Bridging the Gap between Classification and Localization for Weakly Supervised Object Localization

Eunji Kim[1] Siwon Kim[1] Jungbeom Lee[1] Hyunwoo Kim[2] Sungroh Yoon[1,3*]
[1] Department of Electrical and Computer Engineering, Seoul National University [2] LG AI Research
[3] Interdisciplinary Program in AI, AIIS, ASRI, INMC, and ISRC, Seoul National University
{kce407, tuslkkk, jbeom.lee93}@snu.ac.kr, hwkim@lgresearch.ai, sryoon@snu.ac.kr

Abstract

Weakly supervised object localization aims to find a target object region in a given image with only weak supervision, such as image-level labels. Most existing methods use a class activation map (CAM) to generate a localization map; however, a CAM identifies only the most discriminative parts of a target object rather than the entire object region. In this work, we find the gap between classification and localization in terms of the misalignment of the directions between an input feature and a class-specific weight. We demonstrate that the misalignment suppresses the activation of CAM in areas that are less discriminative but belong to the target object. To bridge the gap, we propose a method to align feature directions with a class-specific weight. The proposed method achieves a state-of-the-art localization performance on the CUB-200-2011 and ImageNet-1K benchmarks.

1. Introduction

Object localization aims to find the area of a target object in a given image [5, 13, 18, 19, 23]. However, fully supervised approaches require accurate bounding box annotations, which require a tremendous cost. Weakly supervised object localization (WSOL) has been a great alternative because it requires only image-level labels to train a localization model [3, 4, 17, 21, 27].

The most commonly used approach for WSOL is a class activation map (CAM) [33]. CAM-based methods employ a global average pooling (GAP) layer [12] followed by a fully connected (FC) layer, and generate a CAM with the feature maps prior to the GAP layer. A highly activated area in a CAM is predicted to be an object location. However, it is widely observed that CAM identifies only the most discriminative parts of an object rather than the entire object area, resulting in low localization performance [11, 15, 30].

We ask the question, "*Why does CAM generated from an accurate classifier fail to highlight the entire object area?*" To answer this, we provide a new perspective of decomposing CAM into two terms: (1) activation in a feature map and (2) cosine similarity between the feature vector at each spatial location and the class-specific weight in the FC layer. Fig. 1(a) shows that only the bird's body is highly activated in the CAM of the vanilla model, leaving the wing less activated. However, looking at the activation in the feature map, the wing as well as the body is highly activated. The low similarity of the wing region offsets the activation in the

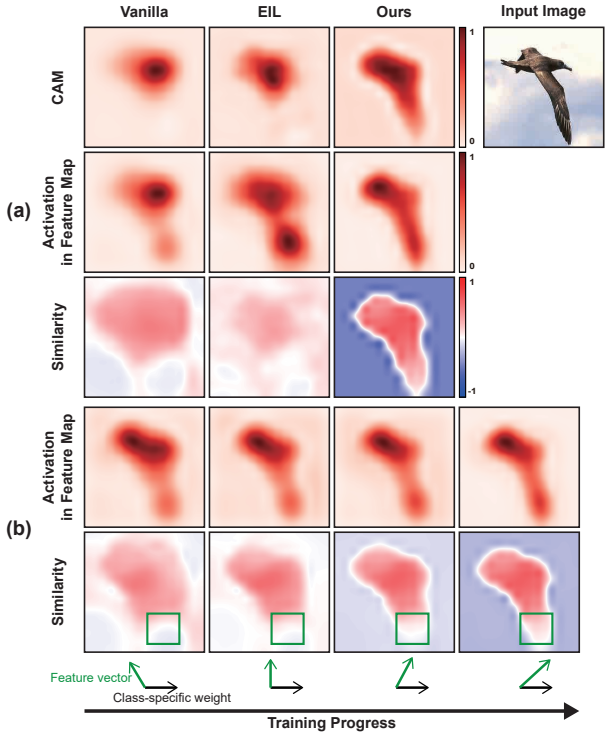

Figure 1. (a) Examples of CAM and decomposed terms from the classifier trained with the vanilla method [33] and with EIL [15]. (b) Visualization of the changes of CAM and decomposed terms as training with our method progresses.

*Correspondence to: Sungroh Yoon (sryoon@snu.ac.kr).

feature map, making the region invisible in the CAM. Here, we find that the low cosine similarity, *i.e*., misalignment of feature directions to the class-specific weights, prevents the less discriminative part belonging to a target object from being highly activated in a CAM. This is because training for classification only considers the feature averaged over all locations, not the feature at each spatial location. This brings the gap between classification and localization.

Although various approaches have been proposed to expand the activated region to the entire object area in a CAM [4, 15, 27, 28, 30, 31], none of them discovered or mitigated the misalignment. Fig. 1(a) shows that EIL [15], one of those approaches, expands the activated region in the feature map. However, it fails to increase the similarity in the object region; hence, the expansion effect is not as large in the CAM as in the activation of the feature map.

To bridge the gap between classification and localization, we propose feature direction alignment, a method to enhance the alignment of feature directions in the entire object region to the directions of class-specific weights while discouraging the alignment in the background region. We also introduce consistency with attentive dropout, which ensures that the target object region has uniformly high activation in the feature map. Fig. 1(b) shows that our method gradually aligns the feature directions to the class-specific weight as the training progresses. The alignment results in high activation of less discriminative regions, *e.g*., wing, in the CAM, enabling accurate localization of the entire object. We evaluate our method on the most widely used WSOL benchmark datasets: CUB-200-2011 [25] and ImageNet-1K [19]. Our method achieves a state-of-the-art localization performance for both datasets.

The contributions of this paper can be summarized as follows:

- We interpret a CAM in terms of the degree of alignment between the direction of input features and the direction of class-specific vectors, and find the gap between classification and localization.
- We propose a method to bridge the gap between classification and localization by aligning feature directions with class-specific weights.
- We demonstrate that our proposed method outperforms other state-of-the-art WSOL methods on the CUB-200-2011 and ImageNet-1K datasets.

2. Related Work

The WSOL method trains a model to localize objects using image-level labels. Zhou *et al*. [33] introduce a CAM to identify the location of a target object via GAP layer [12]. However, it fails to identify the entire object region.

Various methods have been proposed to activate the entire object region in a CAM. HaS [21] trains a classifier using images that are erased with a random patch. ACoL [30] employs two parallel classifiers to identify complementary regions. ADL [2, 4] stochastically drops out the attentive feature in a single forward pass. Ki *et al*. [8] introduced contrastive learning with foreground features and background features. EIL [15] adopts an additional forward pass to classify with the feature whose highly activated regions are erased. SPG [31] utilizes a deep feature to guide a shallow feature and I^2C [32] uses pixel-level correlations between two different images. CutMix [28] combines two patches from different images and assigns a new class label based on the area of each patch. DANet [27] leverages divergent activations with the hierarchy of classification labels.

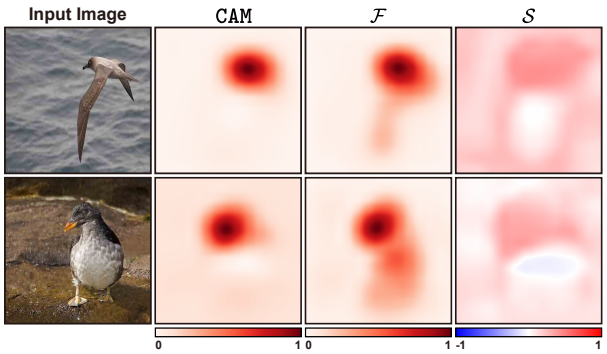

Figure 2. Examples of CAM and decomposed terms $\mathcal{F}$ and $\mathcal{S}$ from a vanilla model. The CAMs and $\mathcal{F}$ are normalized as in $[0, 1]$ for visualization. It shows the misalignment of the feature directions with the class-specific weights.

There have been attempts to obtain localization maps in different ways, pointing out the limitations of CAM-based methods. Pan *et al*. [17] proposed a method to utilize high-order point-wise correlation to generate localization maps. Kim *et al*. [10] proposed a CALM that learns to predict the location of the cue for recognition.

Several normalization methods have been proposed to obtain the bounding boxes around predicted object locations from a continuous localization map. Bae *et al*. [1] proposed several methods to address the bias in GAP, including a new normalization method, PaS, which restricts the maximum value of the activation map. IVR [9] is a normalization method that restricts the minimum value of the activation map.

Some works have adopted an auxiliary module for localization besides classification. GC-Net [14] adopts a separate detector for localization trained with a geometric constraint. FAM [16] generates a class-agnostic foreground map through a memory mechanism. ORNet [26] adopts an additional activation map generator and refines the activation map in an online manner. PSOL [29], SLT-Net [6], and SPOL [24] use two separate networks for classification and localization.

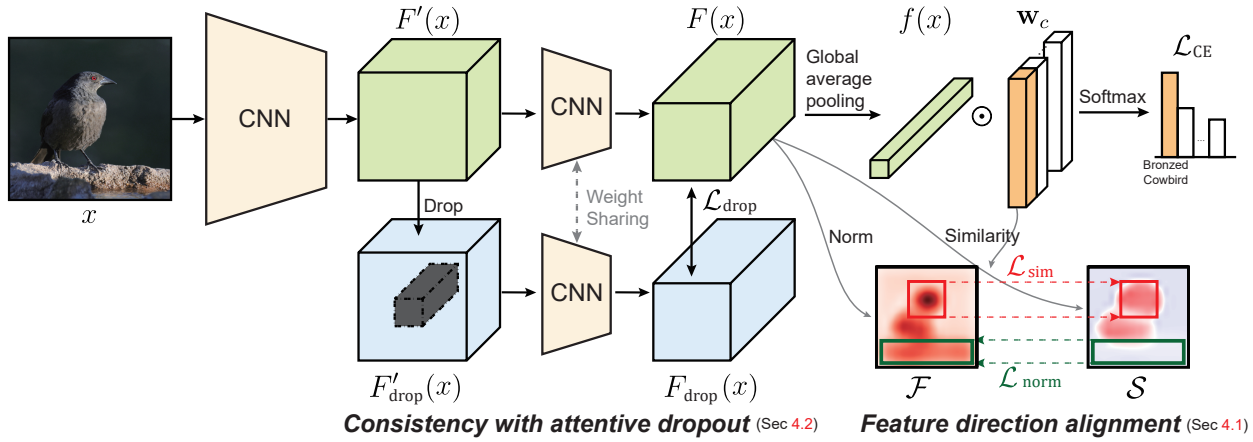

Figure 3. Overview of the proposed method. It consists of two strategies: feature direction alignment and consistency with attentive dropout.

Our method aims to address the gap between classification and localization without adopting any auxiliary module. The methods that adopt additional modules or even separate models use more parameters and computational resources. Therefore, we compare our method mainly with the WSOL methods that use a single branch, for a fair comparison.

3. Finding the Gap with CAM Decomposition

Given an input image x and a typical image classifier comprising convolutional layers and a GAP followed by an FC layer, a CAM for target class c is computed as follows:

$$\texttt{CAM}(x) = \mathbf{w}_c^{\intercal} F(x). \tag{1}$$

$F(x) \in \mathbb{R}^{H \times W \times D}$ is the feature map before the GAP, and $\mathbf{w}_c \in \mathbb{R}^D$ is the weight of the FC layer connected to class c, where H, W, and D are the height, width, and dimension, respectively. Eq. 1 implies that the value of CAM at each spatial location is the dot product of two vectors, $\mathbf{w}_c$ and $F_u(x)$, where $u \in \{1, ..., HW\}$ is the index of spatial location. It can be decomposed as follows:

$$\begin{aligned} \texttt{CAM}_u(x) =& \mathbf{w}_c \cdot F_u(x) \\ =& \|\mathbf{w}_c\| \|F_u(x)\| \underbrace{\frac{\mathbf{w}_c \cdot F_u(x)}{\|\mathbf{w}_c\| \|F_u(x)\|}}_{S(\mathbf{w}_c, F_u(x))}, \end{aligned} \tag{2}$$

where $S(\mathbf{a}, \mathbf{b})$ is the cosine similarity between the two vectors, $\mathbf{a}$ and $\mathbf{b}$. When generating a CAM, target class c is fixed and $\|\mathbf{w}_c\|$ is the same for every u. The CAM value at each position can now be interpreted as the product of the norm of the feature vector at the corresponding location and the similarity between the feature vector and class-specific weight vector. Let $\mathcal{F} \in \mathbb{R}^{H \times W}$ and $\mathcal{S} \in \mathbb{R}^{H \times W}$ be the norm map and the similarity map, respectively, where $\mathcal{F}_u = \|F_u\|$ and $\mathcal{S}_u = S(\mathbf{w}_c, F_u(x))$. Subsequently, CAM can be rewritten as

$$\texttt{CAM}(x) = \|\mathbf{w}_c\| \cdot \mathcal{F} \odot \mathcal{S}. \tag{3}$$

To localize the target object accurately, both $\mathcal{F}_u$ and $\mathcal{S}_u$ should be large for u belonging to the object.

Likewise, the classification score can be interpreted with the output of the GAP, $f(x) = \text{GAP}(F(x)) \in \mathbb{R}^{\text{D}}$.

$$\begin{aligned} \texttt{logit}_c(x) =& \mathbf{w}_c \cdot f(x) \\ =& \|\mathbf{w}_c\| \, \|f(x)\| \, S\left(\mathbf{w}_c, f(x)\right). \end{aligned} \tag{4}$$

Because $\|f(x)\|$ is fixed for x, $\|\mathbf{w}_c\|$ and $S\left(\mathbf{w}_c, f(x)\right)$ determine the logit score of each class c. The scale variation of $\|\mathbf{w}_c\|$ across classes is not very large. Therefore, to classify x correctly, $S(\mathbf{w}_c, f(x))$ must be large for the ground truth class c. Here exists the gap between classification and localization. The classifier is trained to increase $S(\mathbf{w}_c, f(x))$, not $S(\mathbf{w}_c, F_u(x))$ for u belonging to an object region. Cosine similarity is interpreted as the degree of alignment between the directions of the two vectors, meaning that the input feature vector at the object region and class-specific weight vector are not ensured to be aligned with training only for classification. This causes the model to fail to localize the entire object in a CAM.

Fig. 2 shows some examples of norm map $\mathcal{F}$, similarity map $\mathcal{S}$, and CAM from a vanilla model. The less discriminative but object-belonging regions also have noticeably high activation in $\mathcal{F}$, including wings and bodies of birds. However, those regions are not activated in the final CAMs, due to the small values in $\mathcal{S}$. Although $\mathcal{F}$ contains considerable information for localization, its effect diminishes because of the misalignment of the feature directions with the class-specific weight.

In the next section, we propose a method to bridge the gap between classification and localization by aligning feature directions: adjusting the cosine similarity between input features and class-specific weights.

4. Bridging the Gap through Alignment

We describe how to align feature directions in Sec. 4.1. An additional strategy to enhance the effect of the feature direction alignment, consistency with attentive dropout, is introduced in Sec. 4.2. In Sec. 4.3, we describe the overall training scheme. Fig. 3 shows the overview of our proposed method.

4.1. Alignment of Feature Directions

To enhance the activation of the entire object region in CAM, we want the cosine similarity between F_u and $\mathbf{w}_c$ to be high for u belonging to the target object and low for the background region. Because high activation in $\mathcal{F}$ implies that there is a cue for classification at the corresponding location, we divide the region of the feature map into coarse foreground region $\mathcal{R}_{\text{fg}}^{\text{norm}}$ and background region $\mathcal{R}_{\text{bg}}^{\text{norm}}$ based on a normalized $\mathcal{F}$.

$$\begin{aligned} \mathcal{R}_{\text{fg}}^{\text{norm}} &= \{u|\hat{\mathcal{F}}_u > \tau_{\text{fg}}\}, \\ \mathcal{R}_{\text{bg}}^{\text{norm}} &= \{u|\hat{\mathcal{F}}_u < \tau_{\text{bg}}\}, \\ \text{where } \hat{\mathcal{F}} &= \frac{\mathcal{F} - \min_i \mathcal{F}_i}{\max_i \mathcal{F}_i - \min_i \mathcal{F}_i}. \end{aligned} \tag{5}$$

τ_{fg} and τ_{bg} are constant thresholds that determine the foreground and background regions, respectively. Note that τ_{fg} and τ_{bg} are not the same; therefore, there is an unknown region that is not included in either $\mathcal{R}_{\text{fg}}^{\text{norm}}$ or $\mathcal{R}_{\text{bg}}^{\text{norm}}$. To increase $\mathcal{S}_u$ in $\mathcal{R}_{\text{fg}}^{\text{norm}}$ and suppress it in $\mathcal{R}_{\text{bg}}^{\text{norm}}$, we define the similarity loss as follows:

$$\mathcal{L}_{\text{sim}} = -\frac{1}{|\mathcal{R}_{\text{fg}}^{\text{norm}}|} \sum_{u \in \mathcal{R}_{\text{fg}}^{\text{norm}}} \mathcal{S}_u + \frac{1}{|\mathcal{R}_{\text{bg}}^{\text{norm}}|} \sum_{u \in \mathcal{R}_{\text{bg}}^{\text{norm}}} \mathcal{S}_u. \tag{6}$$

There still remains a possibility that some parts of the object region have low activation in $\hat{\mathcal{F}}$. In this case, $\mathcal{L}_{\text{sim}}$ may not be sufficient for the alignment. Therefore, we introduce an additional loss term to increase $\hat{\mathcal{F}}$ in every candidate region belonging to the target object. Because a positive $\mathcal{S}_u$ indicates that u is making a positive contribution to increasing the classification logit, the regions with positive similarity can be treated as candidates for the object region. Therefore, we force this area to be activated. We estimate the object region, $\mathcal{R}_{\text{fg}}^{\text{sim}}$, and background region, $\mathcal{R}_{\text{bg}}^{\text{sim}}$, based on $\mathcal{S}_u$ as

$$\begin{aligned} \mathcal{R}_{\text{fg}}^{\text{sim}} &= \{u|\mathcal{S}_u > 0\}, \\ \mathcal{R}_{\text{bg}}^{\text{sim}} &= \{u|\mathcal{S}_u < 0\}. \end{aligned} \tag{7}$$

With each estimated region, we define the norm loss in a manner similar to Eq. 6, as follows:

$$\mathcal{L}_{\text{norm}} = -\frac{1}{|\mathcal{R}_{\text{fg}}^{\text{sim}}|} \sum_{u \in \mathcal{R}_{\text{fg}}^{\text{sim}}} \hat{\mathcal{F}}_u + \frac{1}{|\mathcal{R}_{\text{bg}}^{\text{sim}}|} \sum_{u \in \mathcal{R}_{\text{bg}}^{\text{sim}}} \hat{\mathcal{F}}_u. \tag{8}$$

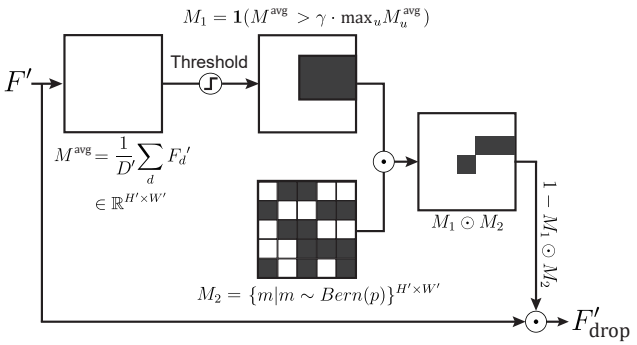

Figure 4. Dropout mechanism of consistency with attentive dropout

For fine-grained classification, such as bird species classification, the object to be recognized is the same across classes. In this case, we define the region with a non-positive similarity with any class as $\mathcal{R}_{\text{bg}}^{\text{sim}}$ and the other as $\mathcal{R}_{\text{fg}}^{\text{sim}}$. In general, the regions $\mathcal{R}_{\text{bg}}^{\text{sim}}$ and $\mathcal{R}_{\text{fg}}^{\text{sim}}$ are defined with a similarity with a target class.

The two loss terms $\mathcal{L}_{\text{sim}}$ and $\mathcal{L}_{\text{norm}}$ operate complementary. Through the minimization of $\mathcal{L}_{\text{sim}}$, the value of $\mathcal{S}$ in the region that is highly activated in $\hat{\mathcal{F}}$ increases. Through the minimization of $\mathcal{L}_{\text{norm}}$, the value of $\hat{\mathcal{F}}$ in the region with high similarity increases. After the joint minimization of $\mathcal{L}_{\text{sim}}$ and $\mathcal{L}_{\text{norm}}$, the activated region in $\hat{\mathcal{F}}$ and that in $\mathcal{S}$ become similar.

4.2. Consistency with Attentive Dropout

We can expect the successful alignment by $\mathcal{L}_{\text{sim}}$ when the estimation of $\mathcal{R}_{\text{fg}}^{\text{norm}}$ and $\mathcal{R}_{\text{bg}}^{\text{norm}}$ is accurate: $\hat{\mathcal{F}}$ is consistently large over the entire object region and small over the background region. Because the value of $\mathcal{F}$ at the most discriminative region is significantly larger than that at the other region, the value of the normalized map $\hat{\mathcal{F}}$ at the less discriminative part but belonging to the object region becomes small.

We introduce consistency with attentive dropout, a method to distribute the activation to the target object region. We adopt L_1 loss between the two feature maps F and F_{drop}: F is the feedforward result of an intermediate feature map F', and F_{drop} is the feedforward result of F'_{drop} obtained by intentionally dropping large activations from F'. Fig. 4 shows the overall process of obtaining F'_{drop} for consistency with attentive dropout. In F', the activation at the spatial location whose channel-wise averaged activation is larger than γ is dropped with probability p. The stochastic dropout prevents all information in the highly activated area from being eliminated. The loss for consistency with attentive dropout is as follows:

$$\mathcal{L}_{\text{drop}} = \|F(x) - F_{\text{drop}}(x)\|_1. \tag{9}$$

There have been several attempts that utilize a similar erasing mechanism [4, 15, 30]. They train a classifier to preserve the predicted labels before and after erasing highly

activated features. In contrast, our method explicitly regularizes a model to yield a similar feature map even after the highly activated features are dropped. This decreases the dependency on the dropped features, resulting in more evenly distributed activation compared to the other methods.

4.3. Training Scheme

With cross-entropy loss for classification, $\mathcal{L}_{\text{CE}}$, the total cost function is defined as follows:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{CE}} + \lambda_{\text{drop}}\mathcal{L}_{\text{drop}} + \lambda_{\text{sim}}\mathcal{L}_{\text{sim}} + \lambda_{\text{norm}}\mathcal{L}_{\text{norm}}, \quad (10)$$

where λ_{drop}, λ_{sim}, and λ_{norm} are hyperparameters for balancing the losses. The feature direction alignment is better applied after training the classifier to some extent to obtain a suitable feature map for classification. Thus, for the first few epochs (*i.e.*, the warm stage), we train a model only with $\mathcal{L}_{\text{CE}}$ and $\mathcal{L}_{\text{drop}}$:

$$\mathcal{L}_{\text{warm}} = \mathcal{L}_{\text{CE}} + \lambda_{\text{drop}}\mathcal{L}_{\text{drop}}. \quad (11)$$

5. Experiments

5.1. Experimental Settings

Datasets. We evaluate our method on two popular benchmarks: CUB-200-2011 [25] and ImageNet-1K [19]. In the CUB-200-2011 dataset, there are 5,994 images for training and 5,794 for testing from 200 bird species. In the ImageNet-1K, there are approximately 1.3 million images in the training set and 50,000 in the validation set from 1,000 different classes.

Evaluation Metrics. Following the work of Russakovsky *et al*. [19], we use Top-1 localization accuracy (Top-1 Loc), Top-5 localization accuracy (Top-5 Loc), and localization accuracy with ground-truth class (GT Loc) as our evaluation metrics. Top-k Loc is the proportion of the images whose predicted bounding box has more than 50% intersection over union (IoU) with the ground-truth bounding box and whose predicted top-k classes include the ground-truth class. GT Loc is the localization accuracy with the ground-truth class, which does not consider the classification result. We also use `MaxBoxAccV2` [3] to evaluate our method. `MaxBoxAccV2`(δ) measures the localization accuracy with ground-truth class with multiple IoU thresholds $\delta \in \{0.3, 0.5, 0.7\}$.

Implementation Details. We evaluate our method using VGG16 [20] and ResNet50 [7] as backbone networks. For VGG16, we adopt the GAP layer following the training settings of the previous work [33]. For ResNet50, we set the stride of the third layer to 1. The attentive dropout is applied before the last pooling layer in VGG16 and after the first block in the fourth layer in ResNet50. We initialize the networks with the pretrained weights using ImageNet-1K [19]. We use a min-max normalization to draw the bounding box from the generated CAM.

Method	Top-1	Top-5	GT Loc
Additional Branch			
SLT-Net [6] CVPR '21	67.8	-	87.6
ORNet [26] ICCV '21	67.74	80.77	86.19
FAM [16] ICCV '21	69.26	-	89.26
Single Branch			
CAM [33] CVPR '16	44.15	52.16	56.00
ADL [4] CVPR '19	52.36	-	75.41
DANet [27] ICCV '19	52.52	61.96	67.70
EIL [15] CVPR '20	56.21	-	-
MEIL [15] CVPR '20	57.46	-	-
DGL [22] ACMMM '20	56.07	68.50	74.63
Ki *et al*. [8] ACCV '20	57.50	-	-
Bae *et al*. [1] ECCV '20	58.96	-	76.30
Pan *et al*. [17] CVPR '21	60.27	72.45	77.29
Ours	**70.83**	**88.07**	**93.17**

Table 1. Comparison of localization performance on the CUB-200-2011 test set, based on VGG16.

5.2. Comparison with State-of-the-art Methods

We compare our method to the recent WSOL methods. For other WSOL methods, we report the localization performance of the original papers or that reproduced by [1, 3, 9, 22][1]. Our method consistently outperforms existing WSOL methods using a single branch, across the datasets and the backbones by a large margin.

Tab. 1 shows the localization performance on the CUB-200-2011 [25] test set, using VGG16 as a backbone. Our method achieves an 11.87%p improvement in Top-1 Loc and a 16.87%p improvement in GT Loc over the work of Bae *et al*. [1], which is the state-of-the-art method among the CAM-based methods. Furthermore, our method outperforms the methods adopting an additional branch for localization. Our method improves Top-1 Loc by 1.57%p and GT-Loc by 3.91%p improvement in GT Loc compared to FAM [16].

Tab. 2 shows the results using ResNet50 as a backbone. It shows that our method consistently outperforms the existing methods by a large margin (>13%p), using a different backbone. Tab. 3 shows the localization performance on the ImageNet-1K [19] validation set, based on VGG16 and ResNet50. Our method achieves the state-of-the-art performance in the ImageNet-1K dataset regardless of the backbone, and only Top-1 Loc with ResNet50 is the second best after I^2C with a marginal difference.

Additionally, we compare our `MaxBoxAccV2` [3] scores with other state-of-the-art methods on the CUB-200-2011 and ImageNet-1K in Tab. 4. It shows that our method outperforms the most recent methods by a large margin for all IoU thresholds with various backbones and datasets. Especially, our method improves the score with IoU threshold of 0.7,

[1] https://github.com/clovaai/wsolevaluation

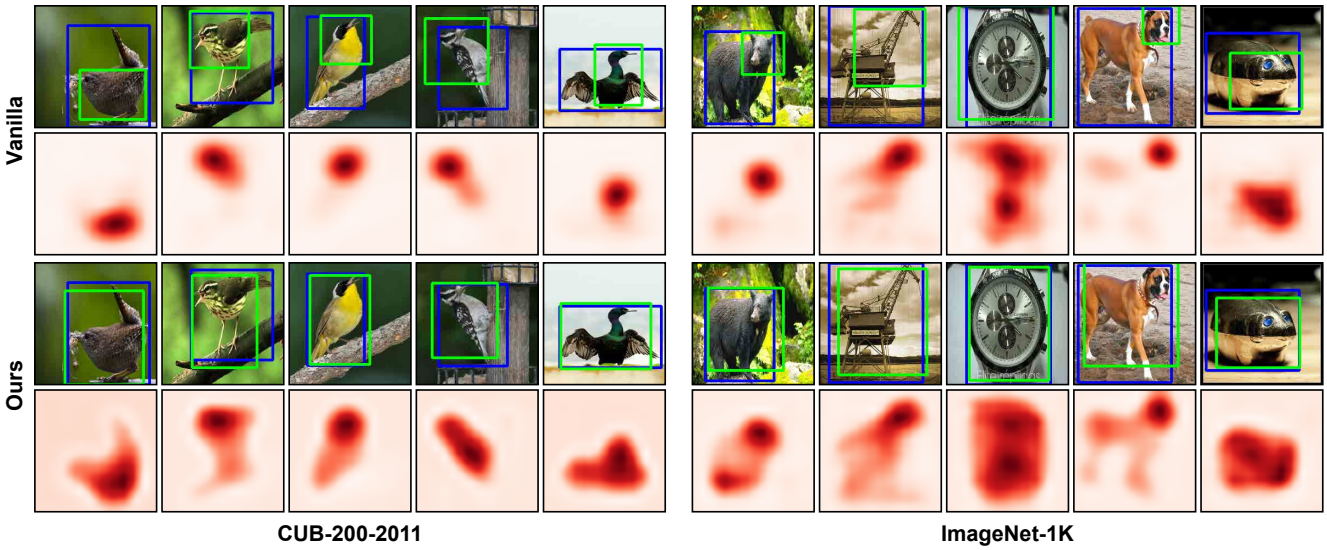

Figure 5. Comparison of localization results from the vanilla method and our method on CUB-200-2011 and ImageNet-1K datasets, using VGG16 as a backbone. Blue boxes denote the ground truth bounding boxes and green boxes denote the predicted bounding boxes.

Method	Top-1	Top-5	GT Loc
CAM [33] CVPR '16	46.91	53.57	-
ADL [4] CVPR '19	57.40	-	71.99
CutMix [28] ICCV '19	54.81	-	-
DGL [22] ACMMM '20	60.82	70.50	74.65
Ki *et al.* [8] ACCV '20	56.10	-	-
Bae *et al.* [1] ECCV '20	59.53	-	77.58
Ours	**73.16**	**86.68**	**91.60**

Table 2. Comparison of localization performance on the CUB-200-2011 test set, based on ResNet50.

which is strict accuracy, by 21.0%p and 17.4%p with VGG16 and ResNet50 on the CUB-200-2011 dataset, respectively, compared with the work of Ki *et al.* [8].

Fig. 5 shows some examples of localization results from the vanilla method [33] and from our method on the CUB-200-2011 and ImageNet-1K datasets. It shows that the model trained with our method captures the target object region more accurately than the vanilla model. On the CUB-200-2011 dataset, while the vanilla model fails to identify the tails, legs, and wings of birds, the classifier trained with our method successfully identifies them.

5.3. Discussion

Feature Direction Alignment. Through the feature direction alignment, we force $\mathcal{S}$ and $\hat{\mathcal{F}}$ to be high in the object region and to be low in the background region. As Fig. 6 shows, the classifier trained with our method yields $\mathcal{S}$ that has a high value in the object region and low value in the background region, different from the vanilla model. It also generates $\hat{\mathcal{F}}$ that has higher activation in less discriminative parts than the vanilla model does. This makes CAM successfully identify the entire object region. As mentioned in Sec. 4.1, the feature direction alignment makes $\hat{\mathcal{F}}$ and $\mathcal{S}$ similar, resulting that CAM becomes also similar with them. We generate a localization map with $\mathcal{F}$ and $\mathcal{S}$ and evaluate the localization performance for each case. We use a min-max

Method	Top-1	Top-5	GT Loc
Backbone: VGG16			
CAM [33] CVPR '16	42.80	54.86	-
ACoL [30] CVPR '18	45.83	59.43	62.96
ADL [4] CVPR '19	44.92	-	-
CutMix [28] ICCV '19	43.45	-	-
I^2C [32] ECCV '20	47.41	58.51	63.90
EIL [15] CVPR '20	46.27	-	-
MEIL [15] CVPR '20	46.81	-	-
Ki *et al.* [8] ACCV '20	47.20	-	-
DGL [22] ACMMM '20	47.66	58.89	64.78
Bae *et al.* [1] ECCV '20	44.62	-	60.73
Pan *et al.* [17] CVPR '21	<u>49.56</u>	<u>61.32</u>	<u>65.05</u>
Ours	**49.94**	**63.25**	**68.92**
Backbone: ResNet50			
ADL [4] CVPR '19	48.23	-	61.04
CutMix [28] ICCV '19	47.25	-	-
Ki *et al.* [8] ACCV '20	48.40	-	-
Bae *et al.* [1] ECCV '20	49.42	-	62.20
I^2C [32] ECCV '20	**54.83**	<u>64.60</u>	68.50
DGL [22] ACMMM '20	53.41	62.69	<u>69.34</u>
Ours	<u>53.76</u>	**65.75**	**69.89**

Table 3. Comparison of localization performance on the ImageNet-1K validation set. The best performance is bold and the second best performance is underlined.

Method	CUB-200-2011								ImageNet-1K							
	VGG16				ResNet50				VGG16				ResNet50			
	δ			Mean	δ			Mean	δ			Mean	δ			Mean
	0.3	0.5	0.7		0.3	0.5	0.7		0.3	0.5	0.7		0.3	0.5	0.7	
CAM [33]	96.8	73.1	21.2	63.7	95.7	73.3	19.9	63.0	81.0	62.0	37.1	60.0	83.7	65.7	41.6	63.7
HaS [21]	92.1	69.9	29.1	63.7	93.1	72.2	28.6	64.6	80.7	62.1	38.9	60.6	83.7	65.2	41.3	63.4
SPG [31]	90.5	61.0	17.4	56.3	92.2	68.2	20.8	60.4	81.4	62.0	36.3	59.9	83.9	65.4	40.6	63.3
ADL [4]	97.7	78.1	23.0	66.3	91.8	64.8	18.4	58.3	80.8	60.9	37.8	59.9	83.6	65.6	41.8	63.7
CutMix [28]	91.1	67.3	28.6	62.3	94.3	71.5	22.5	62.8	80.3	61.0	37.1	59.5	83.7	65.2	41.0	63.3
Ki *et al.* [8]	96.2	77.2	26.8	66.7	96.2	72.8	20.6	63.2	81.5	63.2	39.4	61.3	84.3	67.6	43.6	65.2
HaS + PaS [1]	-	-	-	61.2	-	-	-	61.9	-	-	-	62.1	-	-	-	64.6
CALM [10]	-	-	-	64.8	-	-	-	71.0	-	-	-	62.8	-	-	-	63.4
ADL + IVR [9]	-	-	-	71.5	-	-	-	67.1	-	-	-	63.7	-	-	-	65.1
Ours	**99.3**	**93.2**	**47.8**	**80.1**	**99.4**	**90.4**	**38.0**	**75.9**	**84.8**	**69.2**	**45.9**	**66.6**	**86.7**	**71.1**	**48.3**	**68.7**

Table 4. Comparison of `MaxBoxAccV2` scores on the CUB-200-2011 and ImageNet-1K datasets using various backbones.

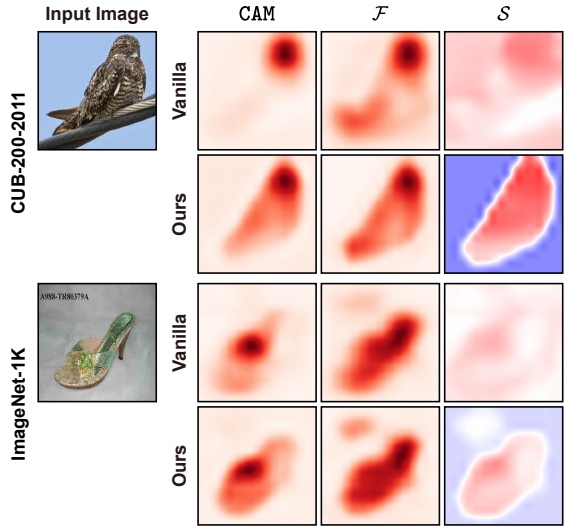

Figure 6. Comparisons of CAM, $\mathcal{F}$, and $\mathcal{S}$ between the vanilla method and our method on the CUB-200-2011 and ImageNet-1K datasets, using VGG16 as a backbone.

Localization map	Top-1	Top-5	GT Loc
CAM	70.83	88.07	93.17
$\mathcal{F}$	69.90	86.68	91.96
$\mathcal{S}$	70.38	87.64	93.13

Table 5. Localization performance with various localization maps on the CUB-200-2011 test set, based on VGG16.

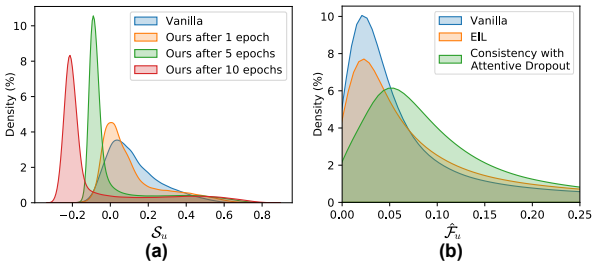

Figure 7. (a) Comparison of density histogram on $\mathcal{S}_u$ with the vanilla method and our method. (b) Comparison of density histogram on $\hat{\mathcal{F}}_u$ with the vanilla method, EIL, and consistency with attentive dropout. The analyzes are performed on the CUB-200-2011 test set using VGG16 as a backbone.

normalization when drawing bounding boxes from $\mathcal{F}$. Since negative values in $\mathcal{S}$ denote the background region, we apply a max-normalization on $\mathcal{S}$. Tab. 5 shows that the localization results with $\mathcal{F}$ and $\mathcal{S}$ also achieve similar localization performance with CAM. This proves the coincidence between CAM, $\mathcal{F}$, and $\mathcal{S}$ with our method.

Fig. 7(a) shows the distribution of $\mathcal{S}_u$ inside the ground truth bounding boxes from the vanilla method and our method. Note that the bounding boxes include not only the target object but also the background region. As the training progresses with our method, the similarity gradually splits into negative and large positive values. This shows that our method effectively increases the similarity for the foreground region and decreases it for the background region. In contrast, for the vanilla method, the similarity is clustered in small positive values, making no distinction between the two.

Consistency with Attentive Dropout. Fig. 7(b) compares the effect of our consistency with attentive dropout on the distributions of $\hat{\mathcal{F}}_u$ with the vanilla method and EIL [15], the state-of-the-art erasing WSOL method. Here, the feature direction alignment with $\mathcal{L}_{\text{sim}}$ and $\mathcal{L}_{\text{norm}}$ is not applied. With the vanilla training, most of $\hat{\mathcal{F}}_u$ are very low. With EIL, overall $\hat{\mathcal{F}}_u$ increase compared with the vanilla method, implying that less discriminative parts become to be highly activated. With consistency with attentive dropout, the distribution of $\hat{\mathcal{F}}_u$ shifts even more to the right. This indirectly shows that our proposed method, consistency with attentive dropout, distributes the activation more over the target object region than the other methods. This results that the consistency

Method	Top-1	Top-5	GT Loc
Align.	62.27	77.48	81.93
EIL [15] + Align.	66.10	82.21	86.78
Attentive Dropout + Align.	**70.83**	**88.07**	**93.17**

Table 6. Comparison of localization performance on the CUB-200-2011 dataset, based on VGG16. Align. denotes the feature direction alignment.

$\mathcal{L}_{\text{drop}}$	$\mathcal{L}_{\text{sim}}$	$\mathcal{L}_{\text{norm}}$	Top-1	Top-5	GT Loc
✗	✗	✗	46.95	57.23	60.74
✓	✗	✗	54.35	70.37	75.06
✗	✓	✗	56.66	71.38	76.10
✗	✓	✓	62.27	77.48	81.93
✓	✓	✗	63.00	79.93	85.35
✓	✓	✓	**70.83**	**88.07**	**93.17**

Table 7. Ablations studies on the CUB-200-2011 test set, based on VGG16.

with attentive dropout achieves higher performance than EIL when used along with feature direction alignment, as shown in Tab. 6. We provide a more detailed analysis in appendix.

5.4. Ablation Study

We perform a series of ablation studies on the CUB-200-2011 dataset using VGG16 as the backbone.

Effect of Each Component. Tab. 7 shows the localization performance of the classifier trained with and without each loss term. Compared to the performance without the proposed loss terms, $\mathcal{L}_{\text{drop}}$ improves the Top-1 Loc by 7.4%p and GT Loc by 14.32%p. The feature direction alignment using only $\mathcal{L}_{\text{sim}}$ improves the Top-1 Loc by 9.71%p and GT Loc by 15.36%p, which shows the largest improvement among the components. Adopting $\mathcal{L}_{\text{norm}}$ improves all metrics more than 5%p. The feature direction alignment using both $\mathcal{L}_{\text{sim}}$ and $\mathcal{L}_{\text{norm}}$ achieves 62.27% of Top-1 Loc and 81.93% of GT Loc, which is higher than the performance reported by Pan *et al*. [17]. Adoption of all components shows the best performance in all metrics.

Sensitivity to Hyperparameters. We analyze the effect of the balancing factors in the loss and the hyperparameters of each loss.

For the balancing factors in loss, we find the best localization performance at 0.5 for λ_{sim}, 0.15 for λ_{norm}, and 3 for λ_{drop}, respectively. As shown in Fig. 8(a), the localization performance is most sensitively affected by λ_{sim}. λ_{norm} insignificantly changes the performance. The performance tends to decrease when the constraint with λ_{drop} becomes too strong as 4.

For the hyperparameters of the feature direction alignment, we set τ_{fg} and τ_{bg} for $\mathcal{L}_{\text{sim}}$ to 0.6 and 0.1, respectively. They determine the coarse foreground and background regions. Fig. 8(b) shows that varying those thresholds has little

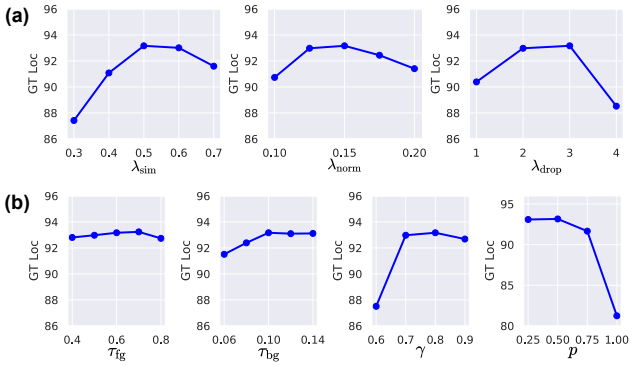

Figure 8. Effect of (a) balancing factors for loss and (b) various hyperparameters.

effect on the performance. The hyperparameters γ and p determine the drop of the activation in the intermediate feature map. γ and p for $\mathcal{L}_{\text{drop}}$ are set to 0.8 and 0.5, respectively. When γ is moderately large between 0.7 and 0.9, there is no significant change in the performance, but when γ is too low, *i.e*., 0.6, the performance decreases. From the results with various p, we observe that stochastic dropout produces little change of GT Loc regardless of the drop probability, but deterministic dropout with a probability of 1.0 yields a significant drop in the localization performance. This indicates that less but sufficient discriminative information should be maintained for a good localization performance.

6. Conclusion

In this paper, we find the gap between classification and localization by decomposing CAM from a new perspective. We claim that the misalignment between the feature vector at each location and class-specific weight causes CAM to be activated only in a small discriminative region. To bridge this gap, we propose a method of aligning feature directions with class-specific weights. We also introduce a strategy to enhance the effect of feature direction alignment. Extensive experiments demonstrate the effectiveness of the proposed method, which outperforms existing WSOL methods by a large margin.

Limitation. There are several hyperparameters to decide in our method. To alleviate the search burden, we discuss a rationale for hyperparameter selection.

Acknowledgements: This work was supported by Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) [NO.2021-0-01343, Artificial Intelligence Graduate School Program (Seoul National University)], LG AI Research, AIRS Company in Hyundai Motor and Kia through HMC/KIA-SNU AI Consortium Fund, and the BK21 FOUR program of the Education and Research Program for Future ICT Pioneers, Seoul National University in 2022.

References

[1] Wonho Bae, Junhyug Noh, and Gunhee Kim. Rethinking class activation mapping for weakly supervised object localization. In *European Conference on Computer Vision*, pages 618–634. Springer, 2020.

[2] Junsuk Choe, Seungho Lee, and Hyunjung Shim. Attention-based dropout layer for weakly supervised single object localization and semantic segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[3] Junsuk Choe, Seong Joon Oh, Sanghyuk Chun, Zeynep Akata, and Hyunjung Shim. Evaluation for weakly supervised object localization: Protocol, metrics, and datasets. *arXiv preprint arXiv:2007.04178*, 2020.

[4] Junsuk Choe and Hyunjung Shim. Attention-based dropout layer for weakly supervised object localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2219–2228, 2019.

[5] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6569–6578, 2019.

[6] Guangyu Guo, Junwei Han, Fang Wan, and Dingwen Zhang. Strengthen learning tolerance for weakly supervised object localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7403–7412, 2021.

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[8] Minsong Ki, Youngjung Uh, Wonyoung Lee, and Hyeran Byun. In-sample contrastive learning and consistent attention for weakly supervised object localization. In *Proceedings of the Asian Conference on Computer Vision*, 2020.

[9] Jeesoo Kim, Junsuk Choe, Sangdoo Yun, and Nojun Kwak. Normalization matters in weakly supervised object localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[10] Jae Myung Kim, Junsuk Choe, Zeynep Akata, and Seong Joon Oh. Keep calm and improve visual feature attribution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

[11] Jungbeom Lee, Eunji Kim, and Sungroh Yoon. Anti-adversarially manipulated attributions for weakly and semi-supervised semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4071–4080, 2021.

[12] Min Lin, Qiang Chen, and Shuicheng Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[13] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017.

[14] Weizeng Lu, Xi Jia, Weicheng Xie, Linlin Shen, Yicong Zhou, and Jinming Duan. Geometry constrained weakly supervised object localization. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI 16*, pages 481–496. Springer, 2020.

[15] Jinjie Mai, Meng Yang, and Wenfeng Luo. Erasing integrated learning: A simple yet effective approach for weakly supervised object localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8766–8775, 2020.

[16] Meng Meng, Tianzhu Zhang, Qi Tian, Yongdong Zhang, and Feng Wu. Foreground activation maps for weakly supervised object localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3385–3395, 2021.

[17] Xingjia Pan, Yingguo Gao, Zhiwen Lin, Fan Tang, Weiming Dong, Haolei Yuan, Feiyue Huang, and Changsheng Xu. Unveiling the potential of structure preserving for weakly supervised object localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11642–11651, 2021.

[18] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015.

[19] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[20] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[21] Krishna Kumar Singh and Yong Jae Lee. 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*, pages 3544–3553. IEEE, 2017.

[22] Chuangchuang Tan, Guanghua Gu, Tao Ruan, Shikui Wei, and Yao Zhao. Dual-gradients localization framework for weakly supervised object localization. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1976–1984, 2020.

[23] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10781–10790, 2020.

[24] Jun Wei, Qin Wang, Zhen Li, Sheng Wang, S Kevin Zhou, and Shuguang Cui. Shallow feature matters for weakly supervised object localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5993–6001, 2021.

[25] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-UCSD birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

[26] Jinheng Xie, Cheng Luo, Xiangping Zhu, Ziqi Jin, Weizeng Lu, and Linlin Shen. Online refinement of low-level feature based activation map for weakly supervised object localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 132–141, 2021.

[27] Haolan Xue, Chang Liu, Fang Wan, Jianbin Jiao, Xiangyang Ji, and Qixiang Ye. Danet: Divergent activation for weakly supervised object localization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6589–6598, 2019.

[28] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6023–6032, 2019.

[29] Chen-Lin Zhang, Yun-Hao Cao, and Jianxin Wu. Rethinking the route towards weakly supervised object localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13460–13469, 2020.

[30] Xiaolin Zhang, Yunchao Wei, Jiashi Feng, Yi Yang, and Thomas S Huang. Adversarial complementary learning for weakly supervised object localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1325–1334, 2018.

[31] Xiaolin Zhang, Yunchao Wei, Guoliang Kang, Yi Yang, and Thomas Huang. Self-produced guidance for weakly-supervised object localization. In *Proceedings of the European conference on computer vision (ECCV)*, pages 597–613, 2018.

[32] Xiaolin Zhang, Yunchao Wei, and Yi Yang. Inter-image communication for weakly supervised localization. In *European Conference on Computer Vision*, pages 271–287. Springer, 2020.

[33] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2921–2929, 2016.