Rethinking Class Activation Mapping for Weakly Supervised Object Localization

Wonho Bae, Junhyug Noh, and Gunhee Kim(✉)

Department of Computer Science and Engineering,
Seoul National University, Seoul, Korea
bwh0324@gmail.com, jh.noh@vision.snu.ac.kr, gunhee@snu.ac.kr
http://vision.snu.ac.kr/projects/rethinking-cam-wsol

Abstract. Weakly supervised object localization (WSOL) is a task of localizing an object in an image only using image-level labels. To tackle the WSOL problem, most previous studies have followed the conventional class activation mapping (CAM) pipeline: (i) training CNNs for a classification objective, (ii) generating a class activation map via global average pooling (GAP) on feature maps, and (iii) extracting bounding boxes by thresholding based on the maximum value of the class activation map. In this work, we reveal the current CAM approach suffers from three fundamental issues: (i) the bias of GAP that assigns a higher weight to a channel with a small activation area, (ii) negatively weighted activations inside the object regions and (iii) instability from the use of the maximum value of a class activation map as a thresholding reference. They collectively cause the problem that the localization to be highly limited to small regions of an object. We propose three simple but robust techniques that alleviate the problems, including thresholded average pooling, negative weight clamping, and percentile as a standard for thresholding. Our solutions are universally applicable to any WSOL methods using CAM and improve their performance drastically. As a result, we achieve the new state-of-the-art performance on three benchmark datasets of CUB-200–2011, ImageNet-1K, and OpenImages30K.

Keywords: Weakly Supervised Object Localization (WSOL) · Class Activation Mapping (CAM)

1 Introduction

Many recent object detection algorithms such as Faster R-CNN [27], YOLO [25], SSD [22], R-FCN [7] and their variants [11,20,26] have been successful in

W. Bae and J. Noh—Equal contribution.

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-58555-6_37) contains supplementary material, which is available to authorized users.

A. Vedaldi et al. (Eds.): ECCV 2020, LNCS 12360, pp. 618–634, 2020.
https://doi.org/10.1007/978-3-030-58555-6_37

challenging benchmarks of object detection [10,21]. However, due to the necessity of heavy manual labor for bounding box annotations, weakly supervised object localization (WSOL) has drawn great attention in computer vision research [5,6,31,36,38–40]. Contrast to fully-supervised object detection, the models for WSOL are trained for the objective of classification solely relying on image-level labels. They then utilize the feature map activations from the last convolutional layer to generate class activation maps from which bounding boxes are estimated.

Since CAM approach [40] was initially introduced, most of previous studies on WSOL have followed its convention to first generate class activation maps and extract object locations out of them. However, this approach suffers from severe underestimation of an object region since the discriminative region activated through the classification training is often much smaller than the object's actual region. For instance, according to the class activation map ($\mathbf{M}_k$) in Fig. 1, the classifier focuses on the *head* of the *monkey* rather than its whole *body*, since the activations of the *head* are enough to correctly classify the image as *monkey*. Thus, the bounding box reduces to delineate the small highly activated *head* region only. To resolve this problem, recent studies have devised architectures to obtain larger bounding boxes; for example, it erases the most discriminative region and trains a classifier only using the regions left, expecting the expansion of activation to the next most discriminative regions [1,5,6,13,17,18,31,34,35,38]. These methods have significantly improved the performance of WSOL as well as other relevant tasks such as semantic segmentation.

In this work, however, we propose an approach different from the previous researches; instead of endeavoring to expand activations by devising a new architecture, we focus on correctly utilizing the information that already exists in the feature maps. The major contribution of our approach is three-fold.

1. We discover three underlying issues residing in the components of the CAM pipeline that hinder from properly utilizing the information from the feature maps for localization. Our thorough analysis on CAM reveals the mechanism of how each component of CAM negatively affects the localization to be limited to small discriminative regions of an object.
2. Based on the analysis, we propose three simple but robust techniques that significantly alleviate the problems. Since our solution does not introduce any new modules but replaces some of existing operations for pooling, weight averaging and thresholding with better ones, it is easily applicable to any CAM-based WSOL algorithms.
3. In our experiments, we show that our solutions significantly improve multiple state-of-the-art CAM-based WSOL models (*e.g.* HaS [31] and ADL [6]). More encouragingly, our approach achieves the new best performance on two representative benchmarks: CUB-200–2011 [33] and ImageNet-1K [28], and one recently proposed one: OpenImages30K [2,4].

2 Approach

In this section, we first outline three fundamental problems of CAM-based approach to WSOL that cause the localization to be limited to small discriminative

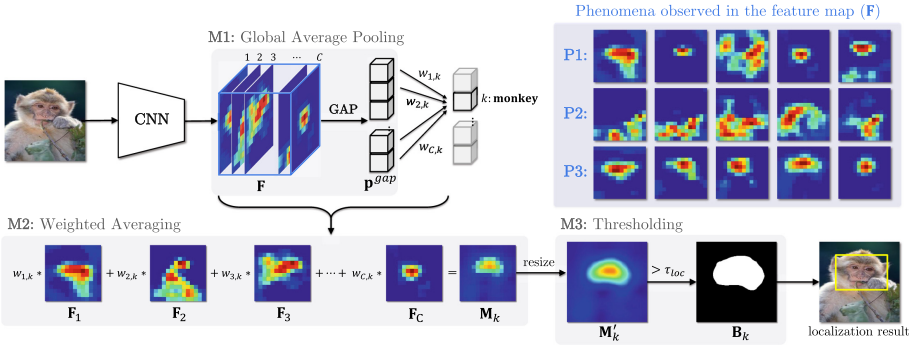


Fig. 1. Overview of the CAM pipeline. We investigate three phenomena of the feature maps (**F**). **P1.** The areas of the activated regions largely differ by channel. **P2.** The activated regions corresponding to the negative weights ($w_c < 0$) often cover large parts of the target object (*e.g. monkey*). **P3.** The most activated regions of each channel largely overlap at small regions. The three modules of CAM in gray boxes (**M1**–**M3**) do not take these phenomena into account correctly. It results in localization being limited to small discriminative regions.

regions of an object. To this end, three phenomena are visualized with feature maps in **P1**–**P3** of Fig. 1, and the corresponding modules where the problems occur related to the phenomena are described in **M1**–**M3**.

(i) **Global Average Pooing.** In practice, the areas of the activated regions largely differ by feature map channel. But, Global Average Pooling (GAP) is biased to assign a higher weight to a channel with small activated area. It results in the small region to be more focused when generating a class activation map.
(ii) **Weighted Averaging.** Ideally, the activated regions in the channel of a feature map corresponding to a negative weight are supposed to be *no-object regions* (*e.g.* background); however, they often occur inside the object, especially less import regions (*e.g. monkey*'s *body*). As a result, less important object regions are further suppressed in the class activation map.
(iii) **Thresholding.** The most activated regions largely overlap across different channels. Since a class activation map is generated by weighted-averaging all the channels and a bounding box is determined based on the threshold proportional to the maximum value of the class activation map, small overlapped regions with too high activations become overdominant to the localization.

Before presenting our solutions to the problems, we first review the class activation mapping (CAM) pipeline in Sect. 2.1. We then elaborate the problems and our solutions one by one in the following Sect. 2.2, 2.3 and 2.4.

2.1 Preliminary: Class Activation Mapping (CAM)

The current CAM approach based on the CNN trained for classification, generates a class activation map and localizes an object in the following way (Fig. 1).

Let the feature map be $\mathbf{F} \in \mathbb{R}_{\geq 0}^{H \times W \times C}$ where $\mathbb{R}_{\geq 0}$ is a non-negative real number. $\mathbf{F}_c \in \mathbb{R}_{\geq 0}^{H \times W}$ denotes c-th channel of $\mathbf{F}$ where $c = 1, \ldots, C$. First, $\mathbf{F}$ is passed into a global average pooling (GAP) layer that averages each $\mathbf{F}_c$ spatially and outputs a pooled feature vector, $\mathbf{p}^{\text{gap}} \in \mathbb{R}_{\geq 0}^{C}$ as

$$p_c^{\text{gap}} = \frac{1}{H \times W} \sum_{(h,w)} \mathbf{F}_c(h, w), \tag{1}$$

where p_c^{gap} denotes a scalar of $\mathbf{p}^{\text{gap}}$ at c-th channel, and $\mathbf{F}_c(h, w)$ is an activation of $\mathbf{F}_c$ at spatial position (h, w).

The pooled feature vector is then transformed into K-dim logits through an FC layer where K is the number of classes. We denote the weights of the FC layer as $\mathbf{W} \in \mathbb{R}^{C \times K}$. Hence, the class activation map $\mathbf{M}_k$ for class k becomes

$$\mathbf{M}_k = \sum_{c=1}^{C} w_{c,k} \cdot \mathbf{F}_c, \tag{2}$$

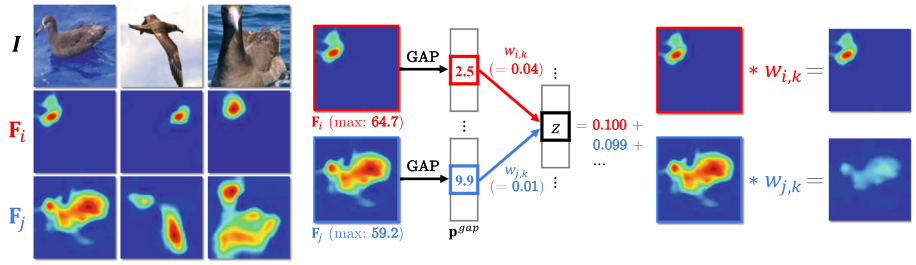
where $\mathbf{M}_k \in \mathbb{R}^{H \times W}$ and $w_{c,k}$ is an (c, k) element of $\mathbf{W}$. For localization, $\mathbf{M}'_k$ is first generated by resizing $\mathbf{M}_k$ to the original image size. Then a localization threshold is computed as

$$\tau_{loc} = \theta_{loc} \cdot \max \mathbf{M}'_k, \tag{3}$$

where $\theta_{loc} \in [0, 1]$ is a hyperparameter. Next, a binary mask $\mathbf{B}_k$ identifies the regions where the activations of $\mathbf{M}'_k$ is greater than τ_{loc}: $\mathbf{B}_k = \mathbb{1}(\mathbf{M}'_k > \tau_{loc})$. Finally, the localization is predicted as the bounding box that circumscribes the contour of the regions with the largest positive area of $\mathbf{B}_k$.

2.2 Thresholded Average Pooling (TAP)

Problem. In a feature map ($\mathbf{F}$), the activated areas largely differ by channel as each channel captures different class information. The GAP layer, however, does not reflect this difference. It naively sums all the activations of each channel and divides them by $H \times W$ without considering the activated area in the channel as in Eq.(1). The difference in the activated area per channel is, however, not negligible. As an example in Fig. 2, suppose i-th channel $\mathbf{F}_i$ in (a) captures the *head* of a *bird* while j-th channel $\mathbf{F}_j$ captures its *body*. Although the area activated in $\mathbf{F}_i$ is much smaller than that in $\mathbf{F}_j$, the GAP layer divides both of them by $H \times W$, and thus the pooled feature value p_i^{gap} of $\mathbf{F}_i$ is also much smaller than p_j^{gap}. However, it does not mean the importance of $\mathbf{F}_i$ for classification is less than $\mathbf{F}_j$. For the GT class k (*bird*), to compensate this difference, the FC weight $w_{i,k}$ corresponding to $\mathbf{F}_i$ is trained to be higher than $w_{j,k}$. As a result,



(a) Two features (*head* and *body*) (b) Classification phase (c) Localization phase

Fig. 2. An example illustrating a problem of the GAP layer. (a) $\mathbf{F}_i$ and $\mathbf{F}_j$ are the features capturing the *head* and *body* of a *bird*, respectively. (b) When the two features are passed to the GAP layer, although their max values are similar, the pooled feature values, p_i^{gap} and p_j^{gap}, are significantly different $(2.5, 9.9)$. Despite the similar contributions of two features to the logit (z) as $(0.100, 0.099)$, the FC weights, $w_{i,k}$ and $w_{j,k}$, are trained to be highly different to compensate the difference introduced by the GAP layer. (c) In the localization phase, the weighted feature with a small activated region, $w_{i,k} \cdot \mathbf{F}_i$, is highly overstated.

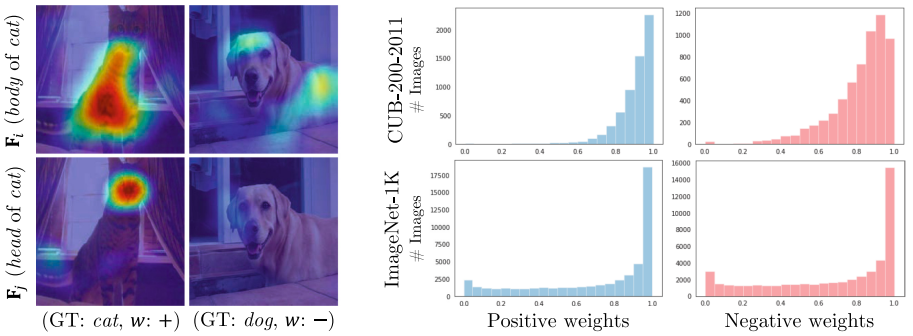
when generating a class activation map ($\mathbf{M}_k$), small activated regions of $\mathbf{F}_i$ are highly overstated due to the large value of $w_{i,k}$, which causes localization to be limited to small regions as localization depends on the maximum value of $\mathbf{M}_k$.

A batch normalization (BN) layer [16] can partially alleviate this issue through normalization as it forces the distributions of the activations to be similar by channel. However, it may also distort the activated area of a channel. For example, when a channel captures a small region like *ears* of a *monkey*, the BN layer expands its originally activated area through normalization, and as a result, localization can be expanded to the background if the channel is activated at the edge of the object. On the other hand, our proposed solution alleviates this problem without distorting the originally activated area.

Solution. To alleviate the problem of the GAP layer, we propose a *thresholded average pooling* (TAP) layer defined as

$$p_c^{\text{tap}} = \frac{\sum_{(h,w)} \mathbb{1}(\mathbf{F}_c(h,w) > \tau_{tap}) \mathbf{F}_c(h,w)}{\sum_{(h,w)} \mathbb{1}(\mathbf{F}_c(h,w) > \tau_{tap})}, \tag{4}$$

where $\tau_{tap} = \theta_{tap} \cdot \max \mathbf{F}_c$ is a threshold value where $\theta_{tap} \in [0, 1)$ is a hyper-parameter. That is, our solution is to replace the GAP layer with the TAP layer (*i.e.* using Eq.(4) instead of Eq.(1)). The TAP layer can be regarded as a generalized pooling layers in between global max pooling (GMP) and global average pooling (GAP). Although GAP has an advantage over GMP for WSOL to expand the activation to broader regions, GMP also has an useful trait that it can precisely focus on the important activations of each channel for pooling.



(a) Activated regions of two features

(b) IoA between the GT boxes and the CAMs

Fig. 3. An example illustrating the characteristics of the feature map channels with negative weights. (a) The activated regions of the i-th and j-th feature map channel for two different images. GT and w denote the ground truth class and sign of the corresponding weight. The less important regions (*e.g. body*) can be activated in multiple classes regardless of the weight sign due to their resemblance between different classes. (b) Intersection over Area (IoA) between the GT boxes and class activation maps generated from only the features corresponding to positive and negative weights, respectively, on both CUB-200–2011 and ImageNet-1K. It indicates how much activated regions and actual object regions are overlapped.

The TAP layer inherits the benefits of both GMP and GAP. By including much broader spatial areas than GMP, it can have the loss propagate to wider feature map activations than GMP which highlights only the most discriminative part [40]. Also, by excluding inactive or less active regions, the pooled channel value p_c^{tap} can better represent the core unique activations of each channel.

2.3 Negative Weight Clamping (NWC)

Problem. When CNNs are trained for classification, a large number of the weights from the FC layer are negative. Since feature map channels corresponding to negative weights only decrease the final logit of the GT class for classification, ideally they should not be activated for the sake of classification, which is not the case in general. According to the underlying assumption of the CAM method from Eq.(2), since only depreciating the values in a class activation map ($\mathbf{M}_k$), they should be activated in *no-object* regions like background. However, in reality, they are activated *inside* the object region, especially *less important* region for classification (**P2** in Fig. 1). As a result, it causes the localization to be limited to the discriminative region of an object further.

To better understand why this happens, we first take an example in Fig. 3(a). For the image of the first column whose GT class is *cat*, i-th and j-th channel of feature maps, $\mathbf{F}_i$ and $\mathbf{F}_j$, have positive weights for the class *cat* and they

successfully capture the *body* and *head* of *cat*, respectively. Contrarily, for the image of the second column whose GT class is *dog*, the two channels have negative weights for the class *dog* and they are supposed to be activated in *no-dog* regions. However, the *body* of *dog* is activated in $\mathbf{F}_i$, because of the resemblance between the *body* of *cat* and *dog*. This phenomenon is very common in practice as *less important* object regions for classification are similar between different classes.

To make sure that this phenomenon commonly happens in WSOL benchmark datasets, we obtain the distributions of Intersection over Area (IoA) between the GT boxes and class activation maps generated from only the features corresponding to positive and negative weights, respectively, on CUB-200–2011 and ImageNet-1k, as described Fig. 3(b). Surprisingly, the distributions of the positive and negative weights are almost identical on both datasets, and it is highly prevalent that GT boxes are overlapped with the activated regions of negatively weighted features.

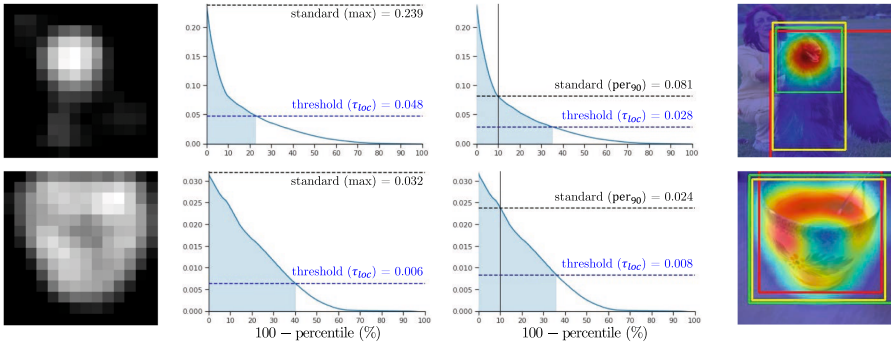
Solution. To mitigate the aforementioned problem, we simply clamp negative weights to zero to generate a class activation map. Hence, Eq.(2) is redefined as

$$\mathbf{M}_k = \sum_{c=1}^{C} \mathbb{1}(w_{c,k} > 0) \cdot w_{c,k} \cdot \mathbf{F}_c. \tag{5}$$

By doing this, we can secure the activations on *less important* object regions that are depreciated by the negative weights. One may think that this negative clamping may increase the chance of background selection if the feature with a negative weight correctly activates background. In our experiments, however, this method does not introduce further background noise, mainly because the features corresponding to positive weights are mostly activated in the object regions and their strengths are sufficiently large. Consequently, the activations in the background are still far below the threshold and can be easily filtered out.

2.4 Percentile as a Standard for Thresholding (PaS)

Problem. As shown in Eq.(3), the thresholding of CAM is simply based on the maximum value of a class activation map. If high activations largely overlap across feature map channels, due to the extremely high maximum value of $\mathbf{M}_k$, the region where activations are greater than the localization threshold is limited to a very small region. The top row in Fig. 4 shows such a case, where the values of a generated class activation map follow Zipf's law in Fig. 4(b) and the values in the discriminative region are exponentially larger than those in non-discriminative regions. On the other hand, when the activations are not solely concentrated in a small region as in the bottom case, the distribution of the activations follows a linearly decreasing pattern as shown in the bottom of (b), and the localization tends to cover the whole region of an object. While the thresholding of CAM works well with the bottom case but fails with the top one, our solution is designed to work robustly with both cases.



(a) Num of channels (activation $> \tau_{0.8}$) (b) CAM values (desc. order) + Maximum as a Standard (c) CAM values (desc. order) + Percentile as a Standard (d) Result with CAM

Fig. 4. An example illustrating the problem of the overlap of high activations. (a) In the problematic case (top), when high activations (activation $> \tau_{0.8}$) are concentrated in the small discriminative region, the localization threshold τ_{loc} in Eq. (3) becomes too high due to the high maximum value of the class activation map, which results in localization being limited to a small region. (b) and (c) are the distributions of class activation map values in the descending order when using the maximum and percentile as thresholding standards, respectively. (d) illustrates the resulting class activation maps. The boxes in red, green and yellow represent the GT, prediction based on the maximum as a standard, and prediction based on the percentile as a standard, respectively. (Color figure online)

Solution. To alleviate the problem of having too large maximum value, a percentile can be employed as a substitute for the maximum value. The percentile is one of the simplest but the most robust metrics that are not sensitive to outliers nor exponential distributions of activations. Hence, Eq. (3) for the localization threshold τ_{loc} is redefined as

$$\tau_{loc} = \theta_{loc} \cdot \text{per}_i(\mathbf{M}'_k), \tag{6}$$

where per_i is an i-th percentile. Although any value in $[0, 1]$ is available for θ_{loc}, for percentile i, due to small object cases where even 80-th percentile of a class activation map is close to zero, we constraint the possible values for i to $[80, 100]$.

Figure 4(c) shows the same distributions as those in (b) except 90-th percentile is used as a standard. When using the maximum as a standard in Fig. 4(b), the thresholded percentiles (*i.e.* the value of x-axis at the threshold) in the top and the bottom are significantly different (*e.g.* 23% in top and 40% in bottom). Contrarily, by using 90-th percentile in Fig. 4(c), they are almost similar around 35%. Figure 4(d) shows the localization results are better when using the proposed percentile (*e.g.* yellow boxes) than using the maximum (*e.g.* green boxes).

3 Related Work

We review two directions of a related research: (i) CAM-based WSOL methods that attempt to resolve the problem of limited localization and (ii) spatial pooling methods for object localization in weakly supervised setting.

3.1 CAM-Based WSOL Methods

The major challenge of WSOL is to capture a whole object region rather than its most discriminative one. Since a CNN backbone is trained for classification, a class activation map is highly activated on the discriminative region not the whole region of an object. Hence, the expansion of the activation beyond the discriminative region in a feature map has been a major research topic for WSOL.

Image Masking. Bazzani *et al.* [1] improve localization by masking out the regions of which classification scores largely drop. Hide-and-Seek (HaS) [31] randomly hide patches in an image to make a classifier seek other regions. Choe *et al.* [5] improve HaS using GoogleNet Resize (GR) augmentation. Wei *et al.* [34] propose an adversarial erasing that progressively erases the most discriminative parts using multiple classifiers and combines them for final localization.

Feature Masking. Instead of masking in the image level, Kim *et al.* [17] and SeeNet [13] propose two-phase feature-level erasing methods, and ACoL [38] designs an end-to-end parallel adversarial architecture where two classifiers are learned to detect complementary regions via adversarially erasing feature maps. ADL [6] is an attention-based dropout layer that randomly generates masks on feature maps to expand the activations to less discriminative regions of an object.

Other Methods. SPG [39] uses highly discriminative regions from the latter layers as a mask supervision to the earlier layers. DANet [36] trains intermediate layers using the classes obtained from knowledge graphs of class hierarchy expecting the expansion of activations to the common object regions.

Unlike the previous methods, we aim at fully and correctly leveraging the information that already exists in the CAM pipeline. Thus, instead of endeavoring to devise a new architecture as done previously, we focus on discovering problematic steps of CAM and proposing simple but robust solutions.

3.2 Spatial Pooling Methods

Due to the absence of bounding box annotations, WSOL relies on the activations of feature maps to localize an object. Several approaches have been proposed to deal with how to extract the information from feature maps.

Representation Pooling. Oquab *et al.* [23] and Pinheiro *et al.* [24] respectively propose to use the max pooling and log-sum-exp layers as pooling methods for CAM. Although the max pooling accurately tells the most discriminative region of an object, its localization is highly limited to the small regions. Zhou *et al.*

[40] use a GAP layer proposed in Lin *et al.* [19] as a replacement for max pooling since the loss for average pooling benefits all the activated regions.

Gradient-Based Pooling. To utilize the GAP layer, the last layer of CNNs has to be converted to a FC layer following the GAP layer, which does not align with the structure of many of well-known classification CNNs [12,14,30,32]. Because of this limitation, GradCAM [29] and GradCAM+ [3] propose gradient-based methods to obtain a class activation map. Although gradient-based methods are applicable to any classification model with no modification of architecture, they are overwhelming in terms of the computation and memory cost without much improvement on the performance. Thus, using the GAP layer is still a de facto standard approach to WSOL, including recent works such as [6,13,36,39].

Score Pooling. Instead of pooling information only from the maximum scoring regions, WELDON [9] and WILDCAT [8] include the minimum scoring regions to regularize the class score. The score pooling (SP) proposed in WILDCAT is the closest idea to our TAP layer but they are fundamentally different in that SP is applied to a fixed number of activations on the class map to consider both positive and negative regions for classification, whereas TAP adaptively includes activations for pooling for every channel of the feature map to correctly estimate a weight for each channel in localization.

4 Experiments

We evaluate the proposed approach on two standard benchmarks for WSOL: CUB-200–2011 [33] and ImageNet-1K [28], and one recently proposed benchmark: OpenImages30K [2,4]. Our approach consistently improves the performance with various CNN backbones and WSOL methods; especially, we achieve the new state-of-the-art performance on all three datasets.

4.1 Experiment Setting

Datasets. CUB-200–2011 [33] consists of 200 bird species. The numbers of images in training and test sets are 6,033 and 5,755, respectively. ImageNet-1K [28] consists of 1,000 different categories; the numbers of images in training and validation sets are about 1.3 million and 50,000, respectively. We use bounding box annotations of the datasets only for the purpose of evaluation. OpenImages30K [2,4] consists of 29,819, 2,500 and 5,000 images for training, validation, and test sets, respectively, with binary mask annotations.

Implementation. To validate the robustness of our methods, we employ four different CNN backbones: VGG16 [30], ResNet50-SE [12,15], MobileNetV1 [14] and GoogleNet [32]. For VGG16, we replace the last pooling layer and two following FC layers with a GAP layer as done in [40]. We add SE blocks [15] on top of ResNet50 to build ResNet50-SE for CUB-200–2011 and ImageNet-1K following ADL [6], and leave ResNet50 as it is for OpenImages30K following Choe *et al.* [4]. For GoogleNet, we replace the last inception block with two

CONV layers based on SPG [39]. For the threshold τ_{tap} of TAP layer in Eq. (4), we set $\theta_{tap} = 0.1$ for VGG16 and MobileNetV1 and $\theta_{tap} = 0.0$ for ResNet50-SE and GoogleNet. Also, localization hyperparameters, i and θ_{loc}, in Eq. (6) are set to $\theta_{loc} = 0.35, i = 90$, which are fixed regardless of the backbones or datasets. The detailed hyperparameter tuning is described in the appendix.

Evaluation metrics. We report the performance of models using *Top-1 Cls*, *GT Loc*, and *Top-1 Loc* on CUB-200–2011 and ImageNet-1K, and PxAP on OpenImages30K. *Top-1 Cls* is the top-1 accuracy of classification, and *GT Loc* measures the localization accuracy with known ground truth classes. For *Top-1 Loc*, the prediction is counted as correct if the predictions on both classification and localization (*i.e.* IoU ≥ 0.5) are correct. Pixel Average Precision (PxAP) [4] is the area under a pixel precision and recall curve. As precision and recall are computed for all thresholds, PxAP is independent to the choice of a threshold.

4.2 Quantitative Results

Comparison with the State-of-the-Arts. As the proposed solutions are applicable to any CAM-based WSOL algorithms, we validate their compatibility with two recent state-of-the-art models. We select HaS [31] and ADL [6] as they are two of the best performing models for WSOL.

Table 1 provides the comparison of the proposed methods on HaS and ADL with various backbone structures and the state-of-the-art models: ACoL [38], SPG [39] and DANet [36]. We validate the proposed approaches further improve both HaS and ADL on CUB-200–2011 and ImageNet-1K. Especially, ADL with our approaches significantly outperforms all the state-of-the-art algorithms on CUB-200–2011, and obtain the comparable results on ImageNet-1K. To the best of our knowledge, Baseline + Ours with VGG16 and ResNet50-SE that are shown in Table 2 are the new state-of-the-art performance on CUB-200–2011 and ImageNet-1K, respectively.

Results with Different Backbones. To validate the robustness of our solutions, we experiment our approach with different backbones. Table 2 summarizes the results on CUB-200–2011 and ImageNet-1K. In terms of *Top-1 Loc* regarded as the most important metric for WSOL, our approach improves the *baseline*, which refers to Vanilla CAM [40], with significant margins (CUB: 14.18, ImageNet: 2.84 on average). The results are compatible or even better than the state-of-the-art methods on both datasets as shown in Table 1.

Results with Different Components. We further investigate the effectiveness of each of the proposed solutions using VGG16 on CUB-200–2011 and ImageNet-1K. Due to space constraint, we defer the results of the other backbones to the appendix. In Table 3, three leftmost columns denote whether each of our solutions is applied to the baseline, Vanilla CAM with VGG-16.

The TAP layer improves the performance of both classification (CUB: 69.95 → 74.91, ImageNet: 65.39 → 67.22) and localization (CUB: 37.05 → 48.53,

Table 1. Comparison of the proposed methods applied to ADL and HaS-32 with other state-of-the-art algorithms. The methods with * indicate the scores are referred from the original paper. – indicates no accuracy reported in the paper.

Backbone	Method	CUB-200–2011			ImageNet-1K		
		Top-1 Cls	GT Loc	Top-1 Loc	Top-1 Cls	GT Loc	Top-1 Loc
VGG16	ACoL*	71.90	–	45.92	67.50	–	**45.83**
	SPG*	**75.50**	–	48.93	–	–	–
	DANet*	75.40	–	52.52	–	–	–
	HaS-32	66.10	71.57	49.46	62.28	61.23	41.64
	HaS-32 + Ours	70.12	**78.58**	57.37	66.21	**61.48**	43.91
	ADL	69.05	73.96	53.40	68.03	59.24	42.96
	ADL + Ours	75.01	76.30	**58.96**	**68.67**	60.73	44.62
ResNet50	HaS-32	71.28	72.56	53.97	74.37	62.95	48.27
	HaS-32 + Ours	72.51	75.34	57.42	73.75	**63.84**	49.40
	ADL	**76.53**	71.99	57.40	75.06	61.04	48.23
	ADL + Ours	75.03	**77.58**	**59.53**	**75.82**	62.20	**49.42**
MobileNetV1	HaS-32	65.98	67.31	46.70	65.45	60.12	42.73
	Has-32 + Ours	71.16	75.04	55.56	65.60	**62.22**	44.31
	ADL	71.90	62.55	47.69	67.02	59.21	42.89
	ADL + Ours	**73.51**	**78.60**	**59.41**	**67.15**	61.69	**44.78**
GoogleNet	ACoL*	–	–	–	–	–	46.72
	SPG*	–	–	46.64	–	–	48.60
	DANet*	71.20	–	49.45	72.50	–	47.53
	Has-32	**75.35**	61.08	47.36	68.92	60.55	44.64
	Has-32 + Ours	74.25	67.03	50.64	67.86	62.36	45.36
	ADL	73.37	66.81	51.29	**74.38**	60.84	47.72
	ADL + Ours	73.65	**69.95**	**53.04**	74.25	**64.44**	**50.56**

Table 2. Performance of the proposed methods applied to Vanilla CAM (Baseline) with various backbone structures.

Backbone	Method	CUB-200–2011			ImageNet-1K		
		Top-1 Cls	GT Loc	Top-1 Loc	Top-1 Cls	GT Loc	Top-1 Loc
VGG16	Baseline	69.95	53.68	37.05	64.56	59.81	41.62
	+ Ours	**74.91**	**80.72**	**61.30**	**67.28**	**61.69**	**44.69**
ResNet50-SE	Baseline	**78.62**	56.49	43.29	77.22	58.21	46.64
	+ Ours	77.42	**74.51**	**58.39**	**77.25**	**64.40**	**51.96**
MobileNetV1	Baseline	72.09	58.92	44.46	67.34	59.45	43.29
	+ Ours	**75.82**	**74.28**	**57.63**	**68.07**	**61.85**	**45.55**
GoogleNet	Baseline	74.35	61.67	46.86	70.50	62.32	46.98
	+ Ours	**75.04**	**65.10**	**51.05**	**71.09**	**62.76**	**47.70**

ImageNet: 41.91 → 45.29). The weight clamping method as well as 90-th percentile standard also constantly improve the performance of localization regardless of datasets (CUB: 37.05 → 44.15, 48.45, ImageNet: 41.91 → 42.39, 44.04). With using all the solutions, the localization accuracies are maximized on both datasets.

Table 3. Performance variations of Vanilla CAM [40] with VGG16 according to different usage of our solutions. TAP, NWC and PaS refer to thresholded average pooling, negative weight clamping and percentile as a standard for thresholding.

Method	TAP	NWC	PaS	CUB-200–2011			ImageNet-1K		
				Top-1 Cls	GT Loc	Top-1 Loc	Top-1 Cls	GT Loc	Top-1 Loc
Baseline				69.95	53.68	37.05	65.39	59.65	41.91
+ Ours	✓			**74.91**	64.10	48.53	**67.22**	62.38	45.29
		✓		69.95	64.30	44.15	65.39	60.44	42.39
			✓	69.95	65.90	48.45	65.39	62.08	44.04
	✓	✓		**74.91**	73.58	54.41	**67.22**	62.48	45.24
	✓		✓	**74.91**	72.87	56.64	**67.22**	61.85	45.01
		✓	✓	69.95	76.42	54.30	65.39	**62.77**	44.40
	✓	✓	✓	**74.91**	**80.72**	**61.30**	**67.22**	62.68	**45.40**

Results on OpenImages30K. A drawback of *GT Loc* and *Top-1 Cls* is that they are sensitive to a localization threshold θ_{loc}. To validate that the robustness of our methods is not originated from a choice of the localization threshold, we compare the performance of our proposed solution applied to Vanilla CAM [40] and ADL [6] to other state-of-the-art algorithms on OpenImages30K using PxAP [4], which is independent to a threshold. Table 4 shows that CAM + Ours outperform all the other methods of which performance is cited from [4]. Also, our proposed methods significantly improve ADL performance.

Table 4. Performance on OpenImages30K.

Method	VGG16	GoogleNet	ResNet50
HaS	56.9	58.5	58.2
ACoL	54.7	63.0	57.8
SPG	55.9	62.4	57.7
CutMix [37]	58.2	61.7	58.7
CAM	58.1	61.4	58.0
CAM + Ours	**59.6**	**63.3**	**60.9**
Δ	(+1.5)	(+1.9)	(+2.9)
ADL	58.3	62.1	54.3
ADL + Ours	59.3	**63.3**	55.7
Δ	(+1.0)	(+1.2)	(+1.4)

Discussion on Datasets. Interestingly, the improvement of localization performance by our methods is much higher on CUB-200–2011 than on ImageNet-1K and OpenImages30K. We conjecture the reasons are two-fold. First, our method works better on harder classification tasks such as CUB-200–2011 where more sophisticate part distinction is required. In other words, the discriminative regions of CUB-200–2011 are relatively smaller than those of the other datasets as many images in CUB-200–2011 share the common features such as *feathers* and *wings*. Since our proposed method focuses on expanding the localization to less-discriminative regions, it works better on such fine-grained classification problem. Second, negative weight clamping is more effective on single-object images such as CUB-200–2011. Contrary to the assumption of WSOL, ImageNet-1K and OpenImages30K contain multiple objects per image despite its single class labels. With an image of multiple objects, the features with negative weights tend to be activated in object regions of different classes. We elaborate it in the appendix more in detail.

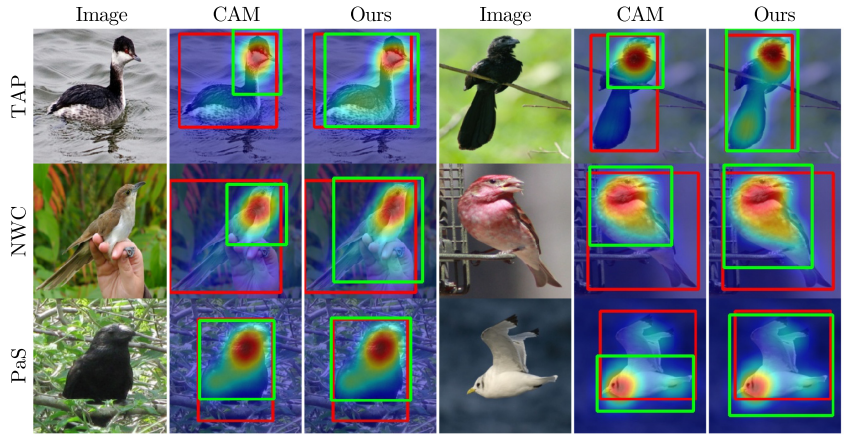


Fig. 5. Comparison of CAM and each of the proposed method. The boxes in red and green represent the ground truths and predictions of localization, respectively. (Color figure online)

4.3 Qualitative Results

Figure 5 provides some localization results that show the effectiveness of each of the proposed methods. All the methods contribute to expand localization from discriminative regions to the whole object regions through (i) TAP: balancing over/underestimated weights of features, (ii) NWC: securing depreciated activations due to negative weights, and (iii) PaS: lowering too high threshold due to the maximum standard. Note that PaS is robustly applicable whether the overlap of high activations is too severe (right) or not (left).

Figure 6 further provides localization results for the proposed methods on Vanilla CAM and ADL with VGG16 and ResNet50-SE. In general, the proposed methods help each model to utilize more activations in object regions, which results in the expansion of bounding boxes compared to the ones from CAM and ADL. We provide additional qualitative results on the other combination of backbones and modules in the appendix.

5 Conclusion

Class activation mapping (CAM), the foundation of WSOL algorithms, has three major problems which cause localization to be limited to small discriminative regions. Instead of devising a new architecture as done in most previous studies, we proposed three simple but robust methods to properly and efficiently utilize the information that already resides in feature maps. We validated the proposed method largely mitigated the problems, and as a result, achieved the new state-of-the-art performance on CUB-200–2011, ImageNet-1K, and OpenImages30K.

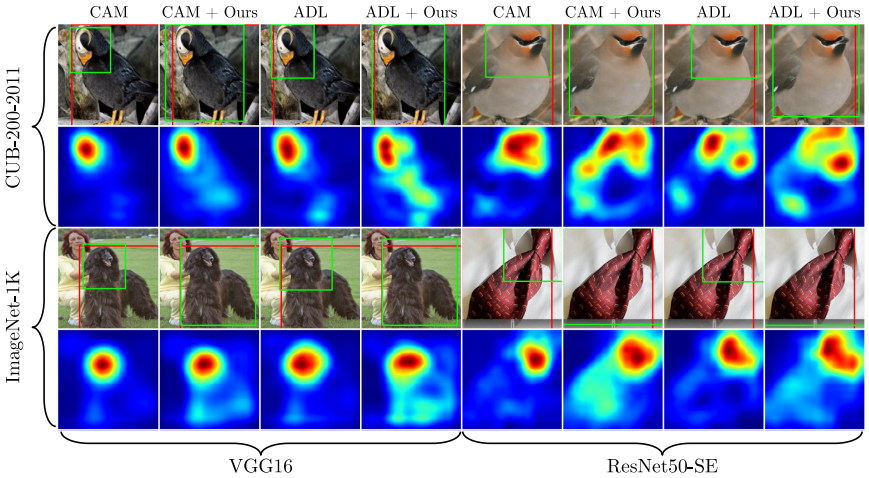


Fig. 6. Localization results of various models with and without our approach applied on CUB-200–2011 and ImageNet-1K datasets. The boxes in red and green represent the ground truths and predictions of localization, respectively. (Color figure online)

As a future work, we will investigate a more integrated algorithm to handle the aforementioned problems of the CAM method. Furthermore, instead of only using the information for a class as done in the current CAM method, using other external information such as weights of other classes may help to better localize an object by utilizing the relationship between different classes.

Acknowledgements. We appreciate Hyunwoo Kim and Jinhwan Seo for their valuable comments. This work was supported by AIR Lab (AI Research Lab) in Hyundai Motor Company through HMC-SNU AI Consortium Fund, and the ICT R&D program of MSIT/IITP (No. 2019-0-01309, Development of AI technology for guidance of a mobile robot to its goal with uncertain maps in indoor/outdoor environments and No.2019-0-01082, SW StarLab).

References

1. Bazzani, L., Bergamo, A., Anguelov, D., Torresani, L.: Self-Taught Object Localization With Deep Networks. In: WACV (2016)
2. Benenson, R., Popov, S., Ferrari, V.: Large-scale Interactive Object Segmentation with Human Annotators. In: CVPR (2019)
3. Chattopadhay, A., Sarkar, A., Howlader, P., Balasubramanian, V.N.: Grad-Cam++: Improved Visual Explanations for Deep Convolutional Networks. In: WACV (2018)
4. Choe, J., Oh, S.J., Lee, S., Chun, S., Akata, Z., Shim, H.: Evaluating Weakly Supervised Object Localization Methods Right. In: CVPR (2020)
5. Choe, J., Park, J.H., Shim, H.: Improved Techniques for Weakly-Supervised Object Localization. Arxiv:1802.07888 (2018)

6. Choe, J., Shim, H.: Attention-Based Dropout Layer for Weakly Supervised Object Localization. In: CVPR (2019)
7. Dai, J., Li, Y., He, K., Sun, J.: R-FCN: Object Detection via Region-Based Fully Convolutional Networks. In: NeurIPS (2016)
8. Durand, T., Mordan, T., Thome, N., Cord, M.: WILDCAT: Weakly Supervised Learning of Deep ConvNets for Image Classification, Pointwise Localization and Segmentation. In: CVPR (2017)
9. Durand, T., Thome, N., Cord, M.: WELDON: Weakly Supervised Learning of Deep Convolutional Neural Networks. In: CVPR (2016)
10. Everingham, M., Eslami, S.M.A., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The Pascal Visual Object Classes Challenge: A Retrospective. IJCV (2015)
11. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: ICCV (2017)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: CVPR (2016)
13. Hou, Q., Jiang, P., Wei, Y., Cheng, M.M.: Self-Erasing Network for Integral Object Attention. In: NeurIPS (2018)
14. Howard, A.G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., Adam, H.: MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. Arxiv:1704.04861 (2017)
15. Hu, J., Shen, L., Sun, G.: Squeeze-and-Excitation Networks. In: CVPR (2018)
16. Ioffe, S., Szegedy, C.: Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In: ICML (2015)
17. Kim, D., Cho, D., Yoo, D., Kweon, I.: Two-Phase Learning for Weakly Supervised Object Localization. In: ICCV (2017)
18. Li, K., Wu, Z., Peng, K.C., Ernst, J., Fu, Y.: Tell Me Where to Look: Guided Attention Inference Network. In: CVPR (2018)
19. Lin, M., Chen, Q., Yan, S.: Network in Network. In: ICLR (2014)
20. Lin, T.Y., Goyal, P., Girshick, R., He, K., Dollár, P.: Focal Loss for Dense Object Detection. In: ICCV (2017)
21. 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 (2014)
22. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: SSD: Single Shot Multibox Detector. In: ECCV (2016)
23. Oquab, M., Bottou, L., Laptev, I., Sivic, J.: Is Object Localization for Free? - Weakly-Supervised Learning With Convolutional Neural Networks. In: CVPR (2015)
24. Pinheiro, P.O., Collobert, R.: From Image-Level to Pixel-Level Labeling With Convolutional Networks. In: CVPR (2015)
25. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You Only Look Once: Unified, Real-Time Object Detection. In: CVPR (2016)
26. Redmon, J., Farhadi, A.: YOLO9000: Better, Faster. CVPR, Stronger. In (2017)
27. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In: NeurIPS (2015)
28. Russakovsky, O., Deng, J., SU, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: Imagenet Large Scale Visual Recognition Challenge. IJCV (2015)
29. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-Cam: Visual Explanations From Deep Networks via Gradient-Based Localization. In: ICCV (2017)

30. Simonyan, K., Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. Arxiv:1409.1556 (2014)
31. Singh, K.K., Lee, Y.J.: Hide-And-Seek: Forcing a Network to Be Meticulous for Weakly-Supervised Object and Action Localization. In: ICCV (2017)
32. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going Deeper with Convolutions. In: CVPR (2015)
33. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD Birds-200-2011 Dataset. Tech. Rep. Cns-Tr-2011-001, California Institute of Technology (2011)
34. 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 (2017)
35. Wei, Y., Shen, Z., Cheng, B., Shi, H., Xiong, J., Feng, J., Huang, T.: TS2C: Tight Box Mining with Surrounding Segmentation Context for Weakly Supervised Object Detection. In: ECCV (2018)
36. Xue, H., Wan, F., Jiao, J., Ji, X., Qixiang, Y.: DANet: Divergent Activation for Weakly supervised Object Localization. In: ICCV (2019)
37. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: CutMix: Regularization Strategy to Train Strong Classifiers with Localizable Features. In: ICCV (2019)
38. Zhang, X., Wei, Y., Feng, J., Yang, Y., Huang, T.S.: Adversarial Complementary Learning for Weakly Supervised Object Localization. In: CVPR (2018)
39. Zhang, X., Wei, Y., Kang, G., Yang, Y., Huang, T.: Self-Produced Guidance for Weakly-Supervised Object Localization. In: ECCV (2018)
40. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning Deep Features for Discriminative Localization. In: CVPR (2016)