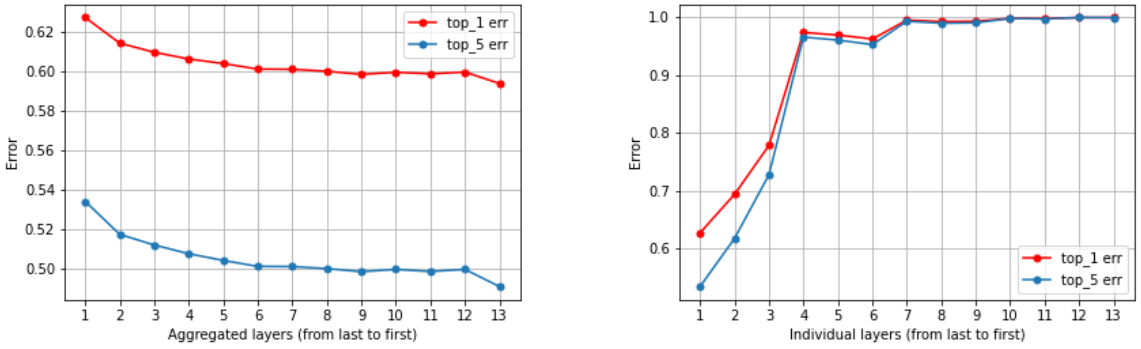
where m', n' indicates the positive elements that are passed by ReLU function. Comparing Eq. (5) and Eq. (8), reveals that the weights for visualization map of the penultimate convolutional layer are the gradients of the final score w.r.t. the features. We refer to the $W_k(m',n')$ as weight masks that are applied (point-wise multiplication) to the feature maps. This is in contrast to the Grad-CAM approach which uses the scalar weighting of the feature maps.

The final visual explanation for Zoom-CAM , is calculated by considering only the positive values in $L^c_{m,n}$ because we are only interested in the activation neurons whose intensity should be increased in order to increase the class score.

$$L^c_{m,n} := \text{ReLU}\big(\frac{1}{Z}\sum_k\sum_p \frac{\partial S^c}{\partial \sum_p B_p(m',n')} B_p(m',n')\big). \tag{9}$$

Eq. (9) can be extended for any intermediate CNN layer, by replacing B_p by that layer. In Grad-CAM , a single backward pass up to the last CL, is performed to calculate the gradients. In practice, $W_k(m,n)$ is computed in the backward pass from the AP layer of Grad-CAM to the target layer.

A Grad-CAM [18] extension visualizes intermediate convolutional layers by average pooling the elements of an intermediate feature maps. In contrast, Zoom-CAM is using Eq. (8) to calculate individual weights for different elements



(a) (b)

Fig. 3. Top-1 and top-5 localization error rates on ILSVRC2012 *val* dataset for ablation study. (a) Aggregating intermediate feature maps can consistently improve the weakly supervised object localization ability, especially when the last two layers are integrated. (b) The localization error rates of Zoom-CAM maps using feature maps from a single intermediate layer. The feature maps of the last layer contributes the most to the performance of object localization.

TABLE I
CLASSIFICATION AND LOCALIZATION ERROR RATES (%) ON ILSVRC2012 VAL DATASET FOR PRE-TRAINED VGG16 FROM PYTORCH. ZOOM-CAM BY AGGREGATING VISUALIZATION MAPS FROM INTERMEDIATE LAYERS ACHIEVES BETTER PERFORMANCE ON OBJECT LOCALIZATION THAN GRAD-CAM. THE THRESHOLDS FOR ZOOM-CAM AND GRAD-CAM MAPS ARE 25% AND 15% OF THE MAXIMUM VALUE. ZOOM-CAM ACHIEVES LOWER ERROR.

	Classification error		Localization error	
	Top-1	Top-5	Top-1	Top-5
Zoom-CAM	31.87	11.54	**59.11**	**48.64**
Grad-CAM	31.87	11.54	61.95	52.35

of intermediate feature maps. The visual explanations for different intermediate layers produced by Grad-CAM and Zoom-CAM are presented in Figure 4.

C. Aggregation of Localization Maps

After generating intermediate layer visualization maps of Zoom-CAM via Eq. (9), we need to aggregate these maps and up-sample them to the input image resolution.

Given two Zoom-CAM visualization maps for different intermediate layers, $L^c_{i,j}$ and $L^c_{m,n}$, where $i \leq m$ and $j \leq n$, the first step is the normalization. visualization maps are normalized such that the values over single localization map range from 0 to 1. Next, we up-sample $L^c_{i,j}$ (smaller feature map) to the size of $L^c_{m,n}$ through bilinear interpolation. Finally, the aggregated visualization maps L^c will be obtained by:

$$\hat{L}^c_{m,n} = \max\{\boldsymbol{N}(L^c_{m,n}), \boldsymbol{U}(\boldsymbol{N}(L^c_{i,j}))\}, \tag{10}$$

where $\boldsymbol{U}(\cdot)$ and $\boldsymbol{N}(\cdot)$ denotes the up-sampling and normalization operations, respectively.

Taking the maximum in Eq. (10) is a simple operation that will preserve the importance of visualization maps, that is reflected by the normalized values. Taking the average is another option but we observed the smoothing of the weights across the layers, which is not desirable for generating crisp visualization maps.

IV. EXPERIMENTS

We evaluate the quality of the generated visual explanations by Zoom-Cam. We mostly follow the validation framework from Grad-CAM and CAM by evaluating on weakly supervised object localization and segmentation tasks, on ImageNet and PASCAL VOC datasets, respectively. Moreover, sample visualization are reported in the supplementary material for qualitative inspection of the results.

A. Weakly Supervised Object Localization

We evaluate weakly supervised object localization using the visual explanations generated by Zoom-CAM. We use a pre-trained VGG16 as a baseline on the ILSVRC2012 [36] val dataset. We resize images to $224 \times 224 \times 3$ and color normalize the mean and the standard deviation. We generate Zoom-CAM saliency map in addition to the class prediction. The pixels with higher value than 25% of the max intensity are preserved, which constructs several connected regions. We keep the largest connected component and draw a bounding box around it. This bounding box reveals the location of the classified object. We follow the evaluation metrics of ILSVRC2012 object localization task and report the top-1 and top-5 classification and localization error in Table I. For localization score, the prediction counts when the classification prediction matches the ground truth image label and the predicted bounding box has over 50% overlap with the ground truth bounding box. The results on the ImageNet localization task exhibits around 2.84% improvement on top-1 error after aggregating all intermediate layers. For both Zoom-CAM and Grad-CAM we use the same CNN model for classification and therefore the classification scores are the same.

1) Ablation Studies: Zoom-CAM aggregates feature maps of all 13 intermediate layers in VGG16. We conduct ablation experiments by aggregating different numbers of intermediate layers including only a single intermediate layer. This is to quantify the contribution of each layer to the accuracy of generated visualizations by Zoom-CAM in terms of weakly

TABLE II

COMPARISON OF QUALITY OF PSEUDO-SEGMENTATION-LABELS OF PASCAL VOC 2012 *val* SET MEASURED IN IOU (%). THE BASE MODEL IS A FINE-TUNED RESNET50, TRAINED ON IMAGE CLASS LABELS. THE ACCURACY OF ZOOM-CAM PSEUDO-LABELS COMPARES FAVORABLY TO OTHERS.

Method	IoU																					mIoU
	backgr	plane	bike	bird	boat	bottle	bus	car	dog	chair	cow	dtable	cat	horse	motor	person	plant	sheep	sofa	train	tv	
Grad-CAM++	64.7	27.8	17.8	25.0	23.8	31.6	47.2	38.8	46.6	18.4	42.1	32.5	40.8	40.0	41.6	32.2	26.8	39.6	33.3	42.1	32.9	35.5
Grad-CAM	66.5	29.7	18.3	25.5	19.3	33.6	51.0	42.4	49.0	19.2	41.2	36.7	41.6	40.5	43.6	41.9	28.9	39.8	34.2	39.3	36.5	37.1
Score-CAM	68.1	**31.8**	19.1	**29.7**	**29.3**	30.9	50.3	**45.3**	47.9	19.8	41.8	32.3	**44.7**	42.0	**47.2**	35.4	27.9	42.8	36.6	**47.1**	31.8	38.2
Zoom-CAM	**68.9**	31.0	**19.7**	26.9	20.6	**34.5**	50.3	42.3	**50.1**	**20.4**	**45.6**	35.3	43.2	**43.8**	46.0	42.0	**31.1**	**45.0**	**38.3**	40.1	**38.6**	**38.8**

TABLE III

QUALITY OF PSEUDO SEMANTIC SEGMENTATION LABELS IN MIOU, EVALUATED ON THE AUGMENTED PASCAL VOC 2012 *train* SET.

Method	mIoU
CAM	48.3 [7]
Zoom-CAM	**49.0**

TABLE IV

SEMANTIC SEGMENTATION PERFORMANCE IN MIOU EVALUATED ON THE PASCAL VOC 2012 *val* SET. THE PERFORMANCE OF WSSS USING PSEUDO-LABELS GENERATED BY ZOOM-CAM IS BETTER THAN THE ONE BY CAM.

Method	*val*
IRNet(ResNet50)-CAM	63.5
IRNet(ResNet50)-Zoom-CAM	**64.6**

supervised object localization. Fig 3 (a) shows that aggregating intermediate layers consistently improves the performance of weakly supervised object localization, especially when the last two layers are integrated. Fig 3 (b) shows the top-1 and top-5 localization errors for Zoom-CAM using the feature maps of only single intermediate layer. As expected the last feature map contributes the most to the performance in object localization.

B. Weakly Supervised Semantic Segmentation

We evaluate the visualization maps generated by Zoom-CAM on weakly supervised semantic segmentation (WSSS) task on PASCAL VOC 2012 [37] dataset. Although the dataset contains semantic and instance segmentation labels, we only take advantage of image-level class labels. The training set for semantic segmentation is augmented by [38], which contains 10,582 images. The original *val* set with 1,449 images are used for validation.

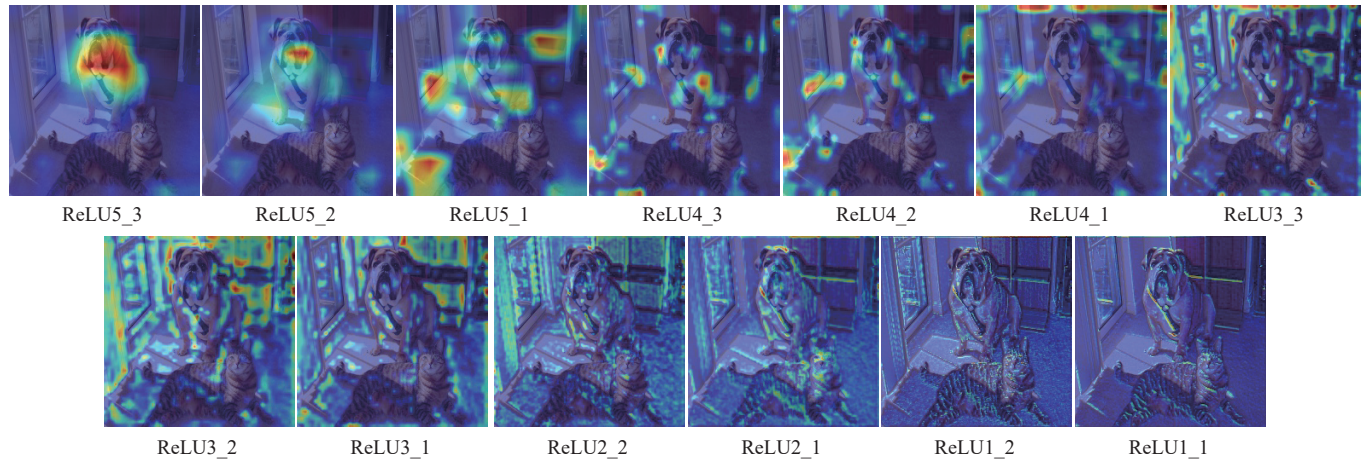
The task of weakly supervised segmentation leverages the image-level class information to segment objects, including semantic and instance segmentation. Recent works on weakly supervised segmentation use CAM or Grad-CAM to generate pseudo-segmentation-labels for training purposes. Therefore, weakly supervised segmentation models are sensitive to the quality of generated pseudo-labels by the visualization techniques. We first compare the quality of pseudo-labels obtained by Zoom-CAM and other visualization methods.

1) Quality of Pseudo-labels: To evaluate the quality of pseudo-labels, we generate saliency maps for each image in the *val* set of PASCAL VOC 2012 via Zoom-CAM. We take pre-trained ResNet50 on ImageNet [36] as the base model and fine-tune it on the classification set of PASCAL VOC 2012. The mean average precision (mAP) is 94.1% for the classification task evaluated on *val* set of PASCAL VOC 2012 classification task. Similarly, we take a threshold, 25% of the max intensity for Zoom-CAM, on the saliency maps and search the largest connected component. Because images have multiple labels, we threshold and fuse the saliency maps by comparing saliency values of multiple labels pixel-wise as the final pseudo-labels.

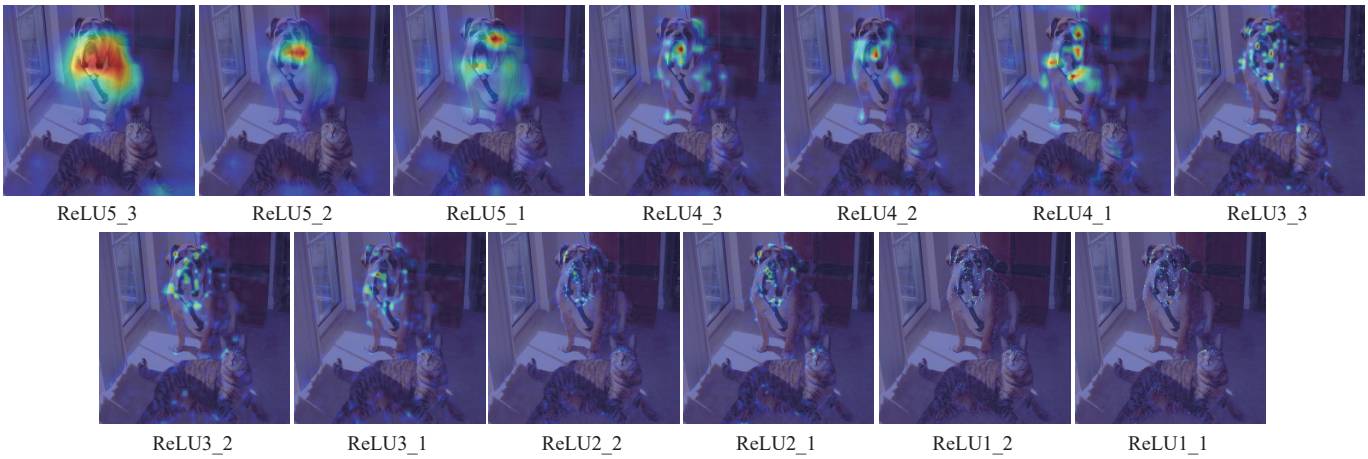
Table II shows the results for pseudo-segmentation-labels using mean Intersection of Union (mIoU) as evaluation metric. For a single class, the quality of pseudo-labels is evaluated by Intersection of Union (IoU). We can see that adding intermediate featuremaps by Zoom-Cam compares favorably to others.

Fig 2 shows sample visual explanations w.r.t. the semantic object segmentation ground truth. These examples confirm that Zoom-CAM captures objects with different sizes, which are commonly lost in the last low-resolution convolutional layer. Interestingly, Grad-CAM outperforms its recent variants such as Score-CAM and Grad-CAM++ once inspected visually in PASCAL VOC 2012 dataset. This is consistent with the results in Table II where pseudo-labels generated by Grad-CAM++ achieve lowest mIoU.

2) Training the Weakly Supervised Semantic Segmentation (WSSS) Baseline with Zoom-CAM Pseudo-labels: We re-train the s.o.t.a. weakly supervised semantic segmentation model [7] with generated psudo-labels by Zoom-CAM . We show in Table II that Zoom-CAM generates pseudo-labels with higher quality so we expect to see improvement in WSSS baseline when trained by Zoom-CAM pseudo-labels. [7] trains ResNet50 from scratch for the classification task of PASCAL VOC 2012. Then they use CAM to generate pseudo-labels for the training of their segmentation model. We replace CAM pseudo-labels with Zoom-CAM pseudo-labels and re-train the WSSS CNN model referred to as IRNet in [7]. Table III shows the quality of pseudo-semantic-segmentation labels in mIoU, evaluated on the PASCAL VOC 2012 segmentation *train* set. The quality of pseudo-labels generated by Zoom-CAM is better than the ones generated by CAM reported



(a) Grad-CAM



(b) Zoom-CAM

Fig. 4. Comparison of visual explanation of single intermediate convolutional layer generated by Grad-CAM and Zoom-CAM. The images for each method represents using the feature maps from the last convolutional layer to the first one. For the same original image (both dog and cat), these saliency maps are generated w.r.t the 'bull mastiff' label. The basic model is pre-trained VGG16 model from PyTorch. Zoom-CAM is using Eq. (8) to calculate the different weights for different elements of intermediate feature maps, while Grad-CAM takes the average of Eq. (8) for all elements of an intermediate feature maps.

in [7], therefore we expect better performance of IRNet on segmentation task once trained with Zoom-CAM pseudo-labels. Finally, Table IV shows the performance of IRNet using pseudo-labels generated by Zoom-CAM and CAM, which confirms our speculation. This is the ultimate experiment to quantify the effect of more precise visualization maps in a down-stream task such as WSSS. We observed that the mIoU evaluated on PASCAL VOC 2012 *val* set for the re-trained model by Zoom-CAM pseudo-labels improved by 1.1%.

V. Conclusion

We presented Zoom-CAM to generate high-quality pseudo-labels by integrating visual maps over all intermediate layers in classification CNNs. Zoom-CAM is a generalization of Grad-CAM but differently we use weight masks to linearly combine the feature maps at any intermediate CL. The results verify our hypothesis that intermediate layers offer more accurate localization of the object, in CNNs. The computation time of Zoom-CAM visualization is mostly dominated by the time of back-propagation, which is the same as Grad-CAM. We would like to evaluate the faithfulness of our generated visual explanations to the model prediction as well.

References

[1] A. Kolesnikov and C. H. Lampert, "Seed, expand and constrain: Three principles for weakly-supervised image segmentation," in *European Conference on Computer Vision*. Springer, 2016, pp. 695–711.

[2] Y. Wei, J. Feng, X. Liang, M.-M. Cheng, Y. Zhao, and S. Yan, "Object region mining with adversarial erasing: A simple classification to semantic segmentation approach," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1568–1576.

[3] X. Zhang, Y. Wei, J. Feng, Y. Yang, and T. S. Huang, "Adversarial complementary learning for weakly supervised object localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1325–1334.

[4] Y. Wei, H. Xiao, H. Shi, Z. Jie, J. Feng, and T. S. Huang, "Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7268–7277.

[5] J. Lee, E. Kim, S. Lee, J. Lee, and S. Yoon, "Ficklenet: Weakly and semi-supervised semantic image segmentation using stochastic inference," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5267–5276.

[6] I. H. Laradji, D. Vazquez, and M. Schmidt, "Where are the masks: Instance segmentation with image-level supervision," *arXiv preprint arXiv:1907.01430*, 2019.

[7] J. Ahn, S. Cho, and S. Kwak, "Weakly supervised learning of instance segmentation with inter-pixel relations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2209–2218.

[8] J. Ahn and S. Kwak, "Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4981–4990.

[9] Y. Zhou, Y. Zhu, Q. Ye, Q. Qiu, and J. Jiao, "Weakly supervised instance segmentation using class peak response," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3791–3800.

[10] Y. Zeng, Y. Zhuge, H. Lu, and L. Zhang, "Joint learning of saliency detection and weakly supervised semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 7223–7233.

[11] Y. Wang, J. Zhang, M. Kan, S. Shan, and X. Chen, "Self-supervised scale equivariant network for weakly supervised semantic segmentation," *arXiv preprint arXiv:1909.03714*, 2019.

[12] D. Wang, B. Wang, and Y. Zhou, "Twinsadvnet: Adversarial learning for semantic segmentation," in *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2019, pp. 1–4.

[13] H. Wang, M. Du, F. Yang, and Z. Zhang, "Score-cam: Improved visual explanations via score-weighted class activation mapping," *arXiv preprint arXiv:1910.01279*, 2019.

[14] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018, pp. 839–847.

[15] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds. Curran Associates, Inc., 2017, pp. 4765–4774. [Online]. Available: http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf

[16] S. Xu, S. Venugopalan, and M. Sundararajan, "Attribution in scale and space," 2020.

[17] K. K. Singh and Y. J. Lee, "Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization," *CoRR*, vol. abs/1704.04232, 2017. [Online]. Available: http://arxiv.org/abs/1704.04232

[18] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 618–626.

[19] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Learning and transferring mid-level image representations using convolutional neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1717–1724.

[20] A. Gudi, N. van Rosmalen, M. Loog, and J. C. van Gemert, "Object-extent pooling for weakly supervised single-shot localization," in *British Machine Vision Conference*, 2017.

[21] S. Yang, Y. Kim, Y.-K. Kim, and C. Kim, "Combinational class activation maps for weakly supervised object localization," *ArXiv*, vol. abs/1910.05518, 2019.

[22] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[23] V. Petsiuk, A. Das, and K. Saenko, "Rise: Randomized input sampling for explanation of black-box models," *arXiv preprint arXiv:1806.07421*, 2018.

[24] T. Viering, Z. Wang, M. Loog, and E. Eisemann, "How to manipulate cnns to make them lie: the gradcam case," 2019.

[25] M. Du, N. Liu, and X. Hu, "Techniques for interpretable machine learning," *Commun. ACM*, vol. 63, pp. 68–77, 2019.

[26] Q. Zhang and S.-C. Zhu, "Visual interpretability for deep learning: a survey," *Frontiers of Information Technology & Electronic Engineering*, vol. 19, pp. 27–39, 2018.

[27] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," *arXiv preprint arXiv:1412.6806*, 2014.

[28] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," *arXiv preprint arXiv:1312.6034*, 2013.

[29] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Object detectors emerge in deep scene cnns," *arXiv preprint arXiv:1412.6856*, 2014.

[30] P. Dabkowski and Y. Gal, "Real time image saliency for black box classifiers," in *Advances in Neural Information Processing Systems*, 2017, pp. 6967–6976.

[31] J. Wagner, J. M. Kohler, T. Gindele, L. Hetzel, J. T. Wiedemer, and S. Behnke, "Interpretable and fine-grained visual explanations for convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 9097–9107.

[32] R. C. Fong and A. Vedaldi, "Interpretable explanations of black boxes by meaningful perturbation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3429–3437.

[33] M. Du, N. Liu, Q. Song, and X. Hu, "Towards explanation of dnn-based prediction with guided feature inversion," *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2018.

[34] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?": Explaining the predictions of any classifier," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '16. New York, NY, USA: Association for Computing Machinery, 2016, p. 1135–1144. [Online]. Available: https://doi.org/10.1145/2939672.2939778

[35] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2921–2929.

[36] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[37] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[38] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik, "Semantic contours from inverse detectors," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 991–998.