Weakly Supervised Object Detection Using Class Activation Map

Anonymous Author Submission

***Abstract*—Class activation map helps visualize the region of a given object category in an image that an accurate classifier uses for prediction. The original work also extended this method to solve weakly supervised object localization. However, results have shown that class activation map only shows the discriminative part of an object instead of the whole object. Recent proposed solutions help alleviate this problem and allow class activation map to achieve state-of-the-art localizing performance. In this paper, we take advantage of those solutions and propose modifications to use CAM for detecting multiple objects instead.**

***Index Terms*—class activation map, weakly supervised object detection**

I. Introduction

Class Activation Map (CAM) introduced by Zhou et al. [1] to visualize the region of a given object category in an image. This method utilizes the weights from the fully connected layer and the feature maps before the global average pooling layer to generate CAM. CAM allows us to know which region of the image is used by an accurate classifier to make a prediction. Zhou et al. extended their work to solve weakly supervised object localization (WSOL). WSOL aims to find a region of a targeted object in an image using image-level labels as training data. Weakly supervised approaches are great alternatives to fully supervised approaches [2] [3] as they require less effort and cost to collect, label, and train data.

However, research has shown that CAM only highlights the most discriminative part of an image, thus affecting the localization performance. Kim et al. [4] solved this problem by addressing issues when training a classifier to generate CAM. The proposed method expands the discriminative region to the whole object, thus achieving state-of-the-art localization performance on ImageNet-1k [5] and CUB-200-2011 [6]. Another solution from Yang et al. [7] is to use a function to combine CAM from different categories.

Although the discriminative part has been solved, CAM only focuses on finding the region of a single object. Utilizing Kim et al. [4] work, we propose modifications to use CAM for object detection instead of object localization.

In summary, the contributions of this paper are as follows:

- We propose a way to extend [4] for object detection instead of object localization.
- We show how using Otsu as a post-processing step can affect the result of the proposed method.

II. Related Works

Weakly Supervised Object Detection (WSOD). WSOD has gained significant attention as it aims to localize and detect objects using only image-level labels during training, alleviating the need for costly instance-level annotations. Existing WSOD approaches can be broadly categorized into two main paradigms: Multiple Instance Learning (MIL) based and CAM based methods.

MIL-based MIL-based techniques [8], [9], [10], [11] treat an image as a bag of region proposals or instances, where at least one instance should contain the object of interest. These methods iteratively learn to classify the proposals and refine the object localization.

CAM-based On the other hand, CAM-based approaches [1], [4], [7] leverage class activation maps generated from classification networks to localize discriminative object regions. However, vanilla CAM often highlights only the most discriminative parts rather than full object extents. Several works have proposed solutions to this issue, either by introducing auxiliary losses [Kim et al., 2022] to encourage complete object activation or by combining activation maps across categories [Yang et al., 2019].

Our proposed multi-label formulation is motivated by the fact that many object detection datasets contain images with multiple object categories present. By extending CAM to multi-label classification, we aim to improve the localization performance of WSOD methods on such datasets.

III. Preliminary

A. Class Activation Map

CAM is acquired by first training a multi-class classifier with a global average pooling (GAP) layer between the last convolutional layer and the fully connected (FC) layer with softmax as activation function [1]. The prediction loss is calculated by cross-entropy (CE) function:

$$\mathcal{L}_{\text{CE}} = -\sum_{i}^{C} y_i log(s_i) \tag{1}$$

with $y_i \in \{0, 1\}$ is the ground truth label and s_i is the logit of the i-th class.

CAM is then generated by taking the weighted sum of the feature maps from the last convolutional layer with the weights from the FC layer:

$$\text{CAM}(x) = \mathbf{w}_c^\top F(x) \tag{2}$$

where $\mathbf{w}_c$ is the weight of the c-th class and $F(x) \in \mathbb{R}^{H \times W \times D}$ is the feature map of the last convolutional layer.

However, it is widely observed that CAM only highlights the most discriminative part of an object. This drawback affects the localization performance of CAM thus detection performance also affected.

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

Although CAM is generated from an accurate classifier, the classification performance does not translate to localization performance. To address this issue, Kim et al. [4] first interpreted CAM as the degree of aligment between the direction of input features and the direction of class-specific vectors.

Kim et al. interpreted from Eq. 2 that the value of CAM at each spatial location $u \in \{1, \dots, HW\}$ is the dot product between the feature map at that location and the weight of the class. From that, they decomposed Eq. 2 further:

$$\begin{aligned}\text{CAM}_u(x) &= \mathbf{w}_c \cdot F_u(x) \\ &= ||\mathbf{w}_c||\ ||F_u(x)||\ \underbrace{\frac{\mathbf{w}_c \cdot F_u(x)}{||\mathbf{w}_c||\ ||F_u(x)||}}_{S(\mathbf{w}_c, F_u(x))}\end{aligned} \tag{3}$$

where $S(a, b)$ is the cosine similarity between two vectors a and b. From this decomposition, the value at each spatial location u is the product between the norm of a class-specific weight and the norm of the feature map at that location, and the cosine similarity between the two vectors. Eq. 3 can be rewritten as:

$$\text{CAM} = ||\mathbf{w}_c|| \cdot \mathcal{F} \odot \mathcal{S} \tag{4}$$

with $\mathcal{F} \in \mathbb{R}^{H \times W}$ and $\mathcal{S} \in \mathbb{R}^{H \times W}$ be the norm map and similarity map, respectively, where $\mathcal{F}_u = ||F_u||$ and $\mathcal{S}_u = S(\mathbf{w}_c, F_u(x))$.

Likewise, the classification score calculated from GAP output, $f(x) = GAP(F(x)) \in \mathbb{R}^D$ can be formulated as:

$$\begin{aligned}\text{logit}_c(x) &= \mathbf{w}_c \cdot f(x) \\ &= ||\mathbf{w}_c||\ ||f(x)||\ S(\mathbf{w}_c, f(x))\end{aligned} \tag{5}$$

Because $f(x)$ is fixed for x, the scale variation of $||\mathbf{w}_c||$ across each class c is not large so the similarity between $||mathbfw_c||$ and $f(x)$ need to be large for c to be predicted. Here, Kim et al. found that the classifier is trained to maximize the similarity between weights of a class and the output of the GAP. This lead to the model only localize discriminative part of the object. To brigde this gap, Kim et al. proposed two method: align feature directions and consistency with attentive dropout. The former method introduce two loss that complement each other: norm loss and similarity loss. The latter method introduce a dropout layer to the model architecture when training.

1) Similarity loss: Similarity loss aims to increasing the similarity in foreground region and suppress it in background region based on high and low value region on a normalized $\mathcal{F}$. Similarity loss helps expand the activation region to the whole object.

$$\begin{aligned}&\mathcal{R}^{\text{norm}}_{\text{fg}} = \{u | \hat{\mathcal{F}}_u > \tau_{\text{fg}}\}, \\ &\mathcal{R}^{\text{norm}}_{\text{bg}} = \{u | \hat{\mathcal{F}}_u < \tau_{\text{bg}}\}, \\ &\text{where } \hat{\mathcal{F}} = \frac{\mathcal{F} - \min_i \mathcal{F}_i}{\max_i \mathcal{F}_i - \min_i \mathcal{F}_i}\end{aligned} \tag{6}$$

$$\mathcal{L}_{\text{sim}} = -\frac{1}{|\mathcal{R}^{\text{norm}}_{\text{fg}}|} \sum_{u \in \mathcal{R}^{\text{norm}}_{\text{fg}}} \mathcal{S}_u + \frac{1}{|\mathcal{R}^{\text{norm}}_{\text{bg}}|} \sum_{u \in \mathcal{R}^{\text{norm}}_{\text{bg}}} \mathcal{S}_u \tag{7}$$

2) Norm loss: Norm loss aims to active area in object region and suppress in background region in $\hat{\mathcal{F}}$ based on positive and negative region estimated from S_u.

$$\begin{aligned}&\mathcal{R}^{\text{sim}}_{\text{fg}} = \{u | \mathcal{S}_u > 0\}, \\ &\mathcal{R}^{\text{sim}}_{\text{bg}} = \{u | \mathcal{S}_u < 0\}\end{aligned} \tag{8}$$

$$\mathcal{L}_{\text{norm}} = -\frac{1}{|\mathcal{R}^{\text{sim}}_{\text{fg}}|} \sum_{u \in \mathcal{R}^{\text{sim}}_{\text{fg}}} \hat{\mathcal{F}}_u + \frac{1}{|\mathcal{R}^{\text{sim}}_{\text{bg}}|} \sum_{u \in \mathcal{R}^{\text{sim}}_{\text{bg}}} \hat{\mathcal{F}}_u \tag{9}$$

3) Consistency with Attentive Dropout: Kim et al. [4] introduces a dropout layer between the backbone's layer in the training process. This dropout layer stochastically dropping the activation of an itermediate feature map $\mathcal{F}'$ at the spatial location of whose channel-wised average value is highers than a threshold. Then $\mathcal{F}'$ is passed through the rest of the model's backbone to attain $\mathcal{F}_{drop}$. Finally, $\mathcal{L}_1$ loss is calculated between $\mathcal{F}_{drop}$ and $\mathcal{F}$.

$$\mathcal{L}_{drop} = ||F(x) - F(x)_{drop}||_1 \tag{10}$$

4) Training Scheme: The final loss function is the combination of CE loss, similarity loss, norm loss, and dropout loss:

$$\mathcal{L}_{total} = \mathcal{L}_{\text{CE}} + \lambda_{\text{sim}}\mathcal{L}_{\text{sim}} + \lambda_{\text{norm}}\mathcal{L}_{\text{norm}} + \lambda_{\text{drop}}\mathcal{L}_{\text{drop}} \tag{11}$$

The model need to be trained for a few epoch without the feature direction alignment loss in order to stabilize feature map for classification. Then, the feature direction alignment loss is added to the total loss function.

$$\mathcal{L}_{warm} = \mathcal{L}_{\text{CE}} + \lambda_{\text{drop}}\mathcal{L}_{\text{drop}} \tag{12}$$

IV. Proposed Method

A. From multi-class to multi-label

In the original work by Zhou et al. [1], CAMs were inferred from a multi-class classifier. This approach, which utilizes a softmax activation function and CE loss, is not well-suited for multi-label data. To address this limitation, we propose a multi-label classification framework for CAM generation. This framework employs a sigmoid activation function $\frac{1}{1+e^{-x}}$ in the final layer, enabling the model to predict the presence or absence of multiple classes simultaneously. Binary cross-entropy (BCE) loss is then used as the loss function during training to optimize these predictions:

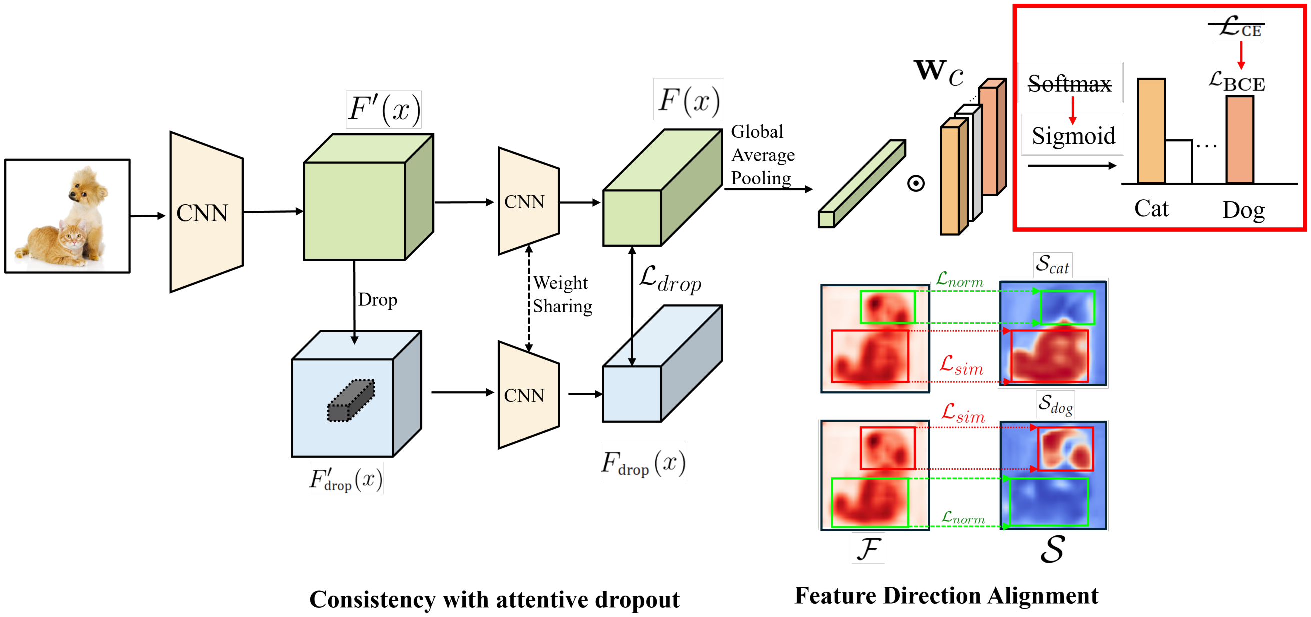


Fig. 1. The proposed modification to Kim et al. [4] work

$$\mathcal{L}_{\text{BCE}} = -\sum_{i}^{C} y_i \log(\sigma(s_i)) \tag{13}$$

We also train both original multi-class architecture [1] [4] on multi-label dataset to compare with our proposed modification. To achieve this, with each image containing multiple labels, we add duplicate images for each label. For example, an image with labels {cat, dog} will be duplicated into two images: one with label cat and another with label dog. This way, the model can learn to predict each label independently and CE loss can be used as proposed by both original works.

B. Architecture

We use most of the architecture from Kim et al. [4] work. Fig. 1. show our proposed modification. We modify the fully connected layer to predict the presence or absence of multiple classes using sigmoid activation function. However, both norm loss and similarity loss are class-specific, so we need to modify the loss function to accommodate multi-label classification. The modification is simply average the loss of each present class:

$$\mathcal{L}'_{\text{sim}} = \frac{1}{n}\sum_{i}^{c} y_i \Big(-\frac{1}{|\mathcal{R}^{\text{norm}}_{\text{fg}}|}\sum_{u\in\mathcal{R}^{\text{norm}}_{\text{fg}}} \mathcal{S}_u + \frac{1}{|\mathcal{R}^{\text{norm}}_{\text{bg}}|}\sum_{u\in\mathcal{R}^{\text{norm}}_{\text{bg}}} \mathcal{S}_u\Big) \tag{14}$$

$$\mathcal{L}'_{\text{norm}} = \frac{1}{n}\sum_{i}^{c} y_i \Big(-\frac{1}{|\mathcal{R}^{\text{sim}}_{\text{fg}}|}\sum_{u\in\mathcal{R}^{\text{sim}}_{\text{fg}}} \hat{\mathcal{F}}_u + \frac{1}{|\mathcal{R}^{\text{sim}}_{\text{bg}}|}\sum_{u\in\mathcal{R}^{\text{sim}}_{\text{bg}}} \hat{\mathcal{F}}_u\Big) \tag{15}$$

where n is the number of present classes and $y_i \in \{0, 1\}$ is the groundtruth for each class.

We also modify the total loss function Eq. 11 to accommodate multi-label classification by changing CE loss to BCE loss:

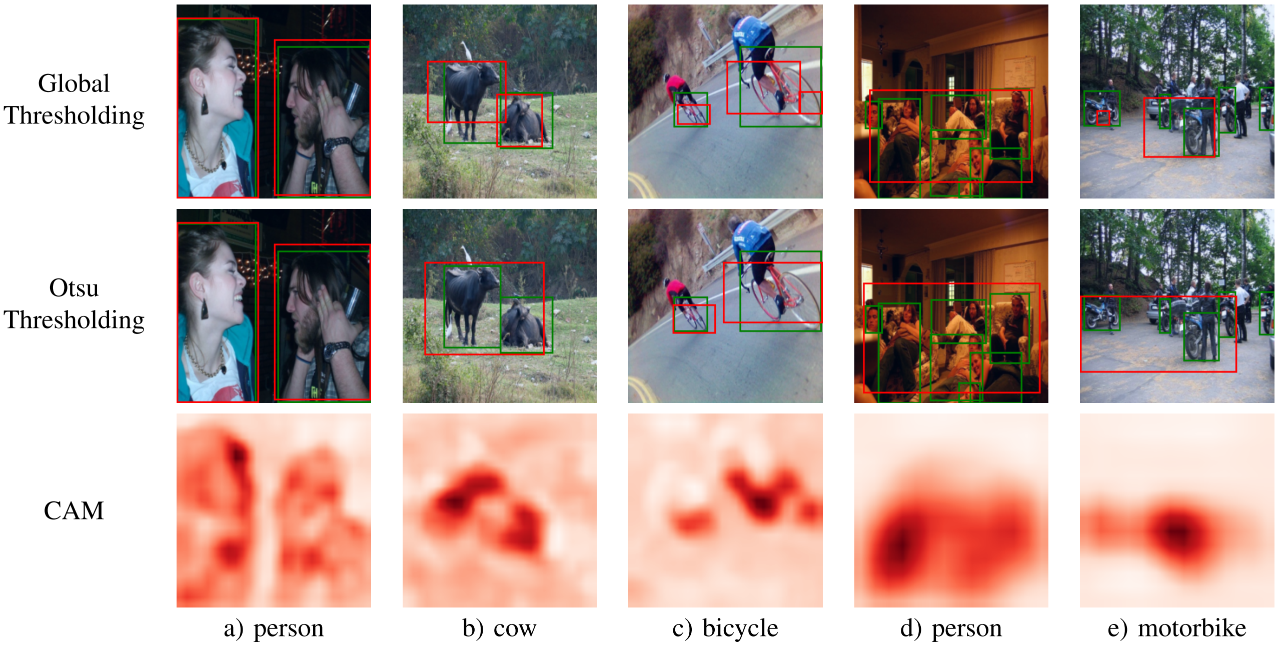
$$\mathcal{L}'_{\text{total}} = \mathcal{L}_{\text{BCE}} + \lambda_{\text{sim}}\mathcal{L}'_{\text{sim}} + \lambda_{\text{norm}}\mathcal{L}'_{\text{norm}} + \lambda_{\text{drop}}\mathcal{L}_{\text{drop}} \tag{16}$$

C. Post processing

1) Threshold method: After accquiring CAM, followed Zhou et al. [1], Kim et al. employed a thresholding technique to segment the CAM, retaining only regions with values exceeding a predefined threshold (e.g., 20%). Then the bounding box covering the contour of the segmented region is extracted as the object localization. However, this thresholding technique is sensitive to the choice of threshold, every CAM may require a different threshold to achieve optimal segmentation. We propose an alternative post-processing step utilizing Otsu's method [12]. This method offers a data-driven approach to threshold selection, potentially leading to more robust CAM segmentation. We apply Gaussian filtering to the CAM to reduce noise before applying Otsu's method. The resulting binary image is then used to extract the bounding box.

To acquire the original threshold method [1] result, we run evaluation on multiple threshold $0 \leq \tau < 1$ and report the highest result. As for Otsu's method, the Gaussian filter requires a kernel size k. We run evaluation on multiple k and report the highest result.

2) Bounding box generation: Kim et al. [4] utilize all contours for bounding box generation. However, in certain scenarios, imperfect CAM segmentation can introduce background regions within the object area. This results in two bounding boxes: one encompassing the outer region's contour and another for the inner region. We propose leveraging the



a) person b) cow c) bicycle d) person e) motorbike

Fig. 2. Comparison between global thresholding, Otsu thresholding. Green boxes denote ground truth, red boxes denote predicted bounding boxes.

TABLE I
CLASSIFICATION RESULT ON VOC 2007

Backbone	Method	Val	Test
ResNet50	CAM [1]	80.09%	81.12%
	Kim et al. [4]	77.86%	77.71%
	CAM [1] + BCE loss (ours)	**84.69%**	**84.33%**
	Kim et al. [4] + BCE loss (ours)	81.92%	81.59%
VGG16	CAM [1]	**77.07%**	**78.00%**
	Kim et al. [4]	75.18%	76.14%
	CAM [1] + BCE loss (ours)	74.95%	75.80%
	Kim et al. [4] + BCE loss (ours)	75.88%	75.89%

TABLE II
CLASSIFICATION RESULT ON VOC 2012

Backbone	Method	Val
ResNet50	CAM [1]	**80.18%**
	Kim et al. [4]	78.50%
	CAM [1] + BCE loss (ours)	71.84%
	Kim et al. [4] + BCE loss (ours)	71.53%
VGG16	CAM [1]	**77.78%**
	Kim et al. [4]	75.66%
	CAM [1] + BCE loss (ours)	76.55%
	Kim et al. [4] + BCE loss (ours)	58.83%

bounding box generated from the external contour to represent the object of interest.

3) Confidence score: Another issue is that confidence score, typically output by object detection models, for each bounding box is not calculated. This lead to difficulty in evaluating the model's performance with COCO mAP [13]. For the time being, we use class probabilities output by model FC layer as confidence score.

V. Experiments

A. Experimental Setup

Dataset. We perform our modifications on two popular multi-label classification datasets: PASCAL VOC 2007 [14] and 2012 [15]. There are 9963 images in VOC 2007 and 11,530 images in VOC 2012 which are splitted 50:50 to train/val and test. Each image contains one or multiple labels from 20 categories.

Metric. We use mAP to evaluate classification performance. mAP is the average of each categories' area under precision and recall curve. For detection, we evaluate using a widely known COCO metric: mAP [13].

TABLE III
HIGHEST MAP ON MULTIPLE CAM THESHOLD ON VOC 2007

Backbone	Method	Val	Test
ResNet50	CAM [1]	6.83%	7.04%
	Kim et al. [4]	6.06%	6.47%
	CAM [1] + BCE loss (ours)	7.66%	7.93%
	Kim et al. [4] + BCE loss (ours)	**8.14%**	**8.40%**
VGG16	CAM [1]	11.54%	11.84%
	Kim et al. [4]	9.97%	10.31%
	CAM [1] + BCE loss (ours)	10.41%	11.56%
	Kim et al. [4] + BCE loss (ours)	**11.61%**	**12.27%**

Implementation Details. We adopt ResNet50 [16] and VGG16 [17] as our backbone. We follow network architecture, hyperparameters setting of the previous work [4]. We resize all images to 224 × 224 before passing to the model. Both backbones are initialized with ImageNet-1k [5] pretrained weights. Each model is trained for 50 epochs with SGD optimizer, learning rate of 0.0001, weight decay of 0.001, momentum of 0.9, and batch size of 16. The implementation is run on Google Colab with Tesla T4 GPU, 16GB of RAM.

TABLE IV
HIGHEST MAP ON MULTIPLE CAM THESHOLD ON VOC 2012

Backbone	Method	Val
ResNet50	CAM [1]	**8.50%**
	Kim et al. [4]	7.90%
	CAM [1] + BCE loss (ours)	7.12%
	Kim et al. [4] + BCE loss (ours)	8.33%
VGG16	CAM [1]	**14.53%**
	Kim et al. [4]	12.80%
	CAM [1] + BCE loss (ours)	12.31%
	Kim et al. [4] + BCE loss (ours)	7.67%

TABLE V
COMPARISON BETWEEN GLOBAL THRESHOLDING AND OTSU THRESHOLDING ON VOC 2007 (VAL)

Backbone	Method	1	7	21	51
ResNet50	CAM [1]	-5.03%	-5.03%	-4.86%	-3.98%
	Bridging [4]	-4.60%	-4.58%	-4.44%	-3.85%
	CAM [1] + BCE loss (ours)	-1.54%	-1.46%	-1.13%	+0.51%
	Bridging [4] + BCE loss (ours)	-1.24%	-1.21%	-0.68%	+1.02%
VGG16	CAM [1]	-1.29%	-1.22%	-1.17%	-1.13%
	Bridging [4]	-1.32%	-1.34%	-1.25%	-1.12%
	CAM [1] + BCE loss (ours)	+0.14%	+0.19%	+0.32%	+0.63%
	Bridging [4] + BCE loss (ours)	-0.91%	-0.83%	-0.68%	-0.65%

TABLE VI
COMPARISON BETWEEN GLOBAL THRESHOLDING AND OTSU THRESHOLDING ON VOC 2007 (TEST)

Backbone	Method	1	7	21	51
ResNet50	CAM [1]	-5.10%	-5.11%	-4.88%	-3.91%
	Bridging [4]	-5.07%	-5.05%	-4.97%	-4.43%
	CAM [1] + BCE loss (ours)	-1.52%	-1.38%	-0.96%	+0.47%
	Bridging [4] + BCE loss (ours)	-0.98%	-0.89%	-0.32%	+1.49%
VGG16	CAM [1]	-1.51%	-1.38%	-1.32%	-1.02%
	Bridging [4]	-1.58%	-1.57%	-1.53%	-1.37%
	CAM [1] + BCE loss (ours)	-0.41%	-0.32%	-0.16%	+0.25%
	Bridging [4] + BCE loss (ours)	-0.48%	-0.38%	-0.35%	-0.09%

TABLE VII
COMPARISON BETWEEN GLOBAL THRESHOLDING AND OTSU THRESHOLDING ON VOC 2012 (VAL)

Backbone	Method	1	7	21	51
ResNet50	CAM [1]	-5.37%	-5.30%	-5.00%	-3.59%
	Bridging [4]	-1.70%	-1.61%	-1.39%	-0.62%
	CAM [1] + BCE loss (ours)	+0.06%	+0.09%	+0.21%	+0.45%
	Bridging [4] + BCE loss (ours)	+0.10%	+0.20%	+0.28%	+0.40%
VGG16	CAM [1]	-1.25%	-1.18%	-1.02%	-0.81%
	Bridging [4]	-1.74%	-1.71%	-1.66%	-1.55%
	CAM [1] + BCE loss (ours)	-0.18%	-0.07%	+0.11%	+0.50%
	Bridging [4] + BCE loss (ours)	-0.61%	-0.56%	-0.54%	-0.40%

B. Results

On VOC 2007 (Tab. I), the proposed multi-label modifications classifies better than both multi-class model from the original works [1] [4] with ResNet50 backbones. However, the performance is lower than the original multi-class model with VGG16 backbone as both multi-class models only predict a single class per image leading to higher precision, thus higher mAP. For object detection (Tab. III), the proposed multi-label modifications show modest improvements over the original methods, but the overall mAP values remain quite low across all methods.

Meanwhile, on the VOC 2012 dataset, the results are even less promising. For classification (Tab. II), the original methods again outperform the proposed modifications, sometimes by a large margin (e.g. VGG16 backbone with Kim et al. method [4]). For object detection on VOC 2012 (Tab. IV), while the proposed ResNet50 modification with Kim et al. method [4] achieved the mAP of 8.33%, it is still very low.

Tab. V, VI, VII compare the performance of global thresholding and Otsu thresholding on VOC 2007 and 2012. Our proposed modifications tend to benefit from using Otsu thresholding instead of global thresholding, especially with the ResNet50 backbone. The improvements are modest but consistent across validation and test sets of both datasets. However, the original single-label methods generally suffer a drop in performance when applying Otsu thresholding.

In Fig. 2, we show the result of our proposed modifications. Examples a, b, and c show that both thresholding methods can detect objects reasonably well. Fig. 2 also shows that each thresholding method contributes to better localizing performance by separating regions of different objects (example b) or merging regions of the same object (example c). However, the result is not as good for objects that are close together (example d) or occluded (example e). In example d, CAM only highlights the region containing a group of people, not each person. This leads to the bounding box encompassing all people in the image. In example e, the proposed modifications predict bounding boxes that do not tightly surround small and occluded motorbikes.

VI. LIMITATION

Our research has produced unsatisfied results in some instances: objects in the same class close together, small objects and occlusion. Our experiments have not been successful on images with adjacent objects because the result obtained after binarizing the CAM will produce a region encompassing all objects. This is a drawback of binarizing the CAM and identifying contours for small objects or obscured objects. We haven't looked into these situations or offered any solutions because of time constraints. These drawbacks lead us to conclude that contour detection and binarization of the CAM are insufficiently efficient methods for identifying objects in a picture.

VII. CONCLUSIONS

In this paper, we proposed modifications to the CAM-based method by Kim et al. [4] to enable weakly supervised object detection for multiple objects in an image. Our key contributions were: 1) Extending the CAM framework to multi-label classification by using a sigmoid activation and binary cross-entropy loss, and 2) Experimenting with Otsu's adaptive thresholding method [12] as an alternative post-processing step to the fixed global thresholding used in previous works.

Our experiments on the PASCAL VOC 2007 and 2012 datasets demonstrated that the proposed multi-label modifications generally improved object detection performance over the original single-label CAM and Kim et al. methods, especially when using the ResNet50 backbone. However, the overall mAP values remained relatively low across all methods evaluated. We also found that our multi-label variants tended to benefit from using Otsu thresholding, achieving modest but consistent improvements compared to global thresholding.

In the future, we plan to explore more advanced post-processing techniques to improve the accuracy of object localization. We also aim to investigate the use of more sophisticated network architectures and loss functions to further enhance the performance of weakly supervised object detection models.

REFERENCES

[1] B. Zhou, A. Khosla, L. A., A. Oliva, and A. Torralba, “Learning Deep Features for Discriminative Localization.” *CVPR*, 2016.

[2] G. Jocher, A. Chaurasia, and J. Qiu, “YOLO by Ultralytics,” Jan. 2023. [Online]. Available: https://github.com/ultralytics/ultralytics

[3] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *Advances in neural information processing systems*, vol. 28, 2015.

[4] E. Kim, S. Kim, J. Lee, H. Kim, and S. Yoon, “Bridging the gap between classification and localization for weakly supervised object localization,” 2022.

[5] 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,” *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015.

[6] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, “The caltech-ucsd birds-200-2011 dataset,” California Institute of Technology, Tech. Rep. CNS-TR-2011-001, 2011.

[7] S. Yang, Y. Kim, Y. Kim, and C. Kim, “Combinational class activation maps for weakly supervised object localization,” 2019.

[8] H. Bilen and A. Vedaldi, “Weakly supervised deep detection networks,” *CoRR*, vol. abs/1511.02853, 2015. [Online]. Available: http://arxiv.org/abs/1511.02853

[9] P. Tang, X. Wang, X. Bai, and W. Liu, “Multiple instance detection network with online instance classifier refinement,” 2017.

[10] Y. Wang, R. Guerrero, and V. Pavlovic, “D2df2wod: Learning object proposals for weakly-supervised object detection via progressive domain adaptation,” 2022.

[11] Z. Huang, Y. Bao, B. Dong, E. Zhou, and W. Zuo, “W2n:switching from weak supervision to noisy supervision for object detection,” 2022.

[12] N. Otsu, “A threshold selection method from gray-level histograms,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[13] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollár, “Microsoft coco: Common objects in context,” 2015.

[14] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, “The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results,” 2007. [Online]. Available: http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html

[15] ——, “The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results,” 2012. [Online]. Available: http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html

[16] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[17] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2015.