Bagging Regional Classification Activation Maps for Weakly Supervised Object Localization

Lei Zhu[1,2,3], Qian Chen[1,2,3], Lujia Jin[1,2,3], Yunfei You[1,2,3],
and Yanye Lu[1,2,3](✉)

[1] Institute of Medical University, Peking University, Beijing, China
zhulei@stu.pku.edu.cn, yanye.lu@pku.edu.cn
[2] Department of Biomedical Engineering, Peking University, Beijing, China
[3] Institute of Biomedical Engineering, Peking University Shenzhen Graduate School, Beijing, China

Abstract. Classification activation map (CAM), utilizing the classification structure to generate pixel-wise localization maps, is a crucial mechanism for weakly supervised object localization (WSOL). However, CAM directly uses the classifier trained on image-level features to locate objects, making it prefers to discern global discriminative factors rather than regional object cues. Thus only the discriminative locations are activated when feeding pixel-level features into this classifier. To solve this issue, this paper elaborates a plug-and-play mechanism called BagCAMs to better project a well-trained classifier for the localization task without refining or re-training the baseline structure. Our BagCAMs adopts a proposed regional localizer generation (RLG) strategy to define a set of regional localizers and then derive them from a well-trained classifier. These regional localizers can be viewed as the base learner that only discerns region-wise object factors for localization tasks, and their results can be effectively weighted by our BagCAMs to form the final localization map. Experiments indicate that adopting our proposed BagCAMs can improve the performance of baseline WSOL methods to a great extent and obtains state-of-the-art performance on three WSOL benchmarks. Code are released at https://github.com/zh460045050/BagCAMs.

Keywords: Weakly supervised learning · Object localization

1 Introduction

Weakly supervised learning, using coarse annotations as supervision during model learning, has attracted extensive attention in recent years, especially for localization relevant vision tasks, such as image segmentation [4,9,13] and object

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/978-3-031-20080-9_11.

S. Avidan et al. (Eds.): ECCV 2022, LNCS 13670, pp. 176–192, 2022.
https://doi.org/10.1007/978-3-031-20080-9_11

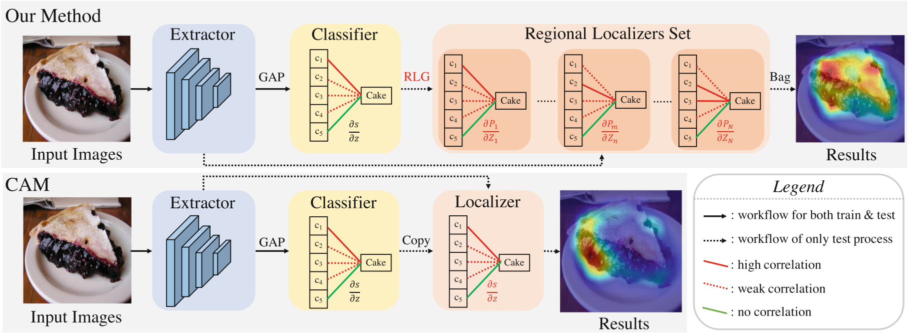

Fig. 1. Comparison between our BagCAMs and CAM. Our BagCAMs (upper part) derives regional localizers from the classifier with the RLG strategy, while CAM (bottom part) only copies the globally-learned classifier to locate objects.

detection [10,27]. Typically, weakly supervised object localization (WSOL) releases the requirements of bounding boxes or even the densely-annotated pixel-level localization masks by only learning the localization model with image-level annotations, *i.e.,* the class of images, which effectively saves human resources for the annotation process. The majority of WSOL methods adopt the mechanism of classification activation map (CAM) [38], utilizing the global average pooling (GAP) to spatially average the pixel-level features into image-level to learn an image classifier with the image-level supervision. Except for generating the classification results, this image classifier also serves as an object localizer that acts on pixel-level features to produce the localization map in the test process.

Though CAM provides an efficient tool for learning a localization model with weak supervision, it directly adopts the classifier as the localizer without considering the difference between them. In detail, the classifier is only learned based on the image-level features, which are spatially aggregated and contain sufficient object features to be discerned. Catching some discriminative factors is enough for the classifier to discern the class of objects. However, the object localizer focuses on discerning the class of all regional positions based on the pixel-level features, where discriminative factors may not be well-aggregated, *i.e.*, insufficient to activate the globally-learned classifier. Thus, the classifier of CAM will only catch the most discriminative parts rather than the whole object locations when directly adopting it to locate objects for pixel-level features.

To solve this issue, a series of methods have been proposed to force the classifier discerning object features more comprehensively, for example, developing augmentation strategies to enrich the global features [17,25,32], aligning the feature distribution between image-level and pixel-level [35,39], adopting multi-classifier to synergistically localize the object [16,30,31,34], or refining the classifier to catch class-agnostic object features [11,37]. Though these strategies show some effect, adopting them requires re-training or revising the baseline structure, enhancing the complexity of the training process. Moreover, they still follow CAM to directly adopt the globally-learned classifier as the localizer, indicating that the gap between classifier and localizer remains unresolved.

Unlike the above methods, our work proposes a plug-and-play approach called BagCAMs, which can better project an image-level trained classifier to comply with the requirement of localization tasks. It can easily replace the classifier projection of CAM and be engaged into existing WSOL methods without re-training the network structure. As visualized in Fig. 1, instead of directly adopting the globally-learned classifier, our method focuses on deriving a set of regional localizers from this well-trained classifier. Those regional localizers can discern object-related factors with respect to each spatial position, acting as the base learners of ensemble learning. With those regional localizers, the final localization results can be obtained by integrating their effect. Experiments show that the proposed BagCAMs significantly improves the performance of the baseline methods and achieves state-of-the-art performance on three WSOL benchmarks.

2 Related Work

Existing WSOLs can be categorized into multi-stage methods [6,11,18–20,33] and one-stage methods [25,31,35–37,39]. The former requires training additional structures upon the classification structure to generate class-agnostic localization results. Our method belongs to the latter, which produces the localization score by projecting the image-classifier back to the pixel-level feature based on CAM, so we just review representative one-stage methods.

To force the classifier to discern some indistinguishable features of objects, Singh *et al.* [17] proposed hide-and-seek (HAS) augmentation that randomly hides the patches of images in the training process. However, hiding patches also causes information loss. Yun *et al.* [32] elaborated a CutMix strategy to solve this issue, which replaces the hidden regions with a patch of another image. Babar [1] adopts the siamese neural network to align location maps of two images that contain complementary patches of the input. Instead of developing augmentation strategies, some one-stage methods also focus on fusing the localization maps of multiple classifiers to comprehensively catch object parts. Typically, Zhang *et al.* [34] suggested learning two classifiers to discern features of objects in a complementary way. Kou *et al.* [16] added an additional classifier to adaptively produce the auxiliary pixel-level mask, which is then utilized by a metric learning loss for supervision. To consider hierarchical cues, Xue *et al.* [30] elaborated the DANet by learning multiple classifiers based on hierarchical features, and Tan *et al.* [25] proposed a pixel-level class selection (PCS) strategy to generalize CAM for hierarchical features. Seunghun *et al.* [31] fused localization maps of different classes with non-local block [29,40] to help catch locations that correlated to multiple classes. Compared with them, our BagCAMs generates multiple localizers for each spatial position by degrading a well-trained classifier with efficient post-processing like CAM, rather than re-training the extractor or additional classifiers, increasing the complexity of the training process.

Beyond the community of WSOL, some methods also improved CAM for the visual explanation of convolutional neural networks, *i.e.*, explaining why CNN makes specific decisions. To engage CAM into CNN without the GAP operator, Selvaraju *et al.* [23] proposed the GradCAM that summarizes the gradient as the importance of neurons to aggregate feature maps. Aditya *et al.* [5] further improved the GradCAM by elaborating a spatial weighing strategy when summarizing the gradient. Recently, Wang *et al.* [28] and Desai [22] explored obtaining neuron importance through forward passing to avoid the gradient calculation. Unlike these methods that aim to better activate the discriminative locations, our method focuses on complying CAM mechanism with the purpose of WSOL, activating object locations as many as possible.

3 Methodology

This section first formally overviews our proposed method that localizes objects with a series of regional localizers. Then, the regional localizer generation (RLG) strategy is illustrated, helping generate these regional localizers for the localization task. Finally, the BagCAMs is proposed to derive these localizers from a well-trained image classifier and produce the final localization map.

3.1 Problem Definition

Given an input image represented by $\boldsymbol{X} \in \mathbb{R}^{3 \times N^I}$, WSOL aims to approximate the localization map $\boldsymbol{Y} \in \mathbb{R}^{K \times N^I}$ by a localization model learned only with the image-level classification mask $\boldsymbol{y} \in \mathbb{R}^{K \times 1}$, where K and N^I are the numbers of classes of interest and pixels, respectively. To learn the localization model with $\boldsymbol{y}$, a backbone network, *i.e.*, ResNet [12] or InceptionV3 [24], is firstly adopted as the feature extractor $e(\cdot)$ to extract pixel-level features $\boldsymbol{Z} = e(\boldsymbol{X}) \in \mathbb{R}^{C \times N}$, where C is the channels of the features with the spatial resolution N. These pixel-level features are fed into the GAP layer to generate the image-level feature $\boldsymbol{z} \in \mathbb{R}^{C \times 1}$. Finally, the classifier $c(\cdot)$ implemented as the fully-connected layer with weight $\mathbf{W} \in \mathbb{R}^{K \times C}$ is acted on the image-level feature to generate the classification result $\boldsymbol{s}$:

$$\boldsymbol{s}_k = c(\boldsymbol{z})_k = (\mathbf{W}\boldsymbol{z})_k = \sum_c \mathbf{W}_{k,c}\boldsymbol{z}_c, \tag{1}$$

where k and c are the index of class and channel, respectively. This classification score $\boldsymbol{s}$ is supervised by the cross-entropy $\mathcal{L}_{ce}(\boldsymbol{y}, \boldsymbol{s})$ to learn the extractor $e(\cdot)$ and the classifier $c(\cdot)$ in the training process.

In the test process, except for generating the classification score $\boldsymbol{s}$, CAM-based methods also utilize the classifier $c(\cdot)$ as a localizer $f(\cdot)$ that acts onto the pixel-level features $\boldsymbol{Z}$ to obtain the localization maps $\boldsymbol{P} \in \mathbb{R}^{K \times N}$:

$$\boldsymbol{P}_{k,i} = f(\boldsymbol{Z})_{k,i} = c(\boldsymbol{Z}_{:,i})_k = \sum_c \mathbf{W}_{k,c}\boldsymbol{Z}_{c,i}. \tag{2}$$

As discussed in Sect. 1, the classifier $c(\cdot)$ is only learned based on the image-level feature $\boldsymbol{z}$, which aggregates the object features on all the positions of $\boldsymbol{Z}$.

This makes the classifier $c(\cdot)$ only discern the most discriminative feature rather than all features that are correlated to the objects. When directly projecting the classifier $c(\cdot)$ as the localizer $f(\cdot)$ that acts on the pixel-level features, some indistinguishable parts, *i.e.*, the body of animals, will not be activated on the output localization maps $\boldsymbol{P}$. Thus, our method adopts the proposed RLG strategy to generate a base localizer set $\mathcal{F} = \{f_1, f_2, ..., f_n\}$ to comprehensively discern the feature of objects. Then, the proposed BagCAMs can implement the base localizer set $\mathcal{F}$ based on the image-classifier $c(\cdot)$ and generate a series of localization maps $\mathcal{P} = \{\boldsymbol{P}_1, \boldsymbol{P}_2, ..., \boldsymbol{P}_n\}$. Finally, these maps are integrated with co-efficient $\{\lambda_1, \lambda_2, ..., \lambda_n\}$ to form the final localization map $\boldsymbol{P}^*$ that determines $\boldsymbol{Y}$:

$$\boldsymbol{P}^* = \sum_n \lambda_n f_n(\boldsymbol{Z}). \tag{3}$$

3.2 Regional Localizers Generation Strategy

The proposed RLG strategy utilizes localization scores and pixel-level feature maps to generate a set of regional localizers, which focuses more on the regional features rather than only discerning the global features as the classifier of the classification task. To better illustrate the proposed RLG strategy, we firstly design the regional localizer inspired by the property of an image classifier. In detail, by differentiating from Eq. 1, the weight $\mathbf{W}$ of the global classifier $c(\cdot)$ can be reformulated [25]:

$$\mathbf{W} = \frac{\partial c(\boldsymbol{z})}{\partial \boldsymbol{z}} = (\frac{\partial \boldsymbol{s}}{\partial \boldsymbol{z}})^\top. \tag{4}$$

Taking it into the Eq. 1, a equivalency of the classifier $c(\cdot)$ can be obtained [25]:

$$c(\boldsymbol{z}) = \mathbf{W}\boldsymbol{z} = (\frac{\partial \boldsymbol{s}}{\partial \boldsymbol{z}})^\top \boldsymbol{z}. \tag{5}$$

Equation 5 indicates that an image classifier $c(\cdot)$ can be represented by the transposition of the partial derivative between the image classification score $\boldsymbol{s}$ and the image feature $\boldsymbol{z}$ [25]. Analogizing this property to the localization task, the regional localizer can be simulated with the following definition.

Definition 1. *Assuming $f(\cdot)$ is a localizer that generates the classification score $\boldsymbol{p}$ on a specific spatial location based on the pixel-level features $\boldsymbol{Z} \in \mathbb{R}^{C\times N}$, i.e., $\boldsymbol{p} = f(\boldsymbol{Z})$, the localizer $f(\cdot)$ can be simulated by a function set $\mathcal{F}$ that contains the partial derivative between this regional classification score $\boldsymbol{p}$ and each regional position of pixel-level features $\boldsymbol{Z}$:*

$$\mathcal{F} = \{f_1, ..., f_n, ..., f_N\} = \{(\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{Z}_{:,1}})^\top, ..., (\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{Z}_{:,n}})^\top, ..., (\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{Z}_{:,N}})^\top\}, \tag{6}$$

where $f_n(\cdot) = (\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{Z}_{:,i}})^\top(\cdot)$ is the regional localizer that catches the relation between regional score $\boldsymbol{p}$ and the pixel-level feature of a specific regional position $\boldsymbol{Z}_{:,i}$.

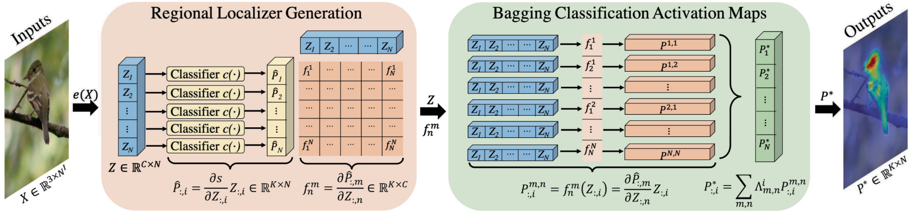
Inputs
$X \in \mathbb{R}^{3\times N^I}$
$e(X)$
Regional Localizer Generation
Z_1
Z_2
Z_N
$Z \in \mathbb{R}^{C\times N}$
Classifier $c(\cdot)$
$\hat{P}_1$
$\hat{P}_2$
$\hat{P}_N$
f_1^1
f_N^1
f_1^N
f_N^N
$\hat{P}_{:,i} = \frac{\partial s}{\partial Z_{:,i}} Z_{:,i} \in \mathbb{R}^{K\times N}$
$f_n^m = \frac{\partial \hat{P}_{:,m}}{\partial Z_{:,n}} \in \mathbb{R}^{K\times C}$
Z
f_n^m
Bagging Classification Activation Maps
f_2^1
f_1^2
$P^{1,1}$
$P^{1,2}$
$P^{2,1}$
$P^{N,N}$
P_1^*
P_2^*
P_N^*
$P^{m,n}_{:,i} = f_n^m(Z_{:,i}) = \frac{\partial \hat{P}_{:,m}}{\partial Z_{:,n}} Z_{:,i}$
$P^*_{:,i} = \sum_{m,n} \Lambda^i_{m,n} P^{m,n}_{:,i}$
P^*
Outputs
$P^* \in \mathbb{R}^{K\times N}$

Fig. 2. Workflow of our method, where the RLG strategy (orange) generates a set of classifiers and the BagCAMs (green) weights their effect to produce localization maps. (Color figure online)

Based on Definition 1, each row vector $\boldsymbol{P}_{:,i}$ of a given localization map $\boldsymbol{P} \in \mathbb{R}^{K\times N}$ can be viewed as a regional classification score $\boldsymbol{p}$ that defines N regional localizers based on the pixel-level feature $\boldsymbol{Z}$. Thus, as indicated in Fig. 2, our RLG strategy (noted by orange) can simulate $N * N$ regional localizers based on the correlation between each vector pair of $\boldsymbol{P}$ and $\boldsymbol{Z}$:

$$f_n^m(\boldsymbol{x}) = (\frac{\partial \boldsymbol{P}_{:,m}}{\partial \boldsymbol{Z}_{:,n}})^\top(\boldsymbol{x}) \longrightarrow f_n^m(\boldsymbol{x})_k = \sum_c \frac{\partial \boldsymbol{P}_{k,m}}{\partial \boldsymbol{Z}_{c,n}} \boldsymbol{x}_c, \tag{7}$$

where $f_n^m(\cdot)_k$ represents the regional localizer of class k and $\boldsymbol{x} \in \mathbb{R}^{C\times 1}$ is a variable that represents a feature vector. With this extension, a localizer set $\mathcal{F}^*$ that contains $N * N$ regional localizer can be defined based on $\boldsymbol{P}$ and $\boldsymbol{Z}$, *i.e.*, $\mathcal{F}^* = \{f_1^1, ..., f_n^m, ..., f_N^N\}$. Compared with the global classifier $(\frac{\partial \boldsymbol{s}}{\partial \boldsymbol{z}})^\top$ used by CAM, our regional localizer set $\mathcal{F}^*$ contains sufficient localizers that catch the regional correlation between scores and features on each position, which helps comprehensively discern features of the objects.

3.3 Bagging Regional Classification Activation Maps

The proposed RLG strategy provides an efficient mechanism to generate a localizer set $\mathcal{F}^*$ based on the localization map $\boldsymbol{P}$. When implementing $\boldsymbol{P}$ as a coarse localization map $\hat{\boldsymbol{P}} \in \mathbb{R}^{K\times N}$, those regional localizers f_n^m can be viewed as the base learners that can be integrated as a strong learner to locate objects. For this purpose, our BagCAMs is proposed as shown in Fig. 2 (noted by green), which generates the base localizers based on a coarse localization map $\hat{\boldsymbol{P}}$ and then weights their localization results as the final localization score:

$$\boldsymbol{P}^*_{k,i} = \sum_m \sum_n \boldsymbol{\Lambda}^i_{m,n} f_n^m(\boldsymbol{Z}_{:,i})_k = \sum_m \sum_n \boldsymbol{\Lambda}^i_{m,n} \sum_c \frac{\partial \hat{\boldsymbol{P}}_{k,m}}{\partial \boldsymbol{Z}_{c,n}} \boldsymbol{Z}_{c,i}, \tag{8}$$

where $\boldsymbol{P}^*$ is the localization map of our proposed BagCAMs whose element $\boldsymbol{P}^*_{k,i}$ represents the score on class k at position i. $\boldsymbol{\Lambda}^i$ is a matrix, and its element $\boldsymbol{\Lambda}^i_{m,n}$ means the co-efficient of regional localizer f_n^m at position i. In detail, PCS

Table 1. Summary of degrading the proposed BagCAMs into other methods

	Initial score $\hat{\boldsymbol{P}}_{k,m}$	Co-efficient matrix $\boldsymbol{\Lambda}^i$	Localization score $\boldsymbol{P}^*_{k,i}$
CAM	$\boldsymbol{s}_k$	$\boldsymbol{\Lambda}^i = \frac{1}{N}\mathbf{I}$	$\sum_c \frac{\partial \boldsymbol{s}_k}{\partial \boldsymbol{z}_c} \boldsymbol{Z}_{c,i}$
GradCAM	$\boldsymbol{s}_k$	$\boldsymbol{\Lambda}^i = \frac{1}{N}\mathbf{I}$	$\frac{1}{N} \sum_{n,c} \frac{\partial \boldsymbol{s}_k}{\partial \boldsymbol{Z}_{c,n}} \boldsymbol{Z}_{c,i}$
GradCAM++	$\boldsymbol{s}_k$	$\boldsymbol{\Lambda}^i = diag(\alpha)$	$\sum_{n,c} \alpha_m \frac{\partial \boldsymbol{s}_k}{\partial \boldsymbol{Z}_{c,n}} \boldsymbol{Z}_{c,i}$
PCS	$\boldsymbol{s}_k$	$\boldsymbol{\Lambda}^i_{m,n} = \begin{cases} 1, & i = n \\ 0, & i \neq n \end{cases}$	$\sum_c \frac{\partial \boldsymbol{s}_k}{\partial \boldsymbol{Z}_{c,i}} \boldsymbol{Z}_{c,i}$
Ours	$\sum_c \frac{\partial \boldsymbol{s}_k}{\partial \boldsymbol{Z}_{c,m}} \boldsymbol{Z}_{c,m}$	$\boldsymbol{\Lambda}^i_{m,n} = \begin{cases} 1, & i = n \\ 0, & i \neq n \end{cases}$	$\sum_{m,c_2} \frac{\partial(\sum_{c_1} \frac{\partial \boldsymbol{s}_k}{\partial \boldsymbol{Z}_{c_1,m}} \boldsymbol{Z}_{c_1,m})}{\partial \boldsymbol{Z}_{c_2,i}} \boldsymbol{Z}_{c_2,i}$

strategy [25] is adopted to initialize the coarse localization map $\hat{\boldsymbol{P}}_{k,m}$ to pursue the convenience of calculation and performance on intermediate feature maps:

$$\hat{\boldsymbol{P}}_{k,m} = \sum_c \frac{\partial \boldsymbol{s}_k}{\partial \boldsymbol{Z}_{c,m}} \boldsymbol{Z}_{c,m}. \tag{9}$$

With this initialization coarse localization map $\hat{\boldsymbol{P}}_{k,m}$ and defining $\bar{\boldsymbol{s}} = log(\boldsymbol{s})$, the formulation of our base localizer generated by our RLG derivatives into the following, whose proof are given in Appendix B:

$$f_n^m(\boldsymbol{x})_k = \sum_{c_1} \boldsymbol{s}_k (1 + \frac{\partial \bar{\boldsymbol{s}}_k}{\partial \boldsymbol{Z}_{c_1,m}} \boldsymbol{Z}_{c_1,m}) \sum_{c_2} (\frac{\partial \bar{\boldsymbol{s}}_k}{\partial \boldsymbol{Z}_{c_2,n}} \boldsymbol{x}_{c_2}). \tag{10}$$

As for the weight matrix $\boldsymbol{\Lambda}^i$, the grouping strategy of PCS [25] is also adopted for consistency, assuming $(\frac{\partial \boldsymbol{p}}{\partial \boldsymbol{Z}_{:,i}})^\top$ is the localizer specifically for the position i:

$$\boldsymbol{\Lambda}^i_{m,n} = \begin{cases} 1, & i = n \\ 0, & i \neq n \end{cases}. \tag{11}$$

This setting assigns the $N * N$ regional localizers into N groups, each applied specifically to position i. Note that $\boldsymbol{\Lambda}^i$ can also be implemented with other mechanisms, for example, spatial average [23] or spatial attention [5], but we find the grouping strategy performs the best due to lesser noise. Finally, taking Eq. 10 and Eq. 11 into Eq. 8, an executable formulation of BagCAMs is obtained:

$$\boldsymbol{P}^*_{k,i} = \sum_m \sum_{c_1} \boldsymbol{s}_k (1 + \frac{\partial \bar{\boldsymbol{s}}_k}{\partial \boldsymbol{Z}_{c_1,m}} \boldsymbol{Z}_{c_1,m}) (\sum_{c_2} \frac{\partial \bar{\boldsymbol{s}}_k}{\partial \boldsymbol{Z}_{c_2,i}} \boldsymbol{Z}_{c_2,i}). \tag{12}$$

As indicated in Eq. 12, the computation of our BagCAMs only relies on the gradients $\frac{\partial \bar{\boldsymbol{s}}}{\partial \boldsymbol{Z}}$, which can be calculated by backward propagating gradients on the logarithm of the classification score $\boldsymbol{s}$. Thus, our BagCAMs can be projected onto the intermediate layer of CNN and retain similar computation complexity as gradient-based CAM mechanisms [5,23,25]. Moreover, Table 1 also shows PCS [25] and other CAM mechanisms [5,23,38] can also be generalized by our

BagCAMs with the assumption that the initial localization result of each position i are all equal to $\boldsymbol{s}_k$, *i.e.*, $\forall \hat{\boldsymbol{P}}_{k,m} = \boldsymbol{s}_k$. However, this assumption is obviously invalid for the localization task because the background locations of the image should not have the same score as the object locations. Compared with them, our BagCAMs generates a specific initial score $\hat{\boldsymbol{P}}_{k,m} \in \mathbb{R}^{K \times N}$ for each position to obtain more valid base localizers to generate high-quality localization maps, rather than defining the localizer only based on the global score $\boldsymbol{s} \in \mathbb{R}^{K \times 1}$. This makes our BagCAMs perform much better than these mechanisms when engaged into WSOL.

The proposed BagCAMs can easily replace CAM step of WSOL methods to generate the localization maps. Algorithm 1 and Fig. 2 show the workflow of localizing objects for an input image $\boldsymbol{X}$ based on a trained WSOL model that contains a feature extractor $e(\cdot)$ and a classifier $c(\cdot)$. Specifically, the input image $\boldsymbol{X}$ is firstly fed into the feature extractor $e(\cdot)$ to generate the pixel-level feature $\boldsymbol{Z} = e(\boldsymbol{X})$. Then, $\boldsymbol{Z}$ is aggregated into image-level feature $\boldsymbol{z}$, which is fed into the classifier to produce the classification score $\boldsymbol{s}$ determining the object class $k = \arg\max(\boldsymbol{s})$. Next, backward propagation is adopted for $\bar{\boldsymbol{s}}_k$ to calculate $\frac{\partial \bar{\boldsymbol{s}}_k}{\partial \boldsymbol{Z}}$ that is crucial for defining the base localizer. Finally, the localization map $\boldsymbol{Y}$ is obtained by weighing the localization scores of base localizers as in Eq. 12.

Algorithm 1. Workflow of BagCAMs for a Given WSOL Model

```
Input: Input image $\boldsymbol{X}$, Classifier $c(\cdot)$, Extractor $e(\cdot)$.
Calculating pixel-level feature $\boldsymbol{Z}$ of input image $\boldsymbol{X}$ with extractor $e(\boldsymbol{X})$.
Obtaining image-level feature $\boldsymbol{z}$ with GAP or other aggregation mechanisms.
Generating image classification score $\boldsymbol{s}$ with classifier $c(\boldsymbol{z})$.
Calculating classification results $k = \arg\max(\boldsymbol{s})$.
Backward propagating $\bar{\boldsymbol{s}} = \log(\boldsymbol{s})_k$ to obtain the gradient $\frac{\partial \bar{\boldsymbol{s}}_k}{\partial \boldsymbol{Z}}$.
Generating BagCAMs localization map $\boldsymbol{P}^*_{k,:}$ by Eq. 12 and upsampling it as $\boldsymbol{Y}$.
Output: Localization Score $\boldsymbol{Y}$, Classification Score $\boldsymbol{s}$.
```

4 Experiments

This section first introduces the setting of experiments. Then, results of our BagCAMs are shown to compare with SOTA methods on three datasets. Finally, we investigate different settings of our BagCAMs to further reflect its validity.

4.1 Settings

The proposed BagCAMs can be engaged into a well-trained WSOL model by simply replacing CAM in the test process. Thus, we reproduced five WSOL methods as the baseline methods to train them with their optimal settings, including CAM [38], HAS [17], CutMix [32], ADL [6], and DAOL [39]. In detail, the ResNet-50, removing the down-sample layer of Res_4, was used as the feature extractor. When using InceptionV3 as the extractor, we follow existing works [20, 25, 34, 35] that add two additional layers at the end of the original structure. The classifier is implemented as a fully-connected layer, whose outputs are supervised by the cross entropy based on the image-level annotation in the training process. Except for the method-specific strategy [6, 17, 32], the random resize with size 256×256 and random horizontal flip crop with size 224×224 were adopted as the augmentation. SGD with weight decay 10^{-4} and momentum 0.9 was set as the optimizer. Note that the learning rate and the method-specific hyper-parameters for all datasets were adopted as the released optimal settings [7, 39]. In the test process, our BagCAMs replaced the CAM step of these methods to project the learned classifier as the localizer based on features outputted by Res_3 of the ResNet (Mix_{6e} for the InceptionV3). All experiments were implemented with Pytorch toolbox [21] on an Intel Core i9 CPU and an NVIDIA RTX 3090 GPU.

Three standard benchmarks were utilized to evaluate our methods:

- **CUB-200 dataset** [26] contains 11,788 images that are fine-grained annotated for 200 classes of birds. We follow the official training/test split to use 5,944 images as the training set that only utilizes image-level annotation to supervise WSOL methods. Other 5,794 images, given additional bounding boxes and pixel-level masks, serve as the test set to evaluate the performance.
- **ILSVRC dataset** [8] contains 1.3 million images that include 1000 classes of objects. Among them, 50,000 images, whose bounding boxes annotation is provided, are adopted as the test set to report the localization performance.
- **OpenImages dataset** [3, 7] contains 37,319 images of 100 classes, where 29,819 images serve as the training set. Following the split released by Junsuk [7], the rest 7,500 images, annotated by pixel-level localization mask, are divided into the validation set (2,500 images) and test set (5,000 images).

Note that our BagCAMs does not contain any hyper-parameters, thus only the test images of these dataset are utilized for comparison. The Top-1 localization accuracy (T-Loc) [17], ground-truth known localization accuracy (G-Loc) [17], and the recently proposed MaxBoxAccV2 [7] (B-Loc) were adopted to evaluate the performance based on bounding box annotations. As for pixel-level localization masks, the peak intersection over union (pIOU) [37] and the pixel average precision (PxAP) [7] were calculated as the metrics.

4.2 Comparison with State-of-the-Arts

Table 2 illustrates the results of SOTA methods and our BagCAMs on the three standard WSOL benchmarks. It shows that adopting our proposed BagCAMs improves the performance of baseline methods to a great extent, especially on the CUB-200 dataset. This is because the CUB-200 dataset is a fine-graining dataset that only contains birds, making the classifier more likely to catch discriminative parts rather than the common parts of birds. As discussed in Sect. 3, this situation basically causes unsatisfactory performance when directly using the

Table 2. Comparison with SOTA methods with ResNet50 (**border** means the best).

	CUB-200					ILSVRC			OpenImages	
	T-Loc	G-Loc	B-Loc	pIoU	PxAP	T-Loc	G-Loc	B-Loc	pIoU	PxAP
DGL [25]	60.82	76.65	–	–	–	53.41	66.52	–	–	–
CAAM [1]	64.70	77.35	–	–	–	52.36	67.89	–	–	–
DANet [30]	61.10	–	–	–	–	–	–	–	–	–
ICLCA [14]	56.10	72.79	63.20	–	–	48.40	67.62	65.15	–	–
PAS [2]	59.53	77.58	66.38	–	–	49.42	62.20	64.72	–	60.90
IVR [15]	–	–	71.23	–	–	–	–	65.57	–	58.90
PSOL [33]	70.68	–	–	–	–	53.98	65.54	–	–	–
SEM [37]	–	–	–	–	–	53.84	67.00	–	–	–
FAM [19]	**73.74**	85.73	–	–	–	**54.46**	64.56	–	–	–
CAM [38]	55.31	66.06	59.21	46.70	65.94	49.93	67.30	62.69	43.13	57.88
+**Ours**	70.89	87.44	76.22	64.40	84.38	52.14	70.78	69.13	47.92	62.52
HAS [17]	54.48	72.55	66.25	51.00	71.87	50.80	66.91	64.67	42.28	55.83
+**Ours**	65.93	89.65	84.45	70.24	88.94	53.32	70.67	69.17	47.71	62.45
CutMix [32]	56.27	64.13	59.08	44.21	65.23	50.17	65.84	63.73	42.85	57.97
+**Ours**	72.96	87.44	79.67	64.93	85.36	53.02	69.92	68.53	46.67	60.16
ADL [6]	52.13	66.75	59.31	45.40	59.49	50.40	66.88	64.50	42.29	56.21
+**Ours**	64.41	86.06	74.48	60.46	81.07	53.05	70.51	68.97	47.04	61.76
DAOL [39]	62.40	81.83	69.87	56.18	74.70	43.26	70.27	68.23	49.68	65.42
+**Ours**	69.67	**94.01**	**84.88**	**74.51**	**90.38**	44.24	**72.08**	**69.97**	**52.17**	**67.68**

Table 3. Comparison with SOTA methods with InceptionV3 (**border** means the best).

Method	CUB-200					ILSVRC			OpenImages	
	T-Loc	G-Loc	B-Loc	pIoU	PxAP	T-Loc	G-Loc	B-Loc	pIoU	PxAP
DGL [25]	50.50	67.64	–	–	–	52.23	68.08	–	–	–
SPG [35]	56.64	–	–	–	–	49.60	64.69	–	–	–
I^2C [36]	55.99	72.60	–	–	–	53.11	68.50	–	–	–
ICLCA [14]	56.10	67.93	–	–	–	49.30	65.21	–	–	–
PAS [2]	69.96	73.65	–	–	–	50.56	64.44	–	–	63.30
IVR [15]	–	–	61.74	–	–	–	–	66.04	–	64.08
UPSP [20]	53.38	72.14	–	–	–	52.73	68.33	–	–	–
PSOL [33]	65.51	–	–	–	–	54.82	65.21	–	–	–
GCNet [18]	–	–	–	–	–	49.06	–	–	–	–
FAM [19]	**70.67**	87.25	–	–	–	**55.24**	68.62	–	–	–
CAM [38]	48.96	63.44	57.14	49.28	70.95	50.75	66.16	63.61	47.51	63.31
+**Ours**	54.75	74.75	65.65	60.34	81.49	52.22	68.84	66.46	49.98	65.91
HAS [17]	52.68	70.89	62.39	52.78	74.07	51.00	66.99	64.26	42.87	59.50
+**Ours**	57.93	79.44	69.65	61.75	83.03	52.22	69.20	66.89	48.44	64.37
CutMix [32]	51.86	66.62	59.44	51.40	74.19	50.72	66.96	64.44	46.30	62.12
+**Ours**	58.48	79.58	68.09	**62.44**	**83.15**	52.60	70.57	**68.04**	49.28	65.23
ADL [6]	49.10	62.62	57.01	49.72	70.06	50.20	66.30	63.66	47.03	63.42
+**Ours**	54.75	74.34	64.87	60.09	81.41	51.63	68.81	66.42	49.22	65.31
DAOL [39]	56.29	80.03	68.01	51.80	71.03	52.70	69.11	64.75	48.01	64.46
+**Ours**	60.07	**89.78**	**76.94**	58.05	72.97	53.87	**71.02**	66.93	**50.79**	**66.89**

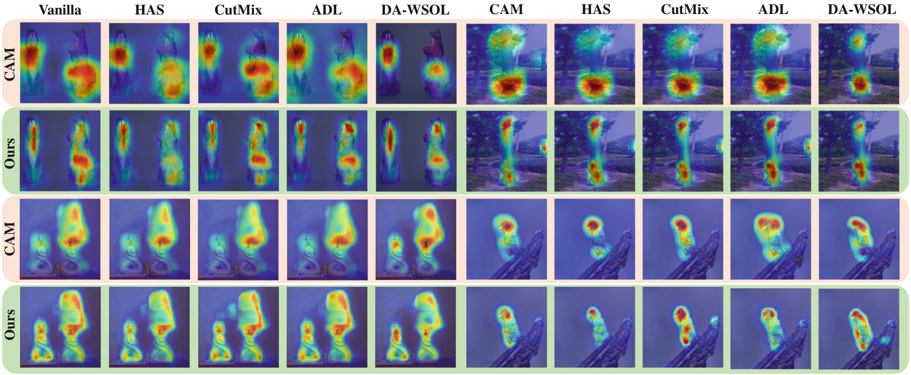

Fig. 3. Visualization on replacing CAM into BagCAMs for different WSOL methods.

classifier to localize objects as CAM. By adopting our BagCAMs to project the classifier into a set of regional localizers, the regional factors of the class of bird can be better concerned, improving nearly 21.38% in G-Loc than the baseline method. Additionally, when more finely evaluated by the pixel-level mask, the improvements of our method are still remarkable, achieving 64.40% pIoU and 84.38% PxAP, which are 17.70% and 18.44% higher than CAM, respectively. As for the larger scale dataset ILSVRC, directly replacing CAM into our BagCAMs in the test process also achieves 3.48% higher performance in G-Loc metric, *i.e.*, correcting the localization of nearly 1,740 images without any fine-tuning process or structure modification. In addition, even using the most recently proposed DAOL [39] that achieves the SOTA performance on the OpenImages dataset, adopting our method can still obviously improve its performance about 2.49% and 2.26%, respectively in the pIOU and PxAP.

Except for the five reproduced methods, the other nine SOTA methods were also used for comparison in Table 2, whose scores are cited from corresponding papers. Our BagCAMs outperforms SOTA methods for nearly all metrics on all three datasets even though engaged in the vanilla WSOL structure, *i.e.*, "CAM + Ours". Only the T-Loc metric of our BagCAMs is lower than the methods that generate class-agnostic localization results and adopt addition stages for classification (noted by underline style) [19,33,37]. This is because our BagCAMs is only adopted in the test process to enhance the localization results, and our classification accuracy is directly determined by the baseline WSOL methods. Moreover, Table 3 also shows the comparison of using InceptionV3 as the feature extractor for WSOL methods to indicate our generalization for the backbone other than ResNet. The results are in accordance with utilizing ResNet, improving the performance on all baseline methods, for example, 11.31% and 9.38% G-Loc improvement for the vanilla structure (CAM) and DAOL on the CUB-200 dataset, respectively. Moreover, our BagCAMs still outperforms other SOTA methods with InceptionV3 on nearly all metric for these three datasets.

Localization maps generated by WSOL methods with our BagCAMs are also visualized in Fig. 3. For localization maps of the vanilla structure, only the most discriminative locations are activated, *e.g.*, the pedestal of the toy, both ends of the pillar, the shade of the lamp, and the head of the bird. Though existing WSOL methods catch more positions of objects, they only enlarge or refine the activation of regions that near the discriminative parts rather than catching more parts of the object. This also visually verifies that the mechanism of CAM limits the performance of these WSOL methods, making the localizer only concern global cues. Profited by the utilization of our base localizer set, more object parts are effectively activated when adopting our BagCAMs to replace CAM for those methods, for example, the head of the toy, the pedestal of the lamp, and the body of the pillar/bird. Moreover, our BagCAMs can generate the localization map on intermediate layers that contains more fining cues such as pixels near the edge of objects, which also contributes to our high performance (Table 6).

Table 4. The best scores of different CAMs on layers of ResNet for CUB-200 dataset

	T-Loc	G-Loc	B-Loc	pIoU	PxAP
CAM	55.31	66.06	59.21	46.70	65.94
PCS	60.27	73.93	65.24	52.05	72.06
Grad	56.68	69.93	61.70	49.51	68.69
Grad++	61.10	73.79	69.14	53.61	76.33
Ours	**70.89**	**87.44**	**76.22**	**64.40**	**84.38**
HAS	54.48	72.55	66.25	51.00	71.87
PCS	53.65	73.24	67.9	54.87	76.72
Grad	56.82	77.79	69.37	55.64	76.77
Grad++	55.31	76.82	70.29	53.34	75.54
Ours	**65.93**	**89.65**	**84.45**	**70.24**	**88.94**
CutMix	56.27	64.13	59.08	44.21	65.23
PCS	57.65	68.13	61.51	48.19	68.74
Grad	60.96	72.68	64.50	52.10	71.49
Grad++	63.17	77.10	67.67	53.77	74.78
Ours	**72.96**	**87.44**	**79.67**	**64.93**	**85.36**
ADL	52.13	66.75	59.31	45.40	59.49
PCS	52.13	66.75	59.31	45.40	59.49
Grad	52.13	66.75	59.31	45.40	59.49
Grad++	53.65	70.89	61.19	44.72	61.14
Ours	**64.41**	**86.06**	**74.48**	**60.46**	**81.07**
DAOL	62.40	81.83	69.87	56.18	74.70
PCS	63.30	84.57	71.49	58.94	76.81
Grad	63.30	84.57	71.49	58.94	76.81
Grad++	66.13	89.60	75.71	63.08	80.23
Ours	**69.67**	**94.01**	**84.88**	**74.51**	**90.38**

Table 5. PxAP on layers of ResNet

Method	Res_1	Res_2	Res_3	Res_4
PCS	42.01	51.36	72.96	65.94
Grad	15.05	19.61	68.69	65.94
Grad++	13.16	32.01	76.33	71.49
Ours	**71.35**	**78.71**	**84.38**	**72.98**

Table 6. PxAP on layers of Inception

Method	Mix_{6b}	Mix_{6c}	Mix_{6d}	Mix_{6e}
PCS	41.42	61.37	75.36	76.32
Grad	28.91	46.62	65.41	76.19
Grad++	26.77	43.26	65.41	68.00
Ours	**78.14**	**81.46**	**82.80**	**81.49**

Table 7. Efficiency (fps) of CAMs

Method	Res_1	Res_2	Res_3	Res_4
PCS	**90.88**	90.40	**91.86**	**91.04**
Grad	89.72	**91.04**	90.94	90.75
Grad++	90.61	89.25	90.62	89.67
Ours	88.44	86.40	87.02	87.77

Table 8. Different weight strategy

	T-Loc	G-Loc	B-Loc	PxAP
CAM	55.31	66.06	59.21	65.94
Ours_1	66.75	82.34	74.80	77.29
Ours_2	70.20	86.20	74.16	81.79
Ours_3	70.89	87.44	76.22	84.38

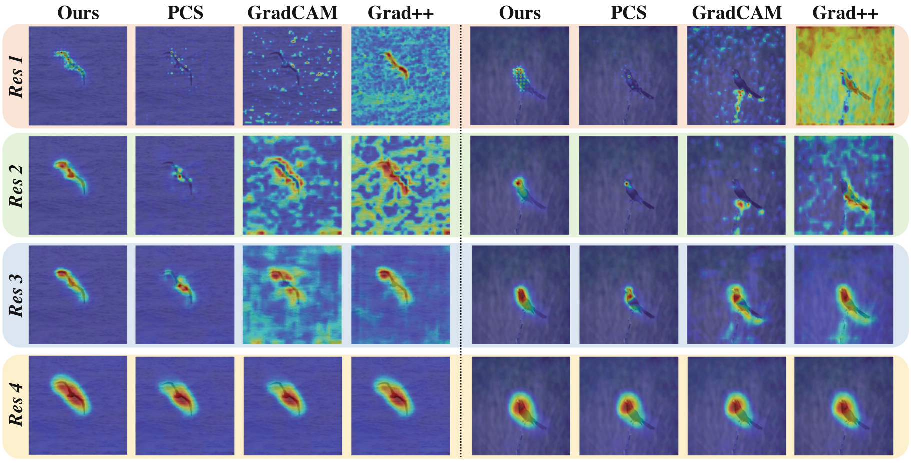

Fig. 4. Localization map generated by the features of different ResNet layer by CAMs.

4.3 Discussion

To deeply investigate the effectiveness of our BagCAMs, we also conducted experiments to compare it with methods that generalize or enhance CAM on GAP-free structures or intermediate layers of CNN, *e.g.*, GradCAM (Grad) [23], GradCAM++ (Grad++) [5], PCS [25]. We adopted the same trained checkpoint for these methods and utilized them to project the classifier in the test process. Except for the original CAM, other methods can be added onto the intermediate layers of the feature extractor. Thus, we generated localization maps on each layer and chose the best performance to report.

Corresponding results are illustrated in Table 4 where the baseline methods, *i.e.*, CAM, HAS, ADL, CutMix, and DAOL, represent directly adopting the classifier for localization as CAM. It shows that for all baseline WSOL methods, our BagCAMs achieves the highest improvement compared with other CAM mechanisms. This is because other CAMs methods all initialize $\hat{\boldsymbol{P}}$ with the global classification results $\mathbf{s}$ for all positions as discussed in Sect. 3.3, resulting in their lower improvement. Unlike them, our BagCAMs adopts $\hat{\boldsymbol{P}}_{:,m}$ to distribute a specific initial localization score for each position m, which helps generate valid localizers and contributes to our outstanding improvement, *e.g.*, 15.68% higher PxAP than using the original CAM for the DAOL.

In addition, our BagCAMs can also achieve satisfactory performance when localizing objects based on features of intermediate layers, which may inspire generating localization maps with higher resolution to consider more details. Table 5 illustrates the PxAP metric for generating localization maps based on the feature of Res_1 (256×56×56), Res_2 (512×28×28), Res_3 (1024×28×28), and Res_4 (2048×28×28). Note that the original CAM can only adopt to the last layer before GAP due to the difference between the number of channels in $\mathbf{W}$ and the intermediate features, thus we did not include the original CAM in this Table 5. It can be seen that GradCAM and GradCAM++ have great performance drops when projected to the prior intermediate layers, *i.e.*, Res_1 and Res_2. Though the PCS, proposed for generating localization results on intermediate layers, slightly decelerates this decline, its PxAP of Res_1 is still 30.97% lower than Res_4. Compared with them, our BagCAMs generates the localization map by bagging the performance of $N \times N$ base localizers, where N is the spatial resolution of the feature map. Thus, for the previous layers with higher resolution, more basic localizers can be defined for bagging, *i.e.*, $3,136$ for Res_1. This makes our BagCAMs achieve 29.34% higher PxAP compared with the best of others, when projected on the feature of Res_1.

Figure 4 also qualitatively visualizes the localization maps generated on the intermediate features. It can be seen that localization maps of GradCAM and GradCAM++ contain more noise on Res_1 and Res_2, and the PCS only activates a few discriminative locations. Compared with them, though our BagCAMs suffers from the grid effect caused by the down-sampling, our localization map can cover more object parts even for Res_1. Finally, the efficiencies of different CAMs are also shown in Table 7, where their mean frame per second (fps) for inferring CUB-200 test are reported. It can be seen that, though considering multiple regional localizers rather than only the global one, the complexity of our BagCAMs is only a bit higher than other methods. This indicates that our method can balance the localization performance and efficiency well.

Except for comparing with other CAM mechanisms, the choice of different weighting strategies, *i.e.*, various settings of the weight matrix $\mathbf{\Lambda}$, were also explored on the CUB-200 dataset. Specifically, we designed three types of BagCAMs: (1) Ours$_1$ that only averages the scores generated by localizers f_n^m, *i.e.*, $\mathbf{\Lambda}^i = \frac{1}{N}\mathbf{I}$. (2) Ours$_2$ that aggregates the scores with the spatially weighting mechanism of GradCAM++ [5], *i.e.*, $\mathbf{\Lambda}^i = diag(\alpha)$. (3) Ours$_3$, the mechanism we used in our paper as defined in Eq. 9, which only selects specific localizers for each position like PCS [25]. Corresponding results are shown in Table 8. It can be seen that using these three weighting mechanisms can all enhance the performance of the baseline methods, profited by adopting the regional localizer set rather than the globally defined classifier. Specifically, when simply averaging the localization scores of the regional localizers (Ours$_1$), the performance improves about 11.35% on the PxAP metric. Adopting the spatial weighting strategy to consider the effect based on each spatial position will bring an additional 4.5% improvement. When grouping the $N * N$ localizers into N clusters that are specifically used for each spatial position to reduce noise as PCS [25], the performance hits the highest, *i.e.*, about 84.38% PxAP. Thus, we suggest adopting this grouping strategy to weight the effect of the regional localizers.

4.4 Conclusion

This paper proposes a novel mechanism called BagCAMs for WSOL to replace CAM [38] when projecting an image-level trained classifier as the localizer to locate objects. Our BagCAMs can be engaged in existing WSOL methods to improve their performance without re-training the baseline structure. Experiments show that our method achieves SOTA performance on three WSOL benchmarks.

Acknowledgements. This work was supported in part by the Beijing Natural Science Foundation under Grant Z210008; in part by the Shenzhen Science and Technology Program under Grant KQTD20180412181221912 and Grant JCYJ20200109140603831.

References

1. Babar, S., Das, S.: Where to look?: Mining complementary image regions for weakly supervised object localization. In: 2021 IEEE Winter Conference on Applications of Computer Vision (WACV) (2021)
2. Bae, W., Noh, J., Kim, G.: Rethinking class activation mapping for weakly supervised object localization. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12360, pp. 618–634. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58555-6_37
3. Benenson, R., Popov, S., Ferrari, V.: Large-scale interactive object segmentation with human annotators. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11700–11709 (2019)
4. Chang, Y.T., Wang, Q., Hung, W.C., Piramuthu, R., Tsai, Y.H., Yang, M.H.: Weakly-supervised semantic segmentation via sub-category exploration. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8991–9000 (2020)
5. Chattopadhay, A., Sarkar, A., Howlader, P., Balasubramanian, V.N.: Grad-CAM++: generalized gradient-based visual explanations for deep convolutional networks. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 839–847. IEEE (2018)
6. Choe, J., Lee, S., Shim, H.: Attention-based dropout layer for weakly supervised single object localization and semantic segmentation. IEEE Trans. Pattern Anal. Mach. Intell. (TPAMI) **43**(12), 4256–4271 (2020)
7. Choe, J., Oh, S.J., Lee, S., Chun, S., Akata, Z., Shim, H.: Evaluating weakly supervised object localization methods right. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3133–3142 (2020)
8. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: a large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 248–255. IEEE (2009)
9. Fan, J., Zhang, Z., Tan, T., Song, C., Xiao, J.: CIAN: cross-image affinity net for weakly supervised semantic segmentation. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), vol. 34, pp. 10762–10769 (2020)
10. Fang, L., Xu, H., Liu, Z., Parisot, S., Li, Z.: EHSOD: CAM-guided end-to-end hybrid-supervised object detection with cascade refinement. In: Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), vol. 34, pp. 10778–10785 (2020)
11. Guo, G., Han, J., Wan, F., Zhang, D.: Strengthen learning tolerance for weakly supervised object localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7403–7412 (2021)
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 770–778 (2016)

13. Hsu, C.C., Hsu, K.J., Tsai, C.C., Lin, Y.Y., Chuang, Y.Y.: Weakly supervised instance segmentation using the bounding box tightness prior. In: Advances in Neural Information Processing Systems (NeurIPS), vol. 32, pp. 6586–6597 (2019)
14. Ki, M., Uh, Y., Lee, W., Byun, H.: In-sample contrastive learning and consistent attention for weakly supervised object localization. In: Proceedings of the Asian Conference on Computer Vision (ACCV) (2020)
15. Kim, J., Choe, J., Yun, S., Kwak, N.: Normalization matters in weakly supervised object localization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 3427–3436 (2021)
16. Kou, Z., Cui, G., Wang, S., Zhao, W., Xu, C.: Improve CAM with auto-adapted segmentation and co-supervised augmentation. In: 2021 IEEE Winter Conference on Applications of Computer Vision (WACV) (2021)
17. Kumar Singh, K., Jae Lee, Y.: Hide-and-Seek: forcing a network to be meticulous for weakly-supervised object and action localization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 3524–3533 (2017)
18. Lu, W., Jia, X., Xie, W., Shen, L., Zhou, Y., Duan, J.: Geometry constrained weakly supervised object localization. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12371, pp. 481–496. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58574-7_29
19. Meng, M., Zhang, T., Tian, Q., Zhang, Y., Wu, F.: Foreground activation maps for weakly supervised object localization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 3385–3395 (2021)
20. Pan, X., et al.: Unveiling the potential of structure preserving for weakly supervised object localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 11642–11651 (2021)
21. Paszke, A., et al.: PyTorch: an imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems (NeurIPS), pp. 8024–8035 (2019)
22. Ramaswamy, H.G., et al.: Ablation-CAM: visual explanations for deep convolutional network via gradient-free localization. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 983–991 (2020)
23. 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: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 618–626 (2017)
24. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2818–2826 (2016)
25. Tan, C., Gu, G., Ruan, T., Wei, S., Zhao, Y.: Dual-gradients localization framework for weakly supervised object localization. In: Proceedings of the 28th ACM International Conference on Multimedia, pp. 1976–1984 (2020)
26. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD Birds-200-2011 dataset (2011)
27. Wan, F., Liu, C., Ke, W., Ji, X., Jiao, J., Ye, Q.: C-MIL: continuation multiple instance learning for weakly supervised object detection. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2199–2208 (2019)
28. Wang, H., et al.: Score-CAM: score-weighted visual explanations for convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 24–25 (2020)

29. Wang, X., Girshick, R., Gupta, A., He, K.: Non-local neural networks. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7794–7803 (2018)
30. Xue, H., Liu, C., Wan, F., Jiao, J., Ji, X., Ye, Q.: DANet: divergent activation for weakly supervised object localization. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 6589–6598 (2019)
31. Yang, S., Kim, Y., Kim, Y., Kim, C.: Combinational class activation maps for weakly supervised object localization. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), pp. 2941–2949 (2020)
32. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J., Yoo, Y.: CutMix: regularization strategy to train strong classifiers with localizable features. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), pp. 6023–6032 (2019)
33. Zhang, C.L., Cao, Y.H., Wu, J.: Rethinking the route towards weakly supervised object localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 13460–13469 (2020)
34. Zhang, X., Wei, Y., Feng, J., Yang, Y., Huang, T.S.: Adversarial complementary learning for weakly supervised object localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1325–1334 (2018)
35. Zhang, X., Wei, Y., Kang, G., Yang, Y., Huang, T.: Self-produced guidance for weakly-supervised object localization. In: Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y. (eds.) ECCV 2018. LNCS, vol. 11216, pp. 610–625. Springer, Cham (2018). https://doi.org/10.1007/978-3-030-01258-8_37
36. Zhang, X., Wei, Y., Yang, Y.: Inter-image communication for weakly supervised localization. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.-M. (eds.) ECCV 2020. LNCS, vol. 12364, pp. 271–287. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-58529-7_17
37. Zhang, X., Wei, Y., Yang, Y., Wu, F.: Rethinking localization map: towards accurate object perception with self-enhancement maps. arXiv preprint arXiv:2006.05220 (2020)
38. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2921–2929 (2016)
39. Zhu, L., She, Q., Chen, Q., You, Y., Wang, B., Lu, Y.: Weakly supervised object localization as domain adaption. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 14637–14646 (2022)
40. Zhu, L., et al.: Unifying nonlocal blocks for neural networks. In: Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 12292–12301 (2021)