Improve CAM with Auto-adapted Segmentation and Co-supervised Augmentation

Ziyi Kou*†
University of Notre Dame
zkou@nd.edu

Guofeng Cui*†
Rutgers University
gc669@cs.rutgers.edu

Shaojie Wang*
Washington University in St. Louis
joss@wustl.edu

Wentian Zhao*
Adobe
wezhao@adobe.com

Chenliang Xu
University of Rochester
chenliang.xu@rochester.edu

Abstract

Weakly Supervised Object Localization (WSOL) methods generate both classification and localization results by learning from only image category labels. Previous methods usually utilize class activation map (CAM) to obtain target object regions. However, most of them only focus on improving foreground object parts in CAM, but ignore the important effect of its background contents. In this paper, we propose a confidence segmentation (ConfSeg) module that builds confidence score for each pixel in CAM without introducing additional hyper-parameters. The generated sample-specific confidence mask is able to indicate the extent of determination for each pixel in CAM, and further supervises additional CAM extended from internal feature maps. Besides, we introduce Co-supervised Augmentation (CoAug) module to capture feature-level representation for foreground and background parts in CAM separately. Then a metric loss is applied at batch sample level to augment distinguish ability of our model, which helps a lot to localize more related object parts. Our final model, CSoA, combines the two modules and achieves superior performance, e.g. 37.69% *and* 48.81% *Top-1 localization error on CUB-200 and ILSVRC datasets, respectively, which outperforms all previous methods and becomes the new state-of-the-art.*

1. Introduction

Weakly-Supervised Object Localization (WSOL) aims to learn object locations in a given image from only image-level labels. It avoids expensive bounding box annotations and thus dramatically reduces the cost of human labors in image annotations. To tackle the problem, utilizing class activation map (CAM) is often adopted as a good choice recently. CAM is a type of 3D feature map with each channel corresponding to one category label. The pixels in it can indicate the discriminative regions of objects belonging to each category. Therefore, by extracting the features via the label index, the model can roughly locate the position of target objects. The main reason for the wide use of CAM is that the generation of CAM needs only little modifications based on classical CNN backbones but the performance is robust. For instance, Zhou et al. [26] propose to replace fully connected layer with global average pooling layer (GAP) to generate CAM for a given image, which achieves a competitive localization result.

Though using CAM for localization is efficient and straightforward, it can only detect some parts of the objects instead of covering the full object extents. The main reason is that traditional classification networks tend to distinguish images by focusing on the most representative regions, which can minimize the classification loss but results in losing other related but non-essential parts. To address the problem, lots of approaches [10, 24, 25, 4, 22, 23] have been proposed, and they can be categorized roughly into the following two classes. The first class of methods [10, 4] manipulates input data samples or internal feature maps directly to enforce the network to search related object parts. It improves localization but sacrifices classification performance because the target objects may become unrecognized after their parts are erased. The second type of methods [24, 22, 23] generate multiple CAMs and combine them for the final localization. Their CAMs are useful as they contain information from different convolutional layers or different levels of semantics.

However, all the above methods only focus on expanding foreground object regions and ignore background parts in CAM. In our observation, determining background not only

*This work was done while they were at University of Rochester.
†Both authors contributed equally to this work.

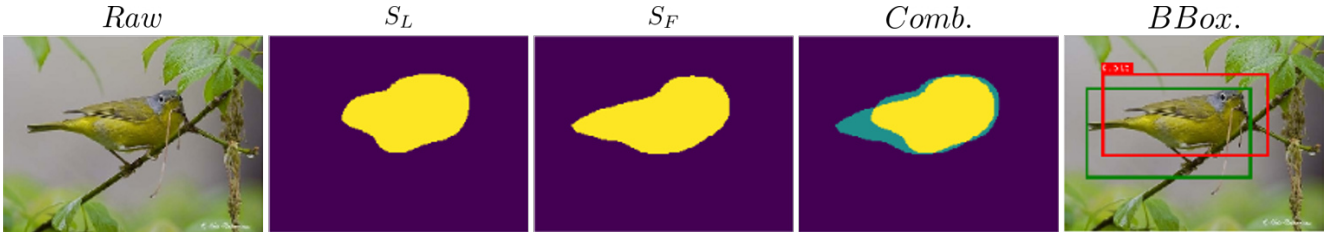


Figure 1. Binary localization maps from two CAMs and the final localization result. S_L is from CAM at the top layer while S_F from another one. The combination of two maps and the final bounding box demonstrates the advantage of our network to produce more complete and complementary results. Red and green bounding boxes denotes predicted localization result can ground-truth label.

helps remove unrelated pixels but also plays as additional supervisions when multiple CAMs are being used. Indeed, to the best of our knowledge, [25] is the only work that considers segmenting background contents of CAM based on internal feature maps. However, it has to set fixed segmentation thresholds for all samples in a one-size-fit-all manner, which is not optimal. Besides, the background segmentation in [25] is only used to regulate a single CAM during training and discarded in the inference time. Therefore, such approach is not an ideal way to generate and utilize background regions to improve localization performance.

Though the CAM can be self-refined by internal feature maps as discussed above, no additional regularization for generated CAM is provided in previous methods. Supervised by only category-level labels, CAM becomes unstable for localization. For instance, [21] indicates for samples belonging to the same category, the model will focus on different object parts due to the various characteristics the target object displays. However, such phenomenon is not expected in our localization task since we prefer the complete prediction of objects for each sample.

To overcome the above limitations, we propose a new framework named CSoA for the WSOL task with two novel modules. We first introduce the confidence segmentation (ConfSeg) module, an internal module that connects and refines two different CAMs inside our network. One of the two CAMs is generated from the top convolutional layer and thus captures high-level semantic information. Another CAM is extended from internal feature maps of the backbone network and contains fine-level object boundary clues. These two CAMs have totally different characteristics and receptive fields but both contribute to the final localization and classification performance. With these two different CAMs, the ConfSeg module segments a dynamic per-sample confidence mask from the first CAM and applies it as additional supervisions to regulate the second one, which finally encourages them to be incorporated together with both high-level and fine-level information. Fig. 1 shows the final binary localization maps extracted from the two CAMs and their combination. Especially, the generated maps concentrate on foreground object parts with similar center area but become complementary for surrounded regions. Without introducing additional hyper-parameters, the ConfSeg module greatly improves the final localization performance compared with the current state-of-the-art results.

Apart from the ConfSeg module, we propose a metric-based approach denoted as Co-supervised Augmentation (CoAug) regularizer to further regulate CAM and augment its integrity. For CoAug, two expectations for CAM are considered. The first one is that an ideal CAM should separate the image into two regions with non-intersected contents, specifically foreground objects and background. The second one is aligned according to foreground regions focused by CAM of different samples. For images belonging to the same category, their foreground parts are supposed to share similar identifications. While for different categories, the samples should be distinct with each other. With the above two assumptions, we construct the CoAug module that enforces batch-level samples to inter-supervise collaboratively by embedded vectors that are also applied in [7, 8]. By this way, the CoAug module enhances the recognition ability of CAM by not only gathering the information of each category from various samples, but also discriminate them. The details of the module will be discussed in Section 3 and its advantage will be demonstrated in Section 4.

In summary, our main contributions are three folds: (1) We propose a novel confidence segmentation module to generate a confidence mask that gets two different CAMs interacted and refined without additional hyper parameters. (2) We propose Co-supervised Augmentation module to refine CAM by regulating feature-level representations, which guides our model to localize more related object regions. (3) With only image-level supervision for training, our method greatly outperforms other state-of-the-art methods on two standard benchmarks, ILSVRC validation set and CUB-200-2011 test set, for weakly supervised localization performance.

2. Related Work

Weakly Supervised Object Localization usually relies on CAM to localize objects. Zhou et al. [26] propose Global Average Pooling (GAP) layer for deep neural networks to generate CAM for localization. Based on it, Zhang et

al. [24] prove that the process for obtaining CAM can be end-to-end and further propose ACoL network that adopts cut-and-search strategy on the feature maps. Moreover, Zhang et al. [25] propose SPG network that extends pixel-level mask from internal feature maps and complement CAM in the final. Recently, Choe et al. [4] design a general dropout algorithm for internal feature maps to refine CAM from bottom level. Besides, the clustering of ground-truth labels is proposed in [22] to obtain multiple semantic level CAM.

Similar to the WSOL problem, the methods for Object Co-Segmentation task attempts to segment target objects based on image-level labels. However, as introduced by Rother et al [13], co-segmentation task aims to segment common objects from a set of images belonging to a specific category instead of multiple ones. The main idea for tackling the problem is to leverage intra-image discovery and inter-sample correlation [7, 2, 11, 12, 19]. For example, Li et al. [12] embed image features by Siamese Encoder and then apply feature matching to extract common objects from image pairs. Hsu et al. [7] introduce co-attention loss based on intra- and inter-sample comparison to guide the object discovery process. Besides, they utilize unsupervised methods to pick object proposals in order to refine generated segmentation maps. Although our CoAug module shares similar idea with co-segmentation methods, it aims to localize objects from images under the multi-category condition and further alleviates discriminative regions biased problem [7].

There are some other methods related to model interpretability but can also be applied to localization tasks. GradCAM [15] combines gradient values and internal feature maps to produce CAM without adding additional pooling layers. Chattopadhyay et al. [1] further improve [15] by using a weighted combination of the positive partial derivatives of the feature maps in the last convolutional layer. These methods are usually engaged to propose new CAM that can interpret internal functions of various neural networks. However, in this work, we focus on improving localization performance of CAM, which is a totally different purpose. Although we utilize original CAM [26] in our method, the proposed modules can also be applied to different kinds of CAM as long as they share similar characteristics with the original one, i.e. highlighting target object parts as localization clues.

3. Method

In this section, we first review the seminal Class Activation Map (CAM) [26], then introduce our Confidence Segmentation (ConfSeg) module along with the Co-supervised Augmentation (CoAug) regularizer. An overview of our proposed method for the training phase is shown in Fig. 2.

We first describe the weakly supervised object localization problem and the basic network proposed in [24] for generating CAM. Given a set of N images, $\{I_n\}_{n=1}^{N}$ with C categories, each image contains objects for only one category. Our goal is to classify each image and locate the corresponding objects with bounding boxes. In [24], a Fully Convolutional Network (FCN) is proposed with a backbone F consisting of L layers, and a classifier $\mathcal{W}_{\mathcal{L}}$. For an input image, the backbone network produces the feature map $M_l \in \mathbb{R}^{K_l \times H_l \times W_l}$ after layer l with K_l channels. We denote $M_L \in \mathbb{R}^{K_L \times H_L \times W_L}$ as the last feature map from F. To generate CAM, the classifier $\mathcal{W}_{\mathcal{L}}$ usually contains several convolutional layers to convert the number of channels from K_L to C, i.e. the number of categories. Following that, a Global Average Pooling (GAP) layer is applied at each channel of M_L to generate the class logit $\mathbf{y_L} = \{y_L^c\}_{c=1}^{C}$, which is then feed into the classification loss calculation. This process can be written as:

$$S_L = \mathcal{W}_{\mathcal{L}}(M_L)\ ,\ \ y_L^c = \frac{\sum_{i,j}(S_L^c)_{i,j}}{H_L \times W_L}\ ,\ \ \forall\, c \in \{1, \ldots, C\}\ , \tag{1}$$

where $(S_L^c)_{i,j}$ refers to a certain pixel on the c-th channel of the feature map S_L. After training, the feature map S_L^c corresponding to the predicted category is extracted. Then the largest connected region with positive values is segmented and finally processed to the bounding box prediction.

3.1. Confidence Segmentation Module

Though the basic framework is straightforward and efficient, it can only capture the most discriminative part of target objects. To address the problem, we propose the confidence segmentation (ConfSeg) module to generate a sample-specific confidence mask for CAM. The mask contains confidence score for each pixel and can segment regions with high confidence scores, including both foreground and background parts from CAM. With a high precision, the mask can serve as additional supervisions to guide other object detectors, encouraging them to explore more object-related regions.

To create another object detector that gets supervised, we extend one more CAM from internal feature maps inside the backbone network F, which can be denoted as S_F. Concretely, we first create a new classifier $\mathcal{W}_{\mathcal{F}}$ that has the same structure with $\mathcal{W}_{\mathcal{L}}$ and also goes through GAP layer to generate logits $\mathbf{y_F}$. Different from [22] that builds several CAMs in multiple semantic levels, $\mathcal{W}_{\mathcal{L}}$ and $\mathcal{W}_{\mathcal{F}}$ share same categories for classification and have the same spatial size. As illustrated in [18], CNNs trained for object recognition have low-level vision features in early convolutional layers while more semantic features in top layers. Therefore, though $\mathcal{W}_{\mathcal{F}}$ can be appended after any convolutional layer, it needs to localize objects precisely as well as achieving

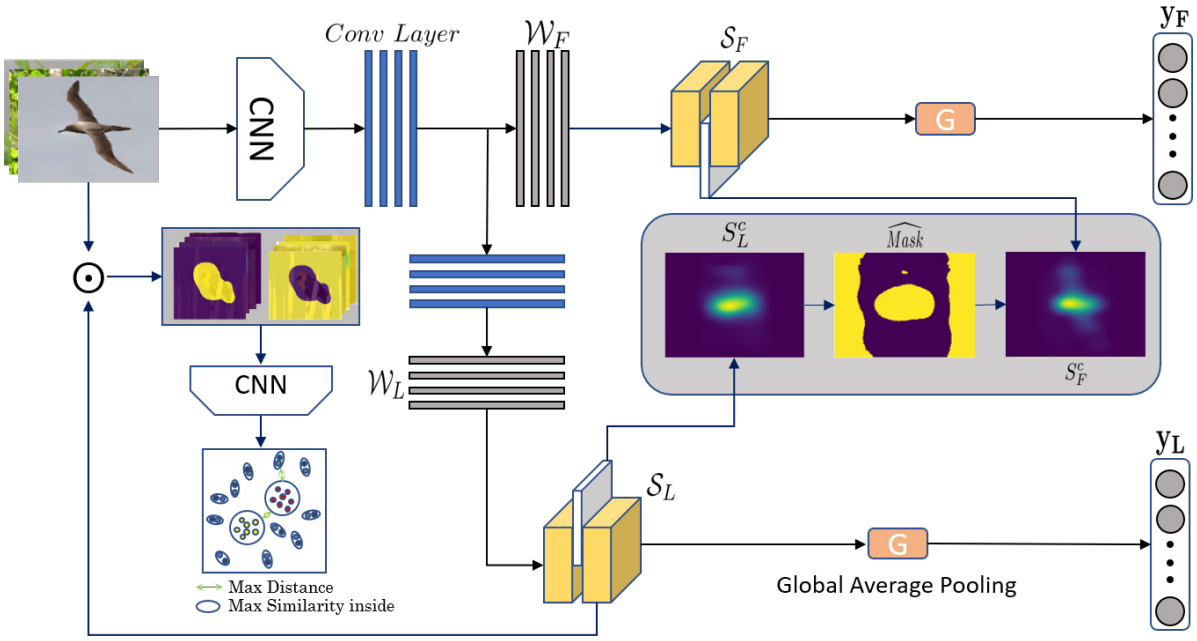


Figure 2. The overall structure of our CSoA network. For each input image, two different CAMs, S_L and S_F, are generated from different classifiers and processed to logits for classification. Besides, the c-th slice of S_L is then extracted and transformed to confidence segmentation mask by the ConfSeg module. The mask serves as additional supervisions by controlling the distance between S_L^c and S_F^c. For samples in the same batch, they are first combined with foreground and background parts of S_L^c separately. Finally all weighted samples are represented as 1-D vectors and play a metric-based learning process.

reasonable recognition performance. We will discuss the exact position for it in Section 4.

With two different CAMs generated, the ConfSeg module connects them together. We first extract one slice from the feature map S_L, denoted as S_L^c according to the ground truth index, or the predicted one in the inference time. Then we calculate the mean value of S_L^c, which is denoted as μ_1. If the value of a pixel in S_L^c is close to μ_1, that means the corresponding position is ambiguous to be determined. In contrast, if a pixel has much larger or smaller value compared with μ_1, it is very likely to be located on the target objects or background parts. Therefore, we can generate a confidence mask with each element calculated as the distance between each pixel and μ_1. The process can be denoted as:

$$Mask_{i,j} = \left|(S_L^c)_{i,j} - \mu_1\right|, \text{ where } \mu_1 = \frac{\sum_{i,j}(S_L^c)_{i,j}}{H_L \times W_L}. \tag{2}$$

After determining the confidence score for each pixel on S_L^c, the regions with high confidence are segmented from the mask by Eq. 3. Instead of setting a fixed threshold for segmenting all image samples, we use a sample-dynamic threshold, denoted as μ_2, for each image by taking the mean value of the mask. Therefore, the threshold for each sample is adaptively computed based on the corresponding confidence mask. If the score for one pixel is higher than μ_2, we conclude that the pixel is very likely to have the correct value and vice versa. The equation can be formulated as:

$$\widehat{Mask}_{i,j} = \begin{cases} 1 & Mask_{i,j} > \mu_2 \\ 0 & \text{otherwise} \end{cases}, \tag{3}$$

$$\text{where } \mu_2 = \frac{\sum_{i,j} Mask_{i,j}}{H_L \times W_L}. \tag{4}$$

With S_L^c and the generated binary confidence mask $\widehat{Mask}$, we create a new supervision for S_F^c by controlling the distance between each pixel in S_L^c and S_F^c. For the positions that their corresponding values are 1 in $\widehat{Mask}$, we encourage S_F^c to be similar with S_L^c, which means S_F^c should follow the decisions made by S_L^c if they are confident enough. For other positions, we allow S_F^c to be different from S_L^c so that it can refine the object boundaries when the decisions are made with low confidence. With such an adversarial strategy, we do not need to worry if additional explorations by S_F^c for object related regions may reach background parts because the confidence mask sets solid restriction for the background part in CAM. We formulate the process as the following loss:

$$\mathcal{L}_{inner} = \sum_{i,j} |(S_F^c)_{i,j} - (S_L^c)_{i,j}| \odot \widehat{Mask}_{i,j}. \tag{5}$$

Finally, the total loss function with the ConfSeg module is:

$$\mathcal{L}_{\mathcal{C}} = \mathcal{L}_{cls} + \alpha \mathcal{L}_{inner}, \tag{6}$$

where α is a factor ranging within $[0, 1]$ that increases along the training epoch to avoid unstable prediction from S_L at the initial training process. $\mathcal{L}_{cls}$ denotes the cross entropy loss for both $\mathbf{y_L}$ and $\mathbf{y_F}$ with same ground-truth categories.

The relation to Zhang et al. [25]. By further formulating our proposed ConfSeg module, we show that it is a generalized version of Zhang et al. [25]. The latter sets prefixed thresholds of foreground and background for all image samples before the training process. Our method can also represent their thresholds through simple transformation, which can be denoted as:

$$\begin{aligned} \xi_1 &= \mu_1 + \mu_2 \ , \\ \xi_2 &= \mu_1 - \mu_2 \ , \end{aligned} \tag{7}$$

where ξ_1 and ξ_2 are thresholds for foreground and background, respectively. Therefore, the ConfSeg module can achieve the same function as [25] but is versatile with sample-level adaption for the segmentation of CAM without introducing additional parameters.

3.2. Co-supervised Augmentation Module

For the fully supervised localization task, the ground-truth box annotations are always utilized to guide the generation of object proposals. However, in the setting of WSOL task, only image-level supervisions are available, which leads to severe bias of recognition models that tends to localize the most discriminative region rather than entire objects. To address the problem, we further introduce a plug-in metric-based module to regulate CAM with feature-level supervisions, since the comparison between different samples is capable of preserving more visual features.

Our approach is inspired by metric learning methods [8] that embed images into representation vectors and leverage distance as metrics to estimate their correlations. Therefore, in CoAug module, we consider two kinds of relationships: 1) foreground and background part that should both represent distinct features; 2) foreground objects of different samples in the batch level.

Before discussing the details about metric-based processes, we first segment out predicted foreground and background regions of input images according to generated CAM. For the reason that CAM is able to highlight foreground object region of target category, we multiply the slide of CAM according to ground-truth index with raw input images to represent corresponding object and then embed the object into feature vector F_m. Similarly, we can also obtain background vector B_m denoted as:

$$\begin{aligned} F_m &= \mathcal{W}_{\mathcal{E}}(S_l^{c_m} \odot I_m) \ , \\ B_m &= \mathcal{W}_{\mathcal{E}}((1 - S_l^{c_m}) \odot I_m) \ , \end{aligned} \tag{8}$$

where $\mathcal{W}_{\mathcal{E}}$ indicates the embedding network, $\odot$ represents pixel-wise multiplication, $S_l^{c_m}$ refers to the localization map from the c_m channel of S_l, and c_m is the c-th category of the m-th image. Please note that I_m can be either the intermediate feature map generated by a CNN or directly the raw m-th image.

Then the Relation.1 can be measured as the distance between F_m and B_m as :

$$D_m^{cam} = \|F_m - B_m\|_2 \ . \tag{9}$$

in which we expect D_m^{cam} to be large. Additionally, we utilize the background part of the confidence mask introduced in the previous section to augment the ability of CAM to avoid mis-classifying foreground part as Eq. 10:

$$\begin{aligned} D_m^{back} &= \|B_m - Mask_B^m\|_2 \ , \\ \text{where } Mask_B^m &= \mathcal{W}_{\mathcal{E}}((1 - S_l^{C_m}) \odot \widehat{Mask} \odot I_m) \ . \end{aligned} \tag{10}$$

Specifically, we obtain $Mask_B^m$ by first multiplying $\widehat{Mask}$ with I_m to extract confident parts in the image, and then incorporating it with $1 - S_l^{c_m}$ to represent the background content.

Apart from comparing foreground and background regions of a single image, we also consider the relationship among multiple samples, denoted as Relation.2 above. We calculate the distance between foreground vectors of different input samples as:

$$D_{m,n} = \|F_m - F_n\|_2 \, . \tag{11}$$

When F_m and F_n belong to the same category, they are supposed to share similar representations and $D_{m,n}$ should be small. In this case we change $D_{m,n}$ to $D_{m,n}^{same}$. For other cases, where the categories of F_m and F_n are different, we convert $D_{m,n}$ to $D_{m,n}^{diff}$ and expect it to be large.

With the definition of the four distances for regulation, we define the following loss function to have images supervising each other:

$$\begin{aligned} \mathcal{L}_D^{same} &= \sum_{\{m,n|cls(m)\neq cls(n)\}} \frac{\gamma \cdot (D_m^{back} + D_n^{back})}{\delta \cdot D_{m,n}^{diff} + \frac{1}{2}(D_m^{cam} + D_n^{cam})}, \\ \mathcal{L}_D^{diff} &= \sum_{\{m,n|cls(m)=cls(n)\}} \frac{\gamma \cdot (D_m^{back} + D_n^{back}) + D_{m,n}^{same}}{\frac{1}{2}(D_m^{cam} + D_n^{cam})}, \\ \mathcal{L}_D &= \mathcal{L}_D^{same} + \mathcal{L}_D^{diff}, \end{aligned} \tag{12}$$

where γ and δ in the equation are two factors that controls the global scale of $\mathcal{L}_{\mathcal{D}}$, while $cls(\cdot)$ refers to category.

For the training time, we combine $\mathcal{L}_{\mathcal{C}}$ and $\mathcal{L}_{\mathcal{D}}$ together. During the inference time, we remove both ConfSeg and CoAug module and only keep the two generated CAMs. We first segment the target object parts following the instruction in [26]. In details, we extract max values S_F^{max}

and S_L^{max} from S_F and S_L respectively. Then we create binary localization maps by Eq. 13 denoted as:

$$\widehat{S_{F/L}} = \begin{cases} 1 & S_{F/L}^{i,j} > \theta \cdot S_{F/L}^{max} , \\ 0 & \text{otherwise} , \end{cases} \tag{13}$$

where θ is a pre-defined parameter for segmentation. Finally, we combine the two localization maps with the pixel value as 1 if either pixel value on $\widehat{S_F}$ or $\widehat{S_L}$ is 1. Otherwise, the pixel value is set to 0 since neither of two CAMs consider it belonging to foreground object parts.

4. Experiment

4.1. Implementation Details

Following the configuration of previous methods [25, 22], our proposed modules are integrated with the commonly used CNNs including VGGnet [16] and GoogLeNet [17]. We construct the same structure for both classifiers $\mathcal{W}_{\mathcal{F}}$ and $\mathcal{W}_{\mathcal{L}}$. The structure consists of two convolutional layers with kernel size 3×3, stride 1, pad 1 with 1024 units, and a convolutional layer of size 1×1, stride 1 with 1000 units (200 units for CUB-200-2011). For GoogLeNet, we remove the covolutional layers after $Mixed\_6e$ to increase the resolution of the final output. The two classifiers are appended after the layer $Mixed\_6b$ and $Mixed\_6e$ respectively. For VGGNet, we remove the final linear layer and append two classifiers after the fourth and final convolutional block. We will discuss the performance of our model in Section 4 with different positions applied for appending $\mathcal{W}_{\mathcal{F}}$.

For the CoAug module, we apply Alexnet [9] as the feature extractor for both estimated foreground and background regions of input samples. The module utilizes S_L as the only CAM for segmentation. The batch size is set to 48 with at most 12 categories for each batch. All the networks are fine-tuned with the pre-trained weights of ILSVRC2016 [5]. We train the model with an initial learning rate of 0.001 and decay of 0.95 each epoch. The optimizer is SGD with 0.9 momentum and 5×10^{-4} weight decay. For classification result, we follow the instructions in [26], which further averages the scores from the softmax layer with 10 crops.

4.2. Experiment Setup

Dataset and Evaluation To draw a fair comparison, we test our model on ILSVRC2016 [5] validation set and CUB-200-2011 [20] test set, which are two most widely-used benchmarks for WSOL. The ILSVRC dataset has a training set containing more than 1.2 million images of 1,000 categories and a validation set of 50,000 images. In CUB-200-2011, there are totally 11,788 bird images of 200 classes, among which 5,994 images are for training and 5,794 for testing. We leverage the localization metric suggested by [14]. Specifically, the bounding box of an image is correctly predicted if: 1) the model predicts the right image label; 2) more than 50% Intersection-over-Union (IoU) is observed in the overlapped area between predicted bounding boxes and ground truth boxes. For more details, please refer to [14]. We also note that a very recent work by Choe et al. [3] proposes a new set of evaluation metrics providing new perspectives to the evaluation. However, we still use the traditional evaluation metrics, i.e. localization and classification errors, in this work for their feasibility to benchmark with a wide spectrum of existing methods.

Table 1. Effect of our individual modules on CUB-200-2011

Methods	Loc. Error		Class. Error	
	Top-1	Top-5	Top-1	Top-5
VGGnet-DANet [22]	47.48	38.04	24.12	7.73
VGGnet-base	53.42	45.85	24.73	8.96
VGGnet-ConfSeg	39.02	27.17	23.14	6.94
VGGnet-CoAug	40.78	29.36	23.06	6.77
VGGnet-CSoA	**37.69**	**26.49**	**21.41**	**5.94**

Table 2. Effect of positions to insert additional classifier on CUB-200-2011. The number after our model indicates the order of convolutional block in VGGnet

Methods	Loc. Error		Class. Error	
	Top-1	Top-5	Top-1	Top-5
CSoA-3-5	52.76	41.78	26.98	8.58
CSoA-4-5	**37.69**	**26.49**	**21.41**	**5.94**
CSoA-5-5	55.61	44.72	28.60	9.23

Table 3. Localization error with different thresholds for segmentation

Thresholds	Top-1 Error	Top-5 Error
0.2	39.13	27.58
0.3	**37.69**	**26.49**
0.4	38.82	27.11

4.3. Ablation Studies

We make some ablation studies on CUB-200-2011 using VGGnet to evaluate the effects of our individual modules. Besides, we discuss the functions of some hyper-parameters related to the network structure and the inference process.

Effect of ConfSeg and CoAug: For the fair comparison, we first construct a baseline network according to [24] which consists of VGGnet as the backbone and a classifier with the same structure as $\mathcal{W}_{\mathcal{F}}$. As shown in Table 1, the performance of the network with only ConfSeg module reduces the top-1/top-5 *loc. err* by over $14\%/18\%$ compared with our baseline model, and $8\%/10\%$ compared with [22] respectively. It demonstrates that two interacted CAMs can

Table 4. Performance comparison on the CUB-200-2011 test set. The method with star apply a novel non-local approach on the backbone to boost the performance.

Methods	Loc. Error		Class. Error	
	Top-1	Top-5	Top-1	Top-5
GoogLeNet-GAP [26]	58.94	49.34	35.0	13.2
GoogLeNet-SPG [25]	53.36	42.28	-	-
GoogLeNet-ADL [4]	46.96	-	25.4	-
GoogLeNet-DANet [22]	50.55	39.94	28.8	9.4
GoogLeNet-Ours	**46.06**	**34.36**	**23.9**	**6.4**
VGGnet-GAP [26]	55.85	47.84	23.4	7.5
VGGnet-ACoL [24]	54.08	43.49	28.1	-
VGGnet-SPG [25]	51.07	42.15	24.5	7.9
VGGnet-ADL [4]	47.64	-	34.7	-
VGGnet-DANet [22]	47.78	38.04	24.6	7.7
NL-CCAM* [23]	47.60	34.97	26.6	-
VGGnet-Ours	**37.69**	**26.49**	**21.4**	**5.9**

Table 5. Performance comparison on the ILSVRC test set

Methods	Loc. Error		Class. Error	
	Top-1	Top-5	Top-1	Top-5
VGGnet-BP [16]	61.12	51.46	-	-
VGGnet-GAP [26]	57.20	45.14	33.4	12.2
VGGnet-Grad [15]	56.51	46.41	30.4	10.9
VGGnet-ACoL [24]	54.17	40.57	32.5	12.0
VGGnet-ADL [4]	55.08	-	39.3	-
VGGnet-CCAM [23]	51.78	40.64	33.4	-
NL-CCAM* [23]	49.83	39.31	27.7	-
GoogLeNet-BP [16]	61.31	50.55	-	-
GoogLeNet-GAP [26]	56.40	43.00	35.0	13.2
GoogLeNet-ACoL [24]	53.28	42.58	29.0	11.8
GoogLeNet-SPG [25]	51.40	40.00	-	-
GoogLeNet-ADL [4]	51.29	-	**27.2**	-
GoogLeNet-DANet [22]	52.47	41.72	27.5	**8.6**
GoogLeNet-CSoA	**48.81**	**37.46**	28.1	9.8

significantly decrease the localization error since the value of each pixel on the final localization map is double confirmed by both classifiers.

For our model with CoAug module only, the localization result is also much better than the baseline and [22]. Please note that the CoAug module do not have any modification inside the model structure. It only regulates generated CAMs from batch-level, serving as a clustering method among samples with various categories. Therefore, the CoAug module is general enough to be applied to any kind of network as long as the network can generate CAMs-like feature maps.

Finally, our model that combines the ConfSeg and CoAug module can outperform all previous ones on both localization and classification tasks. Especially, the *cls. err* is reduced by about 2% on both top-1 and top-5 results, which is valuable since lots of methods [26, 4, 22] cannot keep the classification performance when trying to improve the localization ability. We mainly attribute it to the double classifiers that refine the bottom layers of our network. Besides, the CoAug module also regulates the network, enforcing it to recognize different parts of the target objects rather than only the most discriminative region.

Position for $\mathcal{W}_{\mathcal{F}}$: For applying ConfSeg module, we extend one more classifier $\mathcal{W}_{\mathcal{F}}$ from the backbone network to obtain additional CAM for interaction. Table 2 shows the results for inserting $\mathcal{W}_{\mathcal{F}}$ after different convolutional blocks. We can obtain the best result when inserting $\mathcal{W}_{\mathcal{F}}$ after the fourth block in VGGNet, i.e. the block right before the final block. In such a configuration, $\mathcal{W}_{\mathcal{F}}$ can do the classification task with features in high semantic level and also produce CAM with different receptive fields. It encourages the effective interaction between two CAMs, which makes their decisions more complementary on ambiguous regions.

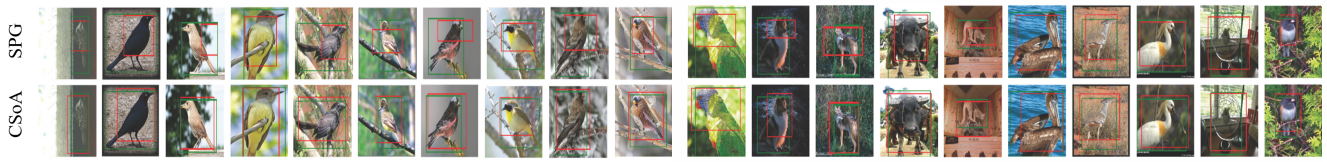
Thresholds for Binary Mask: During the inference, we need one threshold θ to extract foreground regions from two CAMs and then combine them together. To inspect their influence for the localization result, we test different thresholds for our model in Table 3. We can see our model achieves the best result with $\theta = 0.2$. However, θ is still a pre-defined parameter for extracting the final target object. How to remove it or how to make it learnable may be a future work for us to explore.

4.4. Comparison with the state-of-the-arts

We compare our CSoA with state-of-the-art methods on CUB-200-2011 test set and ILSVRC validation set.

As shown in Table 4, on CUB-200-2011 test set, with VGGnet as the backbone network, our method outperforms all others by more than 10% on both Top-1 and Top-5 *loc. err*. It demonstrates the powerful localization ability of our proposed modules with the simple backbone structure. Besides, the classification results of our model are much better than other WSOL methods, which indicates that our proposed method has little negative impact on the recognition performance. This property is important for some real applications, e.g. surveillance cameras that prefer to classify objects correctly and also estimate their positions.

We also evaluate our model with GoogLeNet. Though not as good as VGGnet, our model also becomes the state-of-the-art compared with others. We infer that the smaller gap between GoogLeNet-CSoA and others is because of combination of various operations together for input features in each layer, e.g. *pooling*, 3×3 and 1×1 convolution kernels. It reduces the difference in receptive field between layers, which makes it challenging for multiple classifiers to explore in various levels. The results with Resnet [6] backbone that is only reported in [4] with 37.71% Top-1 *loc. err* has the similar problem since the residual link connects



(a) CUB (b) ILSVRC

Figure 3. Compare localization examples between SPG and CSoA. All visual results from SPG are generated by strictly following author-released code.

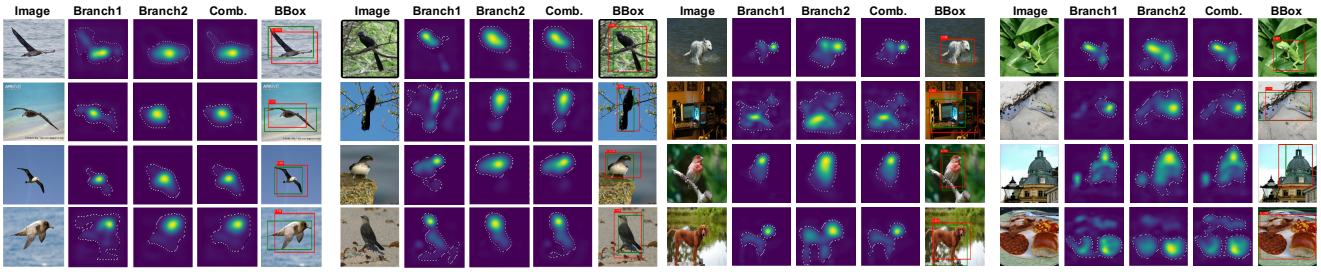


Figure 4. Output visual examples of CSoA. For each dataset, the first three rows show successful results while the last row provides two examples that fail to connect detected parts together.

Table 6. GT-Known localization results for ILSVRC validation set

Methods	Top-1 loc. err
AlexNet-GAP [26]	45.01
AlexNet-HaS [10]	41.26
GoogLeNet-GAP [26]	41.34
GoogLeNet-HaS [10]	39.43
VGGnet-ACoL [24]	37.04
SPG [25]	35.31
ADL [4]	34.59
GoogLeNet-CSoA	**33.80**

different blocks to reduce the receptive differences.

Table 5 shows both classification and localization results on ILSVRC validation set with GoogLeNet. For the localization, our result outperforms all others with the same backbone by over 2% on Top-1 *loc. err*. Besides, our model also achieve better performance compared to the methods with VGGnet. Especially, the NL-CCAM proposed in [23] uses a novel non-local backbone to improve the localization performance, which can be also integrated with our method.

To further demonstrate the localization ability of our model and make a full comparison with other methods, we use ground-truth classification labels for ILSVRC validation set and only evaluate localization performance serving as an "upper-bound" [25]. Denoted as GT-Known *loc. err* in Table 6, our result is still better than others, which indicates an advantage in terms of the pure localization.

Figure 3 visualizes the comparison result between our localization results with SPG [25] since it also considers both foreground and background parts. In most situations, our method can generate more precise bounding boxes than SPG, which demonstrates that the sample-specific segmentation method can achieve better results than using the same pre-fixed thresholds for all images. We will provide more convincing visual examples in the **appendix.**

In addition, in Fig. 4, we visualize the localization maps from CAM at both classifiers, the combined CAM and the final bounding box result of our proposed method on both ILSVRC and CUB-200-2011. The areas inside dashed lines for each CAM indicate the segmented regions for the final result. We can observe that in most cases, the combination of two CAMs has a more stable localization result than any single CAM. It collects the final pixels that are determined by both CAMs and removes ambiguous pixels.

5. Conclusion

We propose CSoA, a novel method for WSOL task. The method consists of two modules that refine the traditional convolutional networks to improve their localization performance without the sacrifice of recognition ability. During learning, the ConfSeg module encourage two classifiers inside the network to generate more precise and complete CAM. In addition, the CoAug module regulate CAM from different samples based on metric approaches in batch level. Our final model outperforms all previous approaches on two public benchmarks. It becomes the new state-of-the-art and provides fresh insights for tackling the WSOL problem.

Acknowledgments

This research was funded in part by the Center of Excellence in Data Science, an Empire State Development-Designated Center of Excellence.

References

[1] Aditya Chattopadhyay, Anirban Sarkar, Prantik Howlader, and Vineeth Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. 10 2017.

[2] Hong Chen, Yifei Huang, and Hideki Nakayama. Semantic aware attention based deep object co-segmentation. In *Asian Conference on Computer Vision*, pages 435–450. Springer, 2018.

[3] Junsuk Choe, Seong Joon Oh, Seungho Lee, Sanghyuk Chun, Zeynep Akata, and Hyunjung Shim. Evaluating weakly supervised object localization methods right. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[4] Junsuk Choe and Hyunjung Shim. Attention-based dropout layer for weakly supervised object localization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[5] J. Deng, W. Dong, R. Socher, L. Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, June 2009.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[7] Kuang-Jui Hsu, Yen-Yu Lin, and Yung-Yu Chuang. Co-attention cnns for unsupervised object co-segmentation. In *IJCAI*, pages 748–756, 2018.

[8] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille, 2015.

[9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[10] Krishna Kumar Singh and Yong Jae Lee. Hide-and-seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[11] Lina Li, Zhi Liu, and Jian Zhang. Unsupervised image co-segmentation via guidance of simple images. *Neurocomputing*, 275:1650–1661, 2018.

[12] Weihao Li, Omid Hosseini Jafari, and Carsten Rother. Deep object co-segmentation. In *Asian Conference on Computer Vision*, pages 638–653. Springer, 2018.

[13] Carsten Rother, Tom Minka, Andrew Blake, and Vladimir Kolmogorov. Cosegmentation of image pairs by histogram matching-incorporating a global constraint into mrfs. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 1, pages 993–1000. IEEE, 2006.

[14] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, Dec 2015.

[15] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[16] K Simonyan, A Vedaldi, and A Zisserman. Deep inside convolutional networks: visualising image classification models and saliency maps. pages 1–8. ICLR, 2014.

[17] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

[18] Peng Tang, Xinggang Wang, Angtian Wang, Yongluan Yan, Wenyu Liu, Junzhou Huang, and Alan Yuille. Weakly supervised region proposal network and object detection. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[19] Zhiqiang Tao, Hongfu Liu, Huazhu Fu, and Yun Fu. Image cosegmentation via saliency-guided constrained clustering with cosine similarity. In *Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[20] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011.

[21] Xiang Wang, Shaodi You, Xi Li, and Huimin Ma. Weakly-supervised semantic segmentation by iteratively mining common object features. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1354–1362, 2018.

[22] Haolan Xue, Chang Liu, Fang Wan, Jianbin Jiao, Xiangyang Ji, and Qixiang Ye. Danet: Divergent activation for weakly supervised object localization. In *The IEEE International Conference on Computer Vision (ICCV)*, October 2019.

[23] Seunghan Yang, Yoonhyung Kim, Youngeun Kim, and Changick Kim. Combinational Class Activation Maps for Weakly Supervised Object Localization. *arXiv e-prints*, page arXiv:1910.05518, Oct. 2019.

[24] Xiaolin Zhang, Yunchao Wei, Jiashi Feng, Yi Yang, and Thomas S. Huang. Adversarial complementary learning for weakly supervised object localization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[25] Xiaolin Zhang, Yunchao Wei, Guoliang Kang, Yi Yang, and Thomas Huang. Self-produced guidance for weakly-supervised object localization. In *The European Conference on Computer Vision (ECCV)*, September 2018.

[26] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.