IMPROVING CLASS ACTIVATION MAP FOR WEAKLY SUPERVISED OBJECT LOCALIZATION

Zhenfei Zhang [1] *Ming-Ching Chang* [1] *Tien D. Bui* [2]

[1] University at Albany, State University of New York, NY, USA
[2] Concordia University, Montreal, Quebec, Canada

ABSTRACT

We propose a Weakly Supervised Object Localization (WSOL) method that can locate an object within a given image using a pre-trained network learned with only class labels without location annotations. Most existing WSOL methods rely on thresholding a Class Activation Map (CAM) generated by the pre-trained network to highlight and localize the object. However such approaches often produce incomplete object bounding boxes, as only the discriminative parts of the object are selected during thresholding. We revisit current CAM-based WSOL approaches and propose a pipeline to: (1) refine the CAM map using Weighted Global Average Pooling (WGAP), (2) recombine weights to make use of the negative features, (3) adaptively select a suitable threshold to achieve better object localization. Our method does not require additional learning or hyperparameter tuning. We show that our simple approach can achieve competitive results when evaluated on the CUB-200-2011 and ILSVRC 2016 datasets against other state-of-the-art methods.

***Index Terms*—** weakly supervised object localization, class activation map, weighted global average pooling

1. INTRODUCTION

With the rise of deep learning in computer vision, *supervised* object detection [1, 2] and localization [3] can achieve satisfactory results with a limitation that data annotation can be costly and time-consuming. To this end, *weakly supervised* learning methods which rely on only partial annotations have attracted significant attentions. In this work, we focus on the **Weakly Supervised Object Localization (WSOL)** task [4, 5] that localizes an object within a given image, by using a network that is pre-trained using only class labels, *i.e.* without location annotation during training.

Since a seminal work on improving CNN localization via global average pooling, the Class Activation Map (CAM) [6] technique allows the classification-trained CNN to both classify the image and localize class-specific image regions, which has been used as a main solution toward the WSOL task [4]. However, the object localization map of CAM can only highlight salient/discriminative parts of the object, rather than highlighting the whole object (which is the original goal of WSOL by definition). This way, the object localization performance evaluated by the standard intersection-over-union (IoU) metric between the detection and the corresponding ground-truth box can be seriously affected [5].

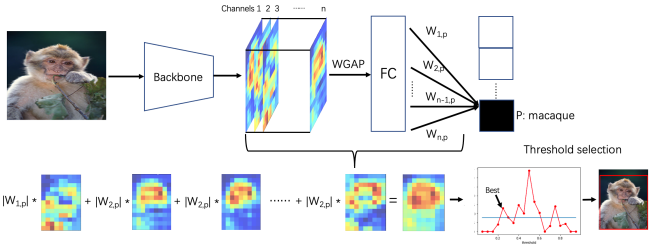

Fig. 1. The proposed weakly-supervised object localization (WSOL) pipeline. The inputs are images with only classification labels (no ground-truth bounding boxes). WGAP is performed between the last convolutional layer and the FC layer. The pooled feature vector of WGAP passes through the FC layer in which the weight of each channel is generated. In the generation of CAM, both positive and negative channels are activated. A newly improved threshold selection method is proposed to determine the final object box localization by thresholding the CAM.

Numerous methods have been proposed to alleviate the above issue by *hiding* or *masking out* salient object parts [7, 8, 9, 10, 11, 12, 13, 14, 15]. Other method relies on data augmentation [16], loss function re-formulation [17], or attention map improvement [18] for better localization. Although the saliency-masking methods can generate better CAM by not focusing only on the most salient peak of the CAM, they rely on the use of additional networks or overheads, which may make the model harder to train. In this way, the backbone network architectures are largely modified, which sacrifices object classification accuracy. It is still an open challenge in WSOL on how to achieve better object localization while maintaining classification accuracy.

Bae *et al.* [12] re-visited the CAM pipeline and determine three root causes of the poor WSOL performance: (1) *bias introduced by the Global Average Pooling (GAP)* [6] in assigning a higher weight to a channel with small activated area; (2) *weighted average of both positive and negative CAM feature maps*, which may inhibit the less discriminative region, and

(3) *the final thresholding* to determine a bounding box of the targeted object. Our work is motivated by these observations, while our approach differs from theirs in two aspects. We directly make use of the spatial information provided by the negative weights of the CAM feature maps to improve object localization, while such insight were completely discarded in [12]. Also, in [12] the values for thresholding the CAM and GAP maps are manually determined from a fixed percentile of the peak value, which does not scale up to new test scenarios.

We propose three solutions corresponding to the aforementioned CAM-based issues to effectively address them. (1) We develop a Weighted Global Average Pooling (WGAP) using spatial-softmax function [19] to automatically enhance the activated information and inhibited the background pixels without the need of a hard threshold on the feature maps. (2) We recombine the weighted feature maps via linear summation to generate a fused CAM that captures richer spatial information for WSOL object localization. (3) Instead of setting a fixed threshold to extract the bounding boxes from CAM, we develop a two-stage localization method that first evaluate potential threshold values and then adaptively select the best threshold for each image. To the best of our knowledge, our method is the first in WSOL that is adaptive, fully automatic, and does not require extra data-driven learning or hyperparameter tuning. Fig. 1 shows the overview of the proposed method. Evaluation is performed on the CUB-200-2011 and ILSVRC 2016 datasets. Results show that our method achieves comparable performance against several state-of-the-art works in terms of both object localization and classification accuracy.

2. RELATED WORKS

Weakly-supervised object localization. Most WSOL works follow the conventional Class Activation Map (CAM) [6] pipeline by training a CNN for object classification and then generating a CAM via Global Average Pooling (GAP) on last layer of convolutional feature maps. Lastly, the targeted object is extracted by thresholding the CAM to localize each object as a bounding box [12]. Depending on how the activated maps are calculated and the corresponding threshold selection, the targeted object might not be localized well.

Many studies address the problems following the CAM pipeline. The Hide-and-Seek (HaS) [4] is an augmentation method by masking out image patches randomly that essentially augments the training set. By removing grid patches, the model can focus on the remaining regions to better discriminate and localize objects during training. The Adversarial Complementary Learning (ACoL) [8] uses two parallel classifiers to remove sparse activation regions via adversarial learning. The Self-Produced Guidance (SPG) [7] adds three different layers to Inception-V3 [20] in order to progressively learn three masks (foreground, unsure, background) to incorporate high-confidence regions. The Attention-based Dropout Layer (ADL) [9] adopts the self-attention mechanism to hide

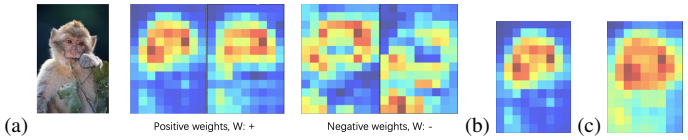

Fig. 2. Examples of (a) positive and negative feature maps. (b) CAM using the original method. (c) CAM of our method.

the discriminative regions of the feature map randomly using fixed probabilities. The Attention-based Selection Strategy (ASS) [14] considers different characteristics of each input image and dynamically determines the region dropping mask. Numerous region hiding methods [10, 15] work similarly in generating the mask to suppress the small and sparse CAM peak regions.

Aside from methods focusing on hiding the peak CAM regions during model training/testing, CutMix [17] cuts the input image and mix with others for data augmentation. DANet [16] leverages two new loss functions to capture the entire patterns. Both methods are very heavy for the localization tasks. The recent work of TS-CAM [18] uses a new token semantic coupled attention map running with a Transformer variant backbone to achieve remarkable performance.

3. THE ISSUES OF CLASS ACTIVATION MAP

The CAM processing pipeline for WSOL is based on the following steps: (1) Apply GAP between the last conv layer and the fully connected (FC) layer. (2) Use the weights of each channel generated from the FC to compute the final CAM. (3) Threshold the CAM to obtain object localization as a bounding box output. In this paper, we address each of the three issues to improve object localization performance, while keeping the simplicity and classification accuracy.

Global Average pooling (GAP) [21] plays an important role in CAM. GAP can effectively solve the issue of input sizes and retain spatial information in the feature maps. However, GAP assigns equal weights to every pixel whether the pixels are activated or not. This way, background pixels are encouraged and the targeted regions are penalized, which is unfair for the stored spatial information.

Weights of the feature maps. During training of the CAM network, the FC layer assigns both positive and negative weights to the feature maps based on the classification outcome. The *positive* feature map carries information for object classification and localization, whiles the *negative* feature map also carries useful information, see Fig. 2(a). Even with pooling performed within each block, feature map of the last conv layer still retains sufficient spatial information, no matter the weights are positive or negative. In other words, the CAM network still 'memorizes' less important parts during training, the less discriminative regions are mainly in the negative channels. Another constraint in this WSOL training process is that the object classification task only requires the most discriminative region. Therefore, the learned FC layer

at the end assigns positive weights to the feature containing only the (most) discriminative parts.

Thresholding CAM produces a binary object mask that yields the bounding box as the WSOL output. Here, how best to determine the threshold is the key. Existing works use a fixed manual threshold calculated by a percentile of the peak CAM value. For the cases of a high peak CAM value or a high threshold value, the obtained object bounding box may only focus on a small region of the activation map, even if the activated regions have already covered a significant portion of the object. This drawback degrades the WSOL performance.

4. THE PROPOSED METHOD

Based on the three issues of CAM reviewed in § 3, we proposed three corresponding solutions to address them and thus improve the WSOL localization, while our method can retain the classification accuracy.

4.1. Weighted Global Average Pooling (WGAP)

Inspired by the motivation in the last section, we proposed a weighted GAP that can assign different weights to the feature map based on the ground-truth classification, inspired by [19]. We compute the weight map using 2D softmax: $S_c(i,j) = \frac{exp(a_c(i,j))}{\sum_{(i',j')} a(i',j')}$, where c is the channel of the feature map and (i, j) is the pixel coordinate. We obtain a probability distribution that represents the pixel weights based on the classification labels. By multiplying the feature maps with the weights map, the weighted feature maps are generated so that the targeted pixels will be encouraged and the background region will be penalized. The proposed WGAP is:

$$G_W = \left[\sum_{(i,j)} P_1(i,j) S_1(i,j), ..., \sum_{(i,j)} P_c(i,j) S_c(i,j) \right] / (W \times H), \quad (1)$$

where $P \subseteq R^{H \times W \times C}$ is the feature map from the last convolutional layer. H and W are the height and width of feature map, C is the channel number. The output of WGAP is highly depending on the activated pixels that stores beneficially spatial information.

4.2. Recombining the Weights of the FC Layer

The weights of the FC layer is used to encourage positive features and inhibit negative ones when computing the CAM. Let N_p and N_n denote the number of positive weights and negative weights, respectively. Let $F \subseteq R^{H \times W \times C}$ denote the feature map with positive weights and $G \subseteq R^{H \times W \times C}$ for negative weights. The original CAM can be expressed as:

$$M_c = M_p + M_n = \sum_{p=1}^{N_p} w_{p,c} \cdot F_p + \sum_{n=1}^{N_n} w_{n,c} \cdot G_n, \quad (2)$$

where $M_c \subseteq R^{H \times W}$ is the original CAM, $M_p \subseteq R^{H \times W}$ and $M_n \subseteq R^{H \times W}$ are the positive and negative CAM with sizes N_p and N_n, respectively. In Eq.(2), $M_n < 0$ since all the weights in sum are negative. This way, the negative channels are thus inhibited.

Recall in § 4.1 that the negative channels still retain very useful information for WSOL, especially at the off-peak regions. Therefore, we specifically exploit the negative channels to improve WSOL. Fig. 2(b) shows the original CAM, where the negative channels are ignored. In contrast, we incorporation the information from the negative channels to generate an improved CAM as shown in Fig. 2(c). Our improved CAM is calculated as:

$$\hat{M}_c = M_p + |M_n| = \sum_{p=1}^{N_p} w_{p,c} \cdot F_p + \sum_{n=1}^{N_n} |w_{n,c}| \cdot G_n \quad (3)$$

By calculating the absolute values of the negative features, these channels can be considered for activation as well. It is important that the recombining weights are only applied during CAM generation, such that all training weights are not altered. Therefore, model training is not affected.

4.3. Automatic CAM Threshold Selection

Object localization can also be regarded as finding the object boundary edges. In digital image processing, object boundary can be determined as the set of points with highest local gradients. In other words, the pixels with largest pixel difference are likely at the edges. Getting back to WSOL, the CAM can be understood as an Attention Map in which the targeted pixels are highly activated. We hypothesize that there exists a group of pixels on CAM that can be associated with large gradients and can thus represent the object boundary. Therefore, we take thresholds at equal intervals from 0 to 1. The set of threshold can be expressed using $[\eta_1, \eta_2, ...\eta_k]$ where k is the number of potential thresholds based on the interval. For each threshold η_l where $l \in [1, k]$, we compare all the pixel values with $\eta_l \cdot \max M_c^f$ where M_c^f denotes the CAM. In other words, we count the pixel values $> \eta_l \cdot maxM_c^f$ in CAM as target pixels. This way, we obtain the boundary of targeted pixels. We then calculate the average of the pixel values of each set of points using $V_{\eta_l} = \sum_1^n a_n/n$, where the n is the number of edge nodes and a is the pixel value. The boundary with the largest difference is determined as the final threshold. However, our experiments show that WSOL performance obtained this way is very similar to CAM. A major reason is that CAM can only focus on the most discriminative part. Although other target pixels are also activated, the activations are still negligible when compared to the most discriminative portion.

Automatic CAM thresholding. In our experiments, we notice that object bounding box can only be adjusted when the difference of CAM pixels is significantly greater than the average. This is because even if the activated pixels are slightly altered, the 4 extreme points of the bounding box (namely, upper-left and lower-right) stay intact. To this end, we propose an *automatic approach to determine a preferable CAM threshold* following a two-stage approach. Our method

Method	Top-1 Cls	Top-1 Loc	GT-known Loc
VGG-CAM[6]	**76.14**	33.95	55.1
VGG-ADL[9]	65.27	53.36	75.4
VGG-DANet[9]	75.40	52.52	67.7
VGG-ACoL[8]	71.90	45.92	59.3
VGG-I^{2}C[13]	68.40	56.00	-
VGG-MEIL[10]	75.00	57.46	73.8
VGG-MCI[15]	72.59	58.12	-
VGG-ICL[11]	73.40	57.50	-
VGG-RCAM[12]	74.91	61.30	-
InceptionV3-SPG[7]	75.50	46.46	-
InceptionV3-ADL[9]	70.43	47.74	-
InceptionV3-DANet[16]	71.20	50.55	67.0
Ours	75.82	**61.85**	**82.32**

Table 1. Quantitative evaluation of performance compared with state-of-the-art on the CUB-200-2011 dataset.

Method	Top-1 Cls	Top-1 Loc	GT-known Loc
VGG-CAM[6]	66.6	42.8	59.0
VGG-ADL[9]	69.48	44.9	75.4
VGG-ACoL[8]	67.5	45.8	63.0
VGG-CutMix[17]	-	43.5	-
VGG-I^{2}C[13]	68.40	47.4	63.9
VGG-MEIL[15]	**73.31**	46.8	-
VGG-ICL[11]	64.0	47.2	-
VGG-RCAM[12]	67.28	44.69	-
InceptionV3-CAM[7]	75.50	46.3	64.7
InceptionV3-ADL[9]	70.43	48.7	-
InceptionV3-ACoL[8]	71.20	46.7	-
Ours	68.32	**50.1**	**65.4**

Table 2. Quantitative evaluation of performance compared with state-of-the-art on the ILSVRC ImageNet-1K dataset.

can effectively eliminate the limitation of using a fixed CAM threshold. We first generate potential boundary values and then select the best among them based on a simple analysis; see Fig. 3 for explanation. The difference of each threshold η_l is defined as $D_{\eta_l} = \left|V_{\eta_l} - V_{\eta_{l-1}}\right|$. In this example, 0.25 is selected as the best threshold.

5. EXPERIMENT RESULTS

Implementation Details. We use VGGnet [22] as the backbone network, and select the best threshold from the set of $[0, 0.05, ...1]$. The network is pre-trained on ILSVRC dataset [23] and fine-tuned with learning rate 0.0001 and batch size 64. Model is trained on GeForce RTX 2080 GPU.

Datasets. The CUB-200-2011 [24] and ImageNet ILSVRC 2016 [23] datasets are used for evaluation. There are approximately 1.3M training data and 50K validation images belonging to 1,000 categories in ImageNet ILSVRC 2016. The CUB-200-2011 dataset has 5,994 training data and 5,794 validation images including 200 different species of birds.

Evaluation metrics. We follow most existing WSOL works in using the Top-1 Localization accuracy and Top-1 Classification accuracy for performance evaluation. It is shown in [5] that those two metrics along are unfair to the WSOL task. Thus, we also provide the Ground-truth localization accuracy for comparison. The Ground-truth localization accuracy computes the localization performance when the ground-truth classes are available.

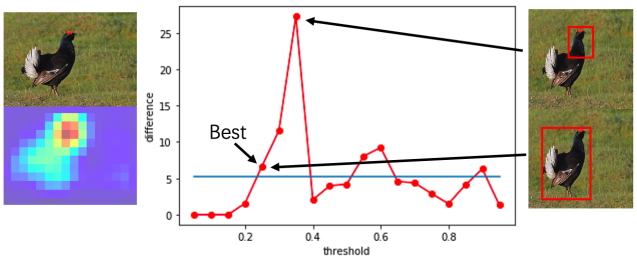

Fig. 3. Automatic CAM threshold selection. (Left) Input image and CAM. (Center) Each point on the red poly-line shows the difference of the average pixel value on the boundary and its immediately-previous average value. Blue line shows the average threshold values, which is used to determine the best threshold. (Right) The resulting object localization box from two corresponding threshold values; clearly the box determined using the blue line better localizes the object.

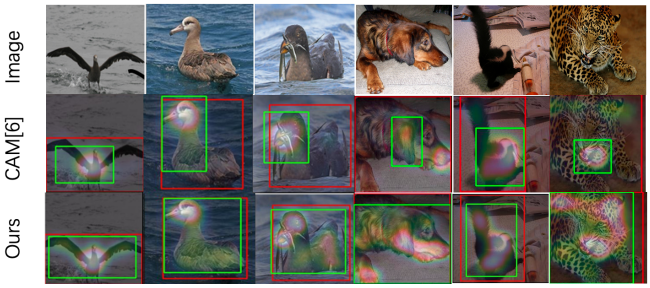

Fig. 4. Comparison with CAM [6] on CUB-200-2011 (first 3 columns) and ILSVRC 2016 datasets (last 3 columns). Green box depicts prediction and red box depicts ground truth.

Tables 1 and 2 show the comparison of our WSOL method in comparison with the state-of-the-art. In the CUB dataset, our method achieves the best localization results with the least classification sacrifices. In the ImageNet dataset, most existing methods obtain higher classification accuracy over CAM. We believe the reason is that ImageNet is a large and containing multiple objects, so methods with deeper architecture may obtain better results. However, observe in Table 1 that our method performs the best in the metric of **GT-known Localization Accuracy**, which only considers the localization results. Fig. 4 shows some localization examples. Observe that our Class Activation Maps cover the objects very well, and thus the bounding boxes better cover the activated regions.

6. CONCLUSION

In this work, we improve the classic CAM based approach for WSOL by addressing the three observed issues using corresponding solutions. Our proposed two stage localization method does not require setting hyper-parameters on thresholding. Our method is the first of the kind that can evaluate the bounding box without the Ground-truth label and assign the best threshold to test images. Experiment results show that our method works very well on evaluation datasets. **Future work** includes the adaptive selection of the threshold value.

7. REFERENCES

[1] T. Lin, P. Dollar, and R. Girshick. Feature pyramid networks for object detection. In *CVPR*, 2017.

[2] J. Redmon, S. Divvala, R. Girshivk, and A. Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016.

[3] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler. Efficient object localization using convolutional networks. In *CVPR*, 2015.

[4] K. K. Singh and Y. J. Lee. Hide-and-Seek: Forcing a network to be meticulous for weakly-supervised object and action localization. In *ICCV*, 2017.

[5] J. Choe, S. J. Oh, S. Lee, S. Chun, Z. Akata, and H. Shim. Evaluating weakly supervised object localization methods right. In *CVPR*, 2020.

[6] B. Zhou, A. Khosla, A. Lapedriza, A. Olive, and A. Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016.

[7] X. Zhang, Y. Wei, G. Kang, Y. Yang, and T. S. Huang. Self-produced guidance for weakly-supervised object localization. In *ECCV*, 2018.

[8] X. Zhang, Y. Wei, J. Feng, Y. Yang, and T. S. Huang. Adversarial complementary learning for weakly supervised object localization. In *CVPR*, 2018.

[9] J. Choe and H. Shim. Attention-based dropout layer for weakly supervised object localization. In *CVPR*, 2019.

[10] J. Mai, M. Yang, and W. Luo. Erasing integrated learning : A simple yet effective approach for weakly supervised object localization. In *CVPR*, 2020.

[11] M. Ki, Y. Uh, W. Lee, and H. Byun. In-sample contrastive learning and consistent attention for weakly supervised object localization. In *ACCV*, 2020.

[12] W. Bae, J. Noh, and G. Kim. Rethinking class activation mapping for weakly supervised object localization. In *ECCV*, 2020.

[13] X. Zhang, Y Wei, and Y. Yang. Inter-image communication for weakly supervised localization. In *ECCV*, 2020.

[14] Z. Zhang and T. D. Bui. Attention-based selection strategy for weakly supervised object localization. In *ICPR*, 2020.

[15] S. Babar and S. Das. Where to look?: Mining complementary image regions for weakly supervised object localization. In *WACV*, 2021.

[16] H. Xue, C. Liu, F. Wan, J. Jiao, X. Ji, and Q. Ye. DANet: Divergent activation for weakly supervised object localization. In *ICCV*, 2019.

[17] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo. CutMix: Regulation strategy to train strong classifiers with localization feature. In *ICCV*, 2019.

[18] W. Gao, F. Wan, X. Pan, Z. Peng, Q. Tian, Z. Han, B. Zhou, and Q. Ye. TS-CAM: Token semantic coupled attention map for weakly supervised object localization. In *ICCV*, 2021.

[19] S. Levine, C. Finn, T. Darrell, and P. Abbeel. End-to-end training of deep visuomotor policies, 2017.

[20] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, and S. Reed. Going deeper with convolutions. In *CVPR*, 2015.

[21] M. Lin, Q. Chen, and S. Yan. Network in network. In *ICLR*, 2014.

[22] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[23] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. ImageNet large scale visual recognition challenge. In *IJCV*, 2015.

[24] 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.