Dual-Gradients Localization Framework for Weakly Supervised Object Localization

Chuangchuang Tan
18120335@bjtu.edu.cn
Institute of Information Science, Beijing Jiaotong University, Beijing Key Laboratory of Advanced Information Science and Network Technology
Beijing, China 100044

Guanghua Gu
guguanghua@ysu.edu.cn
School of Information Science and Engineering, Yanshan University, Hebei Key Laboratory of Information Transmission and Signal Processing
Qinhuangdao, China 066004

Tao Ruan
Shikui Wei
Yao Zhao*
16112064@bjtu.edu.cn
shkwei@bjtu.edu.cn
yzhao@bjtu.edu.cn
Institute of Information Science, Beijing Jiaotong University, Beijing Key Laboratory of Advanced Information Science and Network Technology
Beijing, China 100044

ABSTRACT

Weakly Supervised Object Localization (WSOL) aims to learn object locations in a given image while only using image-level annotations. For highlighting the whole object regions instead of the discriminative parts, previous works often attempt to train classification model for both classification and localization tasks. However, it is hard to achieve a good tradeoff between the two tasks, if only classification labels are employed for training on a single classification model. In addition, all of recent works just perform localization based on the last convolutional layer of classification model, ignoring the localization ability of other layers. In this work, we propose an offline framework to achieve precise localization on any convolutional layer of a classification model by exploiting two kinds of *gradients*, called Dual-Gradients Localization (DGL) framework. DGL framework is developed based on two branches: 1) Pixel-level Class Selection, leveraging gradients of the target class to identify the correlation ratio of pixels to the target class within any convolutional feature maps, and 2) Class-aware Enhanced Maps, utilizing gradients of classification loss function to mine entire target object regions, which would not damage classification performance. Extensive experiments on public ILSVRC and CUB-200-2011 datasets show the effectiveness of the proposed DGL framework. Especially, our DGL obtains a new state-of-the-art Top-1 localization error of 43.55% on the ILSVRC benchmark.

*Yao Zhao is the corresponding author.

MM '20, October 12–16, 2020, Seattle, WA, USA

ACM ISBN 978-1-4503-7988-5/20/10…$15.00
https://doi.org/10.1145/3394171.3413622

CCS CONCEPTS

• **Computing methodologies** → *Interest point and salient region detections*; *Object identification.*

KEYWORDS

Weakly Supervised Object Localization; Pixel-level Class Selection; Class-aware Enhanced Maps; Gradients of Loss Function; Gradients of Target Class

ACM Reference Format:
Chuangchuang Tan, Guanghua Gu, Tao Ruan, Shikui Wei, and Yao Zhao. 2020. Dual-Gradients Localization Framework for Weakly Supervised Object Localization. In *Proceedings of the 28th ACM International Conference on Multimedia (MM '20), October 12–16, 2020, Seattle, WA, USA.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3394171.3413622

1 INTRODUCTION

Weakly supervised learning has made remarkable progress recently. Existing methods have been successfully applied on object localization [2, 9, 17, 20, 28, 31, 33–37], object detection [3, 10, 23, 30] and segmentation [1, 6, 8, 13, 24–26, 32]. Weakly Supervised Object Localization (WSOL) refers to mining information of object locations by only using image-level annotations. Compared with fully-supervised object localization, WSOL uses much cheaper annotations to learn bounding boxes of objects. Existing WSOL approaches discovered target object regions by using a classification network. During training for classification task, CNN models learn the discriminative pattern that makes different object regions contribute to different classes in a given image. Therefore, the location of objects can be generated by mining activation map of each class.

Some previous works have been proposed to produce object attention map of a specific class by using a classification network. Zhou *et al.*[36] proposed Class Activation Mapping (CAM) approach to discover class localization map by revisiting global average pooling (GAP) layer. Unfortunately, since its classification model tends to focus on the most discriminative parts of objects, the CAM [36] can usually localize a small part of objects rather than the whole area of the target objects. To tackle this weakness, recent works proposed various technologies to expand activation regions, and

Figure 1: Observation. (a) Source images. (b-d) Attention maps produced on *Mixed_6g*, *Mixed_6f* and *Mixed_6e* layer, respectively. Not only the last convolutional layer have the localization ability to achieve WSOL. Some layers have better localization performance than the last convolutional layer e.g., on InceptionV3 model, *Mixed_6e* and *Mixed_6f* layers localize entire dog, and the last convolutional layer *Mixed_6g* only discover the head of dog.

then performed CAM to localize the entire object regions. Some previous works [2, 20, 24, 31, 34] proposed erasing-based approaches to discover more regions. Other works [28, 29, 35] exploited spatial correlation information in feature maps to highlight the entire object regions. However, two issues exist in recent WSOL approaches: 1) the localization map can only be produced in the last convolutional layer, which limits the scope for searching the best locations. 2) Both localization and classification tasks are trained online based on a single classification model, so it is difficult to obtain optimal models for both tasks by using only image-level labels. It will lead to the incomplete object region.

To address above-mentioned problems, we propose a simple but effective WSOL framework to excavate the whole object regions on any convolutional layer of classification model, called Dual-Gradients Localization framework. In particular, we investigate the CAM problem from a new view, i.e., gradient view. In this way, any convolutional layer in classification model can be employed for producing localization map, and the searching scope of informative information can expanded for obtaining optimal performance for different application scenarios. In the gradient view, a Pixel-level Class Selection approach is proposed to produce localization map. As shown in Figure 1, the best localization is not always obtained in the last convolutional layer.

Furthermore, we propose Class-aware Enhanced Maps to exploit the integral object regions. Class-aware Enhanced Maps utilizes gradients of classification loss function to enhance information of the specific class on any convolutional layer. It is inspired by the following observations. During testing of classification model, convolutional feature maps predicted to same class are pulled to the same classification region. We believe that class localization map only highlights the discriminative parts of objects when the feature maps are close to the decision boundaries, and the feature maps located at center of the classification region have more information of target class, which can highlight more object regions. To make feature maps have more information of specific class, our key idea of Class-aware Enhanced Maps is pulling the feature maps toward to center of the classification region.

We achieve new state-of-the-art performance on ILSVRC and CUB-200-2011 dataset. Compared with existing WSOL approaches, the proposed DGL can be easily applied to any WSOL approach to improve localization performance. To sum up, the main contributions of this paper are as follows:

- We develop a novel Dual-Gradients Localization (DGL) framework. It is a generation of CAM as a general localization framework for WSOL.
- We provide a new perspective to explain CAM, and further propose a more generic method, called Pixel-level Class Selection. Our approach proves that any convolutional layer of classification model has the localization ability.
- We propose a novel Class-aware Enhanced Maps method which proves that some middle layers have better localization performance than the last convolutional layer.

2 RELATED WORK

Convolutional neural network with full supervise has excellent progress on object localization and detection tasks which are fundamental challenge in computer vision. Although full-supervised methods [7, 12, 14, 15, 18, 19] have demonstrated great performance on object localization and detection, detailed human annotations are expensive and sometimes unaffordable for fully supervising. Weakly-supervised approaches use cheaper annotations to localize object of interest, like image-level label.

Recently, Zhou *et al.*[36] proposed CAM to perform object localization by revisiting global average pooling (GAP) layer [11] on an classification network. However, classification task only relies on the most discriminative regions, resulting in CAM only localizes a part of objects instead of the entire of object. To overcome the weakness, the CAM-based approaches modified from CAM are constantly proposed. Those methods leverage various technologies to expand activation regions on convolutional layers. Most of recent WSOL methods belong to CAM-based approaches.

Previous works probe various technologies to discover integral object regions, consist of erasing [2, 34], data argumentation [20, 31], fusion maps [29], divergent activation [28], self-produced guidance learning [35]. Singh *et al.*[20] proposed a data argumentation approach Hide and Seek (Has) to seek more object regions by randomly hiding an image with patches. Has forced CNN network to focus on object parts that are low-related to classification. However, it locked the high-level guidance and could force CNN focus on background instead of objects. CutMix [31] is an another data argumentation method. It cut patches from training images and past them to other training images where the target labels also mixed. Zhang *et al.*[34] proposed Adversarial Complementary Learning (ACoL) by adopting two parallel-classifiers. ACoL utilized the first classifier to localize the discriminative regions, and compelled second classifier to discover complementary object regions. Similarly, Junsuk *et al.*[2] proposed Attention-based Dropout Layer (ADL) that erased high-scoring regions and forced network to have attention to low-scoring regions within feature maps. SPG [35] leveraged high confident object regions as auxiliary supervision to force classification model to learn more object regions. CCAM [29] introduced a combination strategy to highlight more accurate object regions. It applied activation map of top-1 probability class to discover foreground regions, and utilized activation maps of bottom-10 probability classes to highlight background regions, then

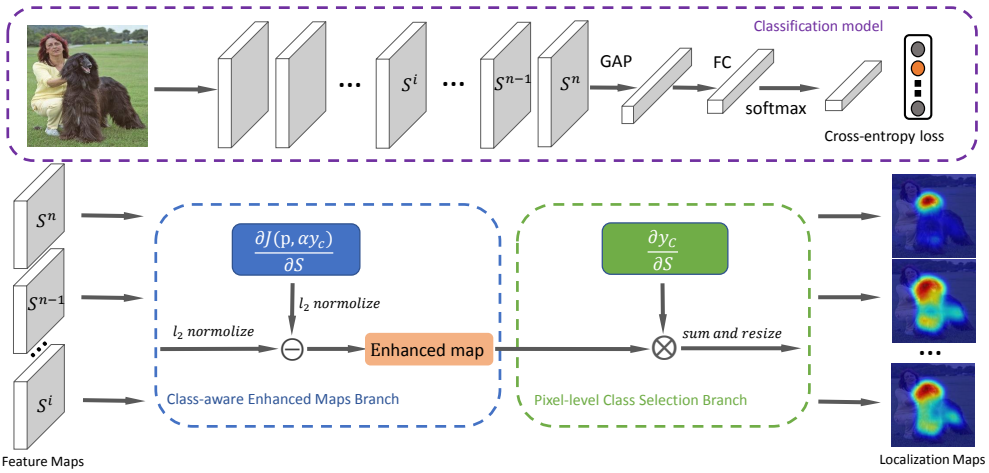

Figure 2: Overview of the proposed Dual-Gradients Localization framework. Our approach can achieve localization on any convolutional layer S. In Class-aware enhanced Maps branch, we produce Enhanced maps by utilizing gradients of classification loss function, which enhances specific-class information to mine entire object regions. Pixel-level Class Selection branch leverages gradients of target class to indicate the importance of pixels on feature maps S for target class, and outputs the localization map.

localization maps were generated by adding the foreground map and subtracting the background map. DANet [28] proposed a divergent activation to extract the discriminative and complementary object regions. All of those works just perform localization on the last convolutional layer of classification model, ignoring the similar ability of other layers. Moreover, most of those works attempt to train classification model for both classification and localization. In our experiments, we demonstrate that any convolutional layer have the localization ability and some layers have better performance than the last convolutional layer in a offline manner.

3 APPROACH

3.1 Overview

In this section, we first provide the overall architecture of our Dual-Gradients Localization (DGL) framework, shown in Figure 2. It consists of two key branches, *i.e.* Pixel-level Class Selection, Class-aware Enhanced Maps. Then, we revisit CAM, and introduce a new view to explain CAM. Next, details of Pixel-level Class Selection and Class-aware Enhanced Maps will be described. Finally, we integrate the two key branches to produce the final localization map, which localizes the entire object regions on any convolutional layer.

3.2 Revisit Class Activation Mapping

The proposed DGL framework is training-free and mainly aims to localize the objects of interest during the inference stage, which has the similar purpose with the well-known CAM [36]. Hence, we first revisit the CAM before introducing the proposed method.

To perform the localization, CAM combines the weights of the final FC layer with the last convolutional feature maps to produce a class-aware localization map, then binarizes it to acquire the excepted object mask. Formally, given a classification network composed of a FCN model, a GAP layer, a FC layer and softmax layer. Let $S^i \in \mathbb{R}^{W \times H \times K}, i = 1, 2, 3, ..., n$ represents feature maps of the i_{th} convolutional layer, where K is the channel dimension, W and H is width and height. Then the feature maps of last convolutional layer S^n are input into the GAP layer followed by the FC layer. For channel k, the output of the GAP layer is donated as $G_k = \frac{\sum_{i,j}(S_k^n)_{i,j}}{W \times H}$, where $(S_k^n)_{i,j}$ is the activation map of channel k in S^n at the i_{th} row and the j_{th} column. The feature vector $l \in \mathbb{R}^C$ obtained by FC layer is input into softmax layer for classification, where C is the total number of classes.

We denote the weights of FC layer as $w \in \mathbb{R}^{K \times C}$. Thus, for a given target class c, the score l_c is defined as

$$l_c = \sum_k w_k^c G_k \tag{1}$$

where w_k^c is the weights of the FC layer at the k_{th} row and the c_{th} column. Here, the bias term is ignored, and it has no effect on classification results. Essentially, w_k^c indicates the contribution of G_k for the score l_c. Then, the class activation map M_c, for class c, will be produced by aggregating the feature maps S^n.

$$M_c = \sum_k w_k^c S_k^n \tag{2}$$

M_c is the localization map of CAM. It can be observed that there are no direct contact between the convolutional layers and weights of the final FC layer except for the last convolutional layer. According to this theory, only the last convolutional layer can perform localization, which limits the scope for searching the best locations.

3.3 Pixel-level Class Selection

CAM utilized a simple strategy to generate localization maps. However, its limitations affect the localization performance, i.e, CAM cannot make use of the localization ability of the other feature maps. In this section, we propose a more generic module to perform localization compared with CAM, and prove that CAM is exactly a special case of our method.

Due to $l \in \mathbb{R}^C$ obtained by FC layer, $l = Gw + b$. The gradients of l_c with respect to feature vector G can be defined as

$$w^c = \frac{\partial l_c}{\partial G} \tag{3}$$

Owing to GAP layer, we can observe that $\{\frac{\partial l_c}{\partial S_k^n}\}_{i,j} = \frac{\partial l_c}{\partial G_k}$. So w^c can be replaced by $\frac{\partial l_c}{\partial S^n}$ to obtain M_c. M_c can be defined as

$$M_c = \sum_k \frac{\partial l_c}{\partial S_k^n} S_k^n \tag{4}$$

We can achieve localization on any convolutional layer, because each convolutional layer S^i can compute the gradients of target class. Therefore, the object localization map M_c^i produced on the convolutional layer S^i for class c can be obtained as follows:

$$M_c^i = \sum_k \frac{\partial l_c}{\partial S_k^i} S_k^i, i = 1, 2, ..., n-1 \tag{5}$$

Hence $\frac{\partial l_c}{\partial S_k^i}$ is pixel-level, which indicates the importance of the pixels of S^i for class c. So in our localization frame, we can achieve object localization on any convolutional layer. Note that Pixel-level Class Selection is a generalization to CAM, which will be degraded to CAM, when $i = n$ in Equation 5.

We prove that CAM is a special case of Pixel-level Class Selection. CAM neglects the influence of GAP layer during back propagation of CNN. CAM actually achieves localization by employing a weighted sum of feature maps and gradients of target class, instead of weights of the final FC layer. During back propagation, gradients of target class with respect to pixels within the last convolutional feature maps are equal to specific class weights of the final FC layer, due to GAP layer. According to our thoughts, we can achieve WSOL on any convolutional layer of classification model.

Relation of Grad-CAM and Pixel-level Class Selection. Grad-CAM is also a generalization to CAM. It looks that Pixel-level Class Selection is similar to Grad-CAM. Actually, they are different on several fields. Firstly, Grad-CAM tried to solve the problem that CAM must be performed on classification network with GAP layer. Differently, we propose a new view way to explain CAM and further develop it. Secondly, Grad-CAM still performed localization on the last convolutional layer. We can achieve localization on any convolutional layer of classification network with GAP layer. Lastly, Grad-CAM summed the gradients on channels to generate localization maps. Our approach directly applies gradients to produce a pixel-level selection, which keeps the spatial information.

To sum up, we provide a simple but useful object localization strategy on each convolutional layer, but it still only localizes the most discriminative regions, and fails to highlight the entire object regions. We further propose Class-aware Enhanced Maps to mine the entire object regions.

3.4 Class-aware Enhanced Maps

A contradiction puzzles WSOL that classification task tends to recognize the most discrimination region of object while localization task tends to localize the entire objects on a single CNN model. CAM performed on CNN model is only trained for classification resulting in only discovering the discrimination regions of object. Several methods attempt to expand activation regions by training classification model for localization. It is hard to achieve a good tradeoff between the two tasks, if only classification labels are employed for training on a single classification model. We propose an offline approach to alleviate this contradiction, named Class-aware Enhanced Maps, which exploits the gradients of classification loss function to mine entire target object regions on a trained classification model.

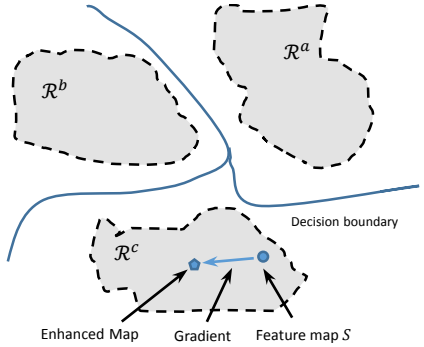

Figure 3: Class-aware Enhanced Maps. Here we show our intuition of Class-aware Enhanced Maps. A trained classification model pulls feature maps with label c to the classification regions $\mathcal{R}^c$. Supposing to the feature maps S predicted to class c, we believe that S has too less information of class c to localize the entire object regions, when S is close to decision boundary and far away the center of $\mathcal{R}^c$. Our main idea is pulling the feature maps toward inside of $\mathcal{R}^c$ along with gradients of classification loss function.

Class-aware Enhanced Maps is inspired by following observations. We assume that a classification CNN model has C outputs where C is the number of classes. Therefore, the classifier can be defined as $f^{cla} : \mathbb{R}^{W\times H\times K} \rightarrow \mathbb{R}^C$. During training for classification, f^{cla} tries to gradually form decision boundaries and pull images with same label together. During testing, CNN outputs feature maps and the predicted probabilities of all classes on different layers. Those feature maps of a convolutional layer with same highest predicted class also locate the same regions. Let classification region $\mathcal{R}^c$ define the region of the space where highest predicted class of f^{cla} is class c. A geometric interpretation is shown in Figure 3. We believe that feature maps predicted to class c only capture the discrimination parts of objects when the feature maps close the boundary of $\mathcal{R}^c$, and the feature maps located at center of classification regions $\mathcal{R}^c$ can highlight more object regions. To enforce feature maps to have more information of specific class, our key idea of Class-aware Enhanced Maps is pulling the feature maps toward inside of the classification region for specific-class.

We leverage gradients of classification loss function with regard to feature maps as the direction of closing center of the classification region $\mathcal{R}^c$. Let A_c^i denote enhanced maps on i_{th} convolutional layer for class c, vector p denote the outputs of classification model, and f denote a mapping function which pulls S^i close center of $\mathcal{R}^c$.

$$A_c^i = f(S^i, \frac{\partial cost(p, y_c)}{\partial S^i}) \tag{6}$$

There are many ways to implement function f. We adopt a simple but effective function to produce enhanced maps rich in information of target class. Feature maps subtract gradients after applied l_2 *normalization* along all dimensions. In addition, when computing gradients, one-hot target label multiplies a constant α for capturing enough information of target class. So, A_c^i can be calculated by

$$A_c^i = l_2(S^i) - l_2(\frac{\partial cost(p, \alpha y_c)}{\partial S^i})) \tag{7}$$

In our experiment, we emprically set α to 10, and set the *cost* function to cross-entropy loss. The hyperparameter α is used to capture enough information of target class. We expect that a sharper signal on the target class could enhance more information of the target, and further enforce the feature maps to generate better object regions. Class-aware Enhanced Maps mines more object regions by enhancing the information of class c. The enhanced maps will be visualized and qualitative analysis in Section 4.4.

3.5 Localization Map Generation

For generating final localization map $\overline{M}_c^i$ to localize entire object regions on feature maps S^i for class c, we replace feature maps S^i with enhanced maps A_c^i. So we obtain $\overline{M}_c^i$ by,

$$\overline{M}_c^i = \sum_k \frac{\partial l_c}{\partial S_k^i} \{A_c^i\}_k \tag{8}$$

Hence $\overline{M}_c^i$ localizes the whole object by enhancing the information of class c, and selecting the important pixels of feature maps on i_{th} convolutional layer for target label y_c.

During testing, we generate localization maps $\overline{M}_c^i$ by using the class with the highest predicted scores. For producing bounding boxes, we resize localization maps $\overline{M}_c^i$ to the size of the original images by bilinear interpolation and apply the same strategy utilized in [36] to segment the heat-map. Meanwhile, we use grid search method to adjust thresholds for generating bounding boxes.

4 EXPERIMENTS

4.1 Dataset and evaluation

We evaluate the proposed approach on ILSVRC [16] and CUB-200-2011 [22]. ILSVRC 2015 dataset has more than 1.2 million images of 1000 classes for training. CUB-200-2011 consists 11788 images of 200 species of birds with 5994 images for training and 5794 for testing. We compare our approach with other methods on CUB-200-2011 test set and ILSVRC validation set containing 50000 images.

We use three evaluation metrics suggested by [16] to evaluate the proposed method: Top-1 localization error (*Top-1 loc error*), Top-5 localization error (*Top-5 loc error*) and Localization error with known ground-truth box (*GT-known Loc error*).

4.2 Implementation details

We evaluate the proposed DGL framework on following popular CNNs: VGGnet [19], InceptionV3 [21], Resnet50 [4], Resnet50 [4] with SE block [5], and modify those backbones as following baselines[2, 35, 36]. For VGG16 network, we remove the layers after *conv5_3*, and add a convolutional layer named *'conv5_4'*, followed by a GAP layer and a FC layer with softmax function. For InceptionV3, we use the customized InceptionV3 as backbone, remove the layers after the second Inception block which strides are 1, and add two convolutional layers named *'mixed-6f'* and *'mixed-6g'*. Finally, a GAP layer and a FC layer are added to classify. For ResNet50 and ResNet50-SE, we set the stride of last two blocks to 1, resulting in a mapping resolution of 28×28.

We train the models pre-trained on ILSVRC [16]. For a fair comparison, the input images are randomly cropped to 224×224 after being resized to 256×256. During testing, we directly reshape input images to 224×224 and average 10 crops (4 corners plus center, same with horizontal flip) to obtain localization maps and classification results respectively. We set the same hyper-parameters for all experiments: batch size 32, weight decay 0.0005, initial learning rate 0.002 and the learning rate of added layers is 10 times. For training on ILSVRC, we fine-tune all models 6 epochs. The learning rate is decreased by a factor of 10 after every two epochs. For training on CUB-200-2011, we fine-tune all models 125 epochs and divide the learning rate by 10 at 75, 100 epochs. We implement the models by using tensorpack [27] on Tensorflow and train the models using GeForce RTX 2080 Ti GPU with 11GB memory.

4.3 Object localization quantitative results

In this subsection, we will report quantitative results of our approach on ILSVRC and CUB-200-2011. For demonstrating the effectiveness of our approach, we generate the localization map on multiple layers of different backbones, as shown in Table 1.

We first analyze localization performance on ILSVRC val set. On VGGnet, *conv4_2 - conv5_3* layers have better performance in term of *gt-know loc error*, compared with *conv5_4* layer. In addition, the *Top-1 loc error* on *conv5_2* layer is 52.34%, which outperforms *conv5_4* layer by 3.55%. On InceptionV3, our approach achieves 50.72%, 48.79%, 47.77%, 48.25% of *Mixed_6g, Mixed_6f, Mixed_6e, Mixed_6d* layer in term of *Top-1 loc error*, respectively. The *Mixed_6e* layer outperforms other layers on three metrics. On Resnet50, the *top-1 loc error* of *group2/block5_shortcut* is 46.59%, and gives a boost of 2.18% over *group3/block2_shortcut* layer. When Resnet50-SE is utilized as a backbone, the performance can be further boosted to 43.55% on *group2/block5_shortcut*.

We observe that the third convolutional layer from last outperforms other layers on non-skipping-layer-network, like VGG16 and InceptionV3. The localization performance decreases as performed on the lower-level layer from the third convolutional layer from last. We think this is because the lower-level convolutional layers contain general features unrelated to target class, lead to only discovering the foreground regions of image. Compared with VGGnet and InceptionV3, there are more layers have similar localization performance on Resnet50 and Resnet50-SE. We believe it is caused by residual block that leads gradients to lower-level layers.

Furthermore, we analyze localization performance on CUB test set, as shown in Table 1. Our DGL achieves 43.93%, 49.50%, 39.18%, 38.28% in terms of *Top-1 loc error* on VGGnet, InceptionV3, Resnet50 and Resnet50-SE model, respectively. For CUB-200-2011 dataset, the background of images is simple and one image just have one object. Therefore, lower-level layer can achieve better localization performance, due to lower-level layer tends to highlight the foreground regions. Through extensive experiments, we prove that any convolutional layer of classification model has the localization ability, and some middle layers have better performance.

4.4 Object localization qualitative results

In this subsection, we will report qualitative results of our approach on ILSVRC dataset. Firstly, localization maps produced on different layers will be visualized and analyzed. Secondly, we show the attention maps about Class-aware Enhanced Maps to demonstrate the effectiveness of our approach.

Table 1: localization error and gt-know loc error with different layers.

model	layer	shape	ILSVRC val set			CUB-200-2011 test set		
			top-1 err.	top-5 err.	gt-k err.	top-1 err.	top-5 err.	gt-k err.
VGGnet	conv5_4	$14 \times 14 \times 1024$	55.89	45.37	40.80	58.22	49.05	47.14
	conv5_3	$14 \times 14 \times 512$	54.35	43.48	37.92	61.39	52.50	48.27
	conv5_2	$14 \times 14 \times 512$	**52.34**	**41.11**	**35.22**	52.83	41.84	36.42
	conv5_1	$14 \times 14 \times 512$	52.93	41.75	36.07	48.10	36.11	30.67
	pool4	$14 \times 14 \times 512$	55.49	44.74	39.33	51.07	39.37	33.19
	conv4_3	$28 \times 28 \times 512$	56.12	45.74	40.27	47.93	36.24	30.81
	conv4_2	$28 \times 28 \times 512$	53.97	42.86	37.32	**43.93**	**31.50**	**25.37**
InceptionV3	Mixed_6g	$28 \times 28 \times 1024$	50.72	40.20	36.67	58.23	49.22	47.14
	Mixed_6f	$28 \times 28 \times 1024$	48.79	37.89	33.33	58.82	49.41	44.29
	Mixed_6e	$28 \times 28 \times 768$	**47.77**	**36.62**	**31.92**	54.45	44.17	38.61
	Mixed_6d	$28 \times 28 \times 768$	48.25	37.23	32.68	**49.50**	**37.85**	**32.36**
	Mixed_6c	$28 \times 28 \times 768$	50.67	40.06	35.74	50.57	41.18	32.81
	Mixed_6b	$28 \times 28 \times 768$	51.67	41.17	36.93	52.11	40.97	34.50
Resnet50	group3/block2_shortcut	$28 \times 28 \times 2048$	48.87	40.06	37.23	**39.18**	**29.50**	**25.35**
	group3/block1_shortcut	$28 \times 28 \times 2048$	47.59	38.48	34.81	42.53	33.19	29.08
	group3/block0_shortcut	$28 \times 28 \times 2048$	48.02	38.95	35.37	47.17	38.37	34.45
	group2/block5_shortcut	$28 \times 28 \times 1024$	**46.59**	**37.31**	**33.48**	43.44	34.21	30.00
	group2/block4_shortcut	$28 \times 28 \times 1024$	47.12	37.90	34.00	43.32	34.00	29.89
	group2/block3_shortcut	$28 \times 28 \times 1024$	48.00	38.87	35.14	43.06	33.83	29.88
	group2/block2_shortcut	$28 \times 28 \times 1024$	48.28	39.25	35.53	41.89	32.53	28.29
Resnet50-SE	group3/block2_shortcut	$28 \times 28 \times 2048$	47.14	38.43	35.89	53.31	46.67	45.41
	group3/block1_shortcut	$28 \times 28 \times 2048$	47.17	38.39	35.22	45.63	36.69	33.48
	group3/block0_shortcut	$28 \times 28 \times 2048$	45.80	36.80	33.49	46.84	38.01	34.47
	group2/block5_shortcut	$28 \times 28 \times 1024$	**43.55**	**34.23**	**30.66**	44.75	35.59	31.88
	group2/block4_shortcut	$28 \times 28 \times 1024$	44.05	34.85	31.09	42.58	32.91	29.41
	group2/block3_shortcut	$28 \times 28 \times 1024$	45.37	36.32	32.68	**38.28**	**28.25**	**25.18**
	group2/block2_shortcut	$28 \times 28 \times 1024$	46.15	37.12	33.72	38.73	28.56	25.51

Figure 4 shows the localization maps of *Mixed_6g*, *Mixed_6f*, *Mixed_6e* layers on ILSVRC. It can be observed that attention maps produced by *Mixed_6f*, and *Mixed_6e* layer can highlight almost the entire target object regions, and attention maps produced by *Mixed_6g* layer only discover a part of target object. The *Mixed_6f*, and *Mixed_6e* layer are more suitable to perform localization than *Mixed_6g* layer. It also can be concluded in Table 1.

Furthermore, Figure 5 shows the enhanced maps. For intuitive visualization, enhanced maps and gradients of target class multiply -1. It has no effect on producing localization map in Equation 8. Compared with feature maps S, enhanced maps A discovers more target object regions, which means that the target objects could be better represented. This certifies the effectiveness of our thought that pulls feature maps to the class center in the feature space. Attention map $\overline{M}$ is the final localization map of our DGL. On first and second rows, when M only highlights the parts of object, $\overline{M}$ discovers the entire object regions. On third row, when M highlights the whole object regions, $\overline{M}$ doesn't discover other image regions. In particular, on second row, even image has complex background, $\overline{M}$ still only highlights the target object regions.

4.5 Comparison with the state-of-the-arts

We compare the proposed DGL with the-state-of-arts approaches on CUB-200-2011 test set and ILSVRC val set. Table 2 summarizes the localization performance of both various baselines and our DGL on ILSVRC val set. It can be observed that our DGL is valid and outperforms all of baselines on four models in terms of *gt-know loc error*. Furthermore, we obtain best results in term of *Top-1 loc error* and *Top-5 loc error* expect for VGGnet. This is because we keep training configurations of VGGnet in line with other models, resulting in the reduced classification performance.

We adopt SPG [35] and SPG-plain [35] as baseline, our method outperforms SPG [35] and SPG-plain [35] by 3.63% and 5.94% in terms of *Top-1 loc error* on InceptionV3, respectively. Our approach

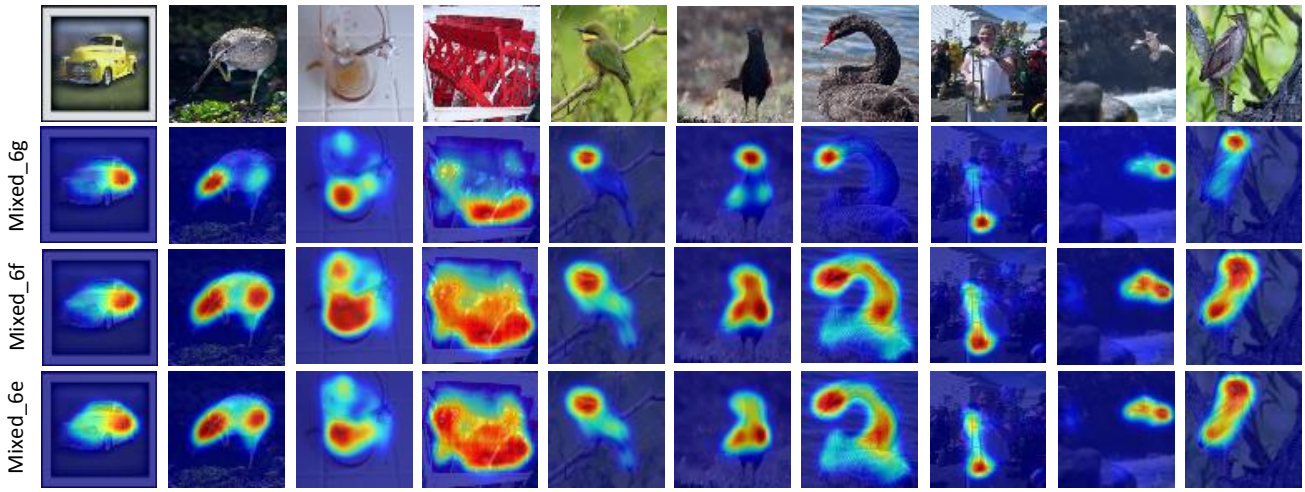

Figure 4: Examples of attention maps produced on *Mixed_6g*, *Mixed_6f*, and *Mixed_6e* layer of InceptionV3 model [21] with our DGL.

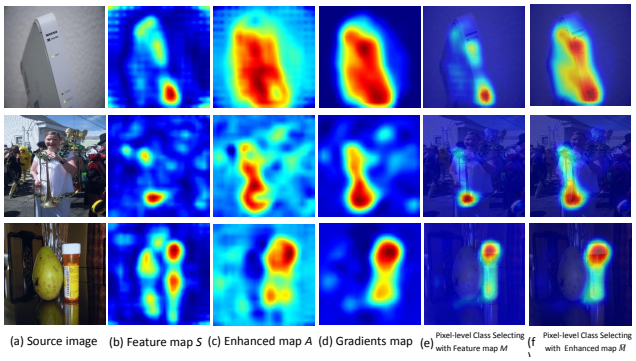

Figure 5: (a) Source image. (b) Feature maps S. (c) Enhanced maps A times -1, produced from Class-aware Enhanced Maps branch. (d) Gradients of target class times -1. (e) Localization map M produced by Pixel-level Class Selection with (b). (f) Localization map $\overline{M}$ produced by Pixel-level Class Selection with (c).

achieves 31.92%, outperforms SPG[35] by 3.39% in term of *gt-know loc error*. In addition, to quantitatively show the time cost, we compare DGL with SPG-plain on a single 2080Ti GPU. DGL takes 0.1514s per image using the *Mixed_6e* layer, and SPG-plain takes 0.1223s per image. Although the DGL is a little more complex than SPG-plain, it is tolerable.

We show the results on CUB-200-2011 in Table 3. Our DGL obtains 49.50% and outperforms SPG [35] and SPG-plain [35] by 3.86% and 6.83% in terms of *Top-1 loc error* on inceptionV3, respectively. Note that ADL [2] central crops the input images to 224×224 after reshaping to 256×256 during testing. For a fair comparison, we adopt same process to compare with ADL [2].Our DGL has a boost to 38.73% of *Top-1 loc error* on Resnet50-SE. In summary, our approach has superior efficiency and fewer parameters than ACol [34] and SPG [35], and achieves the new state-of-the-art results on ILSVRC and CUB-200-2011 dataset.

4.6 Ablation study

We employ the VGGnet and InceptionV3 models pre-trained for DGL as backbones, and conduct ablation analyses of our Dual-Gradients Framework by removing the Class-aware Enhanced Maps and Pixel-level Class Selection, respectively.

Table 2: Localization error on the ILSVRC validation set.

Method	top-1 err	top-5 err.	gt-k err.
CAM-VGG-GAP [36]	57.20*	45.14*	41.00 *
ACoL-VGG [34]	54.17*	**40.57***	37.04*
CutMix-VGG-GAP [31]	56.55*	-	-
ADL-VGG-GAP [2]	55.08*	-	-
CCAM-VGG-GAP [29]	**51.78***	40.64*	36.42*
DGL-VGG-GAP	52.34	41.11	**35.22**
CAM-GooLeNet-GAP [36]	56.40*	43.00*	41.34*
Has-GoogLeNet-32 [20]	54.79*	-	39.43*
SPG-plain [35]	53.71*	41.81*	37.32*
DANet-inceptionV3 [28]	52.47*	40.00*	-
SPG [35]	51.40*	40.00*	35.31*
ADL-InceptionV3 [2]	51.29*	-	-
DGL-InceptionV3	**47.77**	**36.62**	**31.92**
CAM-Resnet50-SE [36]	47.14	38.43	35.89
ADL-Resnet50-SE [2]	51.47*	-	-
DGL-Resnet50-SE	**43.55**	**34.23**	**30.66**
CAM-Resnet50 [36]	48.87	40.06	37.23
CutMix-Resnet50 [31]	52.75*	-	-
DGL-Resnet50	**46.59**	**37.31**	**33.48**

* denotes results reported in the original papers.

Effect of the Class-aware Enhanced Maps One of our main contributions is the proposed Class-aware Enhanced Maps for mining entire object regions. For investigating the effect of Class-aware Enhanced Maps, we test the localization performance by replacing enhanced maps with feature maps. Table 4 shows the localization results on ILSVRC val set. We can observe that three metrics of all layers increase without Class-aware Enhanced Maps except for *Mixed_6g* layer. On *Mixed_6g* layer, performance with Class-aware Enhanced Maps is same to results without Class-aware Enhanced Maps. We believe this is because gradients of loss function with

Table 3: Loc errors on the CUB-200-2011 test set.

Method	top-1 err.	top-5 err.
CAM-Googlenet-GAP [36]	59.00*	-
SPG-plain [35]	56.33*	46.47*
SPG [35]	53.36*	42.28*
DANet-Googlenet [28]	50.55*	39.54*
DGL-inceptionV3	**49.50**	**37.85**
ACoL-VGG-GAP [34]	54.08*	43.49*
DANet-VGG-GAP [28]	47.48*	38.04*
CutMix-VGG-GAP [31]	47.47*	-
DGL-VGG-GAP	**43.93**	**31.50**
CAM-Resnet50 [36]	53.09	46.43
CutMix-ResNet50 [31]	45.19*	-
DGL-ResNet50	**39.18**	**29.50**
CAM-Resnet50-SE [36]	53.31	46.67
DGL-Resnet50-SE	**38.73**	**28.56**
ADL-inceptionV3 [2]	46.86*	-
DGL-inceptionV3**	**46.55**	**34.21**
ADL-VGG-GAP [2]	47.64*	-
DGL-VGG-GAP**	**40.28**	**26.55**
ADL-Resnet50-SE[2]	37.71*	-
DGL-Resnet50-SE**	**36.12**	**26.10**

* denotes results reported in the original papers.
** indicates approach using image process of ADL.

regard to *Mixed_6g* layer are channel-level, and have less influence to feature maps in Equation 7. Specifically, without the Class-aware Enhanced Maps, *Top-1 loc error* of *Mixed_6e* increase 6.54% on InceptionV3 model.In general, it proves that Class-aware Enhanced Maps indeed catch the more object regions on some middle convolutional layers.

Effect of the Pixel-level Class Selection Next, we replace Pixel-level Class Selection to grad-cam to investigate its effect. In other word, we sum the gradients of target class on channels, and employ a weighted sum of feature maps to generate localization map. Note that approach with grad-cam is equal to CAM due to GAP layer, when applied on the last convolutional layer. We still perform localization on different layers of VGGnet and InceptionV3 model, as shown in Table 5. Localization performance on *conv5_3*, *conv5_2*, *conv5_1*, *pool4*, *mixed_6f*, *mixed_6e* is decrease, compared with Table 1 and Table 4. On *conv4_3*, *conv4_2*, *mixed_6d*, *mixed_6c*, *mixed_6b* layer, the performance with grad-cam shown in Table 5 is similar to results with Pixel-level Class Selection shown in Table 4. We believe this is because the lower-level feature maps include general features unrelated to the target class, resulting in the localization map only discovers the foreground of images. To sum up, it reveals that the proposed Pixel-level Class Selection is effective to localize more preciser localization map on some middle layers of classification model.

5 CONCLUSIONS AND FUTURE WORK

In this work, we propose a simple but effective WSOL framework, *i.e.* Dual-Gradients Localization framework to mine the entire object regions on any convolutional layer only given image-level annotations. Our framework leverages two kinds of gradients to achieve localization. Firstly, we provide a new view to explain CAM by using gradients of target class, and further propose Pixel-level Class Selection to perform localization on any convolutional layer. Next, we propose Class-aware Enhanced Maps to explore the whole object regions by utilizing gradients of classification loss function. The proposed approach can effectively identify entire target object regions on some middle layers. In addition, it is easy to embedded into any weakly supervised approach to improve performance.

Table 4: Ablation study on effect of Class-aware Enhanced Maps.

model	layer	Loc top1	Loc top5	gt-know
VGGnet	conv5_4	55.89	45.37	40.80
	conv5_3	55.80	45.24	40.94
	conv5_2	60.82	51.48	47.38
	conv5_1	68.19	60.23	56.76
	pool4	68.53	60.57	56.99
	conv4_3	70.25	62.31	59.01
	conv4_2	70.46	62.57	59.21
InceptionV3	Mixed_6g	50.72	40.20	36.67
	Mixed_6f	51.11	40.74	37.28
	Mixed_6e	53.83	44.02	40.48
	Mixed_6d	68.35	61.07	58.25
	Mixed_6c	69.17	61.92	59.28
	Mixed_6b	69.16	61.91	59.28

Table 5: Ablation study on effect of Pixel-level Class Selection.

model	layer	Loc top1	Loc top5	gt-know
VGGnet	conv5_4	55.89	45.37	40.80
	conv5_3	60.63	51.03	47.07
	conv5_2	62.08	52.88	48.86
	conv5_1	69.89	61.90	58.56
	pool4	70.49	62.54	59.25
	conv4_3	70.52	62.56	59.29
	conv4_2	70.50	62.54	59.27
InceptionV3	Mixed_6g	50.72	40.20	36.67
	Mixed_6f	63.80	55.76	52.54
	Mixed_6e	62.81	54.54	51.26
	Mixed_6d	65.85	58.19	55.30
	Mixed_6c	68.90	61.66	59.06
	Mixed_6b	69.11	61.86	59.25

ACKNOWLEDGMENTS

This work is supported in part by National Key Research and Development of China (2017YFC1703503, 2018AAA0102100) and in part by National Natural Science Foundation of China (61972022, 61532005, U1936212).

REFERENCES

[1] Jiwoon Ahn and Suha Kwak. 2018. Learning pixel-level semantic affinity with image-level supervision for weakly supervised semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 4981–4990.

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

[3] Xuanyi Dong, Liang Zheng, Fan Ma, Yi Yang, and Deyu Meng. 2018. Few-example object detection with model communication. *IEEE transactions on pattern analysis and machine intelligence* 41, 7 (2018), 1641–1654.

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

[5] Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7132–7141.

[6] Zilong Huang, Xinggang Wang, Jiasi Wang, Wenyu Liu, and Jingdong Wang. 2018. Weakly-supervised semantic segmentation network with deep seeded region growing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7014–7023.

[7] Huaizu Jiang, Jingdong Wang, Zejian Yuan, Yang Wu, Nanning Zheng, and Shipeng Li. 2013. Salient object detection: A discriminative regional feature integration approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2083–2090.

[8] Peng-Tao Jiang, Qibin Hou, Yang Cao, Ming-Ming Cheng, Yunchao Wei, and Hong-Kai Xiong. 2019. Integral Object Mining via Online Attention Accumulation. In *Proceedings of the IEEE International Conference on Computer Vision*. 2070–2079.

[9] Zequn Jie, Yunchao Wei, Xiaojie Jin, Jiashi Feng, and Wei Liu. 2017. Deep self-taught learning for weakly supervised object localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 1377–1385.

[10] Xiaoyan Li, Meina Kan, Shiguang Shan, and Xilin Chen. 2019. Weakly Supervised Object Detection With Segmentation Collaboration. In *Proceedings of the IEEE International Conference on Computer Vision*. 9735–9744.

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

[12] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. 2016. Ssd: Single shot multibox detector. In *European conference on computer vision*. Springer, 21–37.

[13] Seong Joon Oh, Rodrigo Benenson, Anna Khoreva, Zeynep Akata, Mario Fritz, and Bernt Schiele. 2017. Exploiting saliency for object segmentation from image level labels. In *2017 IEEE conference on computer vision and pattern recognition (CVPR)*. IEEE, 5038–5047.

[14] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 779–788.

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

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

[17] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*. 618–626.

[18] Pierre Sermanet, David Eigen, Xiang Zhang, Michaël Mathieu, Rob Fergus, and Yann LeCun. 2013. Overfeat: Integrated recognition, localization and detection using convolutional networks. *arXiv preprint arXiv:1312.6229* (2013).

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

[20] Krishna Kumar Singh and Yong Jae Lee. 2017. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *2017 IEEE international conference on computer vision (ICCV)*. IEEE, 3544–3553.

[21] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2818–2826.

[22] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. 2011. The caltech-ucsd birds-200-2011 dataset. (2011).

[23] Fang Wan, Chang Liu, Wei Ke, Xiangyang Ji, Jianbin Jiao, and Qixiang Ye. 2019. C-MIL: Continuation multiple instance learning for weakly supervised object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2199–2208.

[24] Yunchao Wei, Jiashi Feng, Xiaodan Liang, Ming-Ming Cheng, Yao Zhao, and Shuicheng Yan. 2017. Object region mining with adversarial erasing: A simple classification to semantic segmentation approach. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1568–1576.

[25] Yunchao Wei, Xiaodan Liang, Yunpeng Chen, Xiaohui Shen, Ming-Ming Cheng, Jiashi Feng, Yao Zhao, and Shuicheng Yan. 2016. Stc: A simple to complex framework for weakly-supervised semantic segmentation. *IEEE transactions on pattern analysis and machine intelligence* 39, 11 (2016), 2314–2320.

[26] Yunchao Wei, Huaxin Xiao, Honghui Shi, Zequn Jie, Jiashi Feng, and Thomas S Huang. 2018. Revisiting dilated convolution: A simple approach for weakly-and semi-supervised semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 7268–7277.

[27] Yuxin Wu et al. 2016. Tensorpack. https://github.com/tensorpack/.

[28] Haolan Xue, Chang Liu, Fang Wan, Jianbin Jiao, Xiangyang Ji, and Qixiang Ye. 2019. DANet: Divergent Activation for Weakly Supervised Object Localization. In *Proceedings of the IEEE International Conference on Computer Vision*. 6589–6598.

[29] Seunghan Yang, Yoonhyung Kim, Youngeun Kim, and Changick Kim. 2020. Combinational Class Activation Maps for Weakly Supervised Object Localization. In *The IEEE Winter Conference on Applications of Computer Vision*. 2941–2949.

[30] Zhenheng Yang, Dhruv Mahajan, Deepti Ghadiyaram, Ram Nevatia, and Vignesh Ramanathan. 2019. Activity driven weakly supervised object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2917–2926.

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

[32] Bingfeng Zhang, Jimin Xiao, Yunchao Wei, Mingjie Sun, and Kaizhu Huang. 2019. Reliability Does Matter: An End-to-End Weakly Supervised Semantic Segmentation Approach. *arXiv preprint arXiv:1911.08039* (2019).

[33] Jianming Zhang, Sarah Adel Bargal, Zhe Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. 2018. Top-down neural attention by excitation backprop. *International Journal of Computer Vision* 126, 10 (2018), 1084–1102.

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

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

[36] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. 2016. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2921–2929.

[37] Yi Zhu, Yanzhao Zhou, Qixiang Ye, Qiang Qiu, and Jianbin Jiao. 2017. Soft proposal networks for weakly supervised object localization. In *Proceedings of the IEEE International Conference on Computer Vision*. 1841–1850.