A Robust Context-Aware Proposal Refinement Method for Weakly Supervised Object Detection

MEHWISH AWAN© AND JITAE SHIN©, (Member, IEEE)
College of Information and Communication Engineering, Sungkyunkwan University, Suwon 16419, South Korea
Corresponding author: Jitae Shin (jtshin@skku.edu)

This work was supported in part by the National Research Foundation of Korea (NRF) Grant funded by the Korean Government (MSIT) under Grant 2020R1F1A1065626, and in part by the Ministry of Science and ICT (MSIT), South Korea, through the Information Technology Research Center (ITRC) support program supervised by the Institute for Information & Communications Technology Promotion (IITP) under Grant IITP-2020-2018-0-01798.

ABSTRACT
Supervised object detection models require fully annotated data for training the network. However, labeling large datasets is a very time-consuming task, therefore, weakly supervised object detection (WSOD) is a substitute approach to fully supervised learning for the object detection task. Many methods have been proposed for WSOD to date, their performance is still lower than supervised approaches since WSOD is a very challenging task. The major problem with existing WSOD methods is partial object detection and false detection in an objects cluster with the same category. The majority of the methods on WSOD follow multiple instance learning approaches, which does not guarantee the completeness of detected objects. To address these issues, we propose a three-fold refinement strategy to proposals to learn complete instances. We generate class-specific localization maps by fused class activation maps obtained from fused complementary classification networks. These localization maps are used to amend the detected proposals from the instance classification branch (detection network). Deep reinforcement learning networks are proposed to learn decisive-agent and rectifying-agent based on policy gradient algorithm to further refine the proposals. The refined bounding boxes are then fed to instance classification network. The refinement operations result in learning complete objects and greatly improve detection performance. Experimental results show better detection performance by the proposed WSOD method compared to the state-of-the-art methods on PASCAL VOC2007 and VOC2012 benchmarks.

INDEX TERMS
Weakly supervised object detection, complementary learning, discriminative features, proposal refinement, class activation maps, reinforcement learning, and deep learning.

I. INTRODUCTION
Weakly supervised object detection (WSOD) has acquired enormous attention in the literature due to its great ease of demanding only image-level annotated data for training object detector. This has been made possible by the development of convolutional neural networks (CNNs) [1] and large-scale datasets [2] with at least image-level annotations. In this paper, we aim to effectively learn and infer whole object detections trained with coarse image-level labels indicating the particular categories of objects present in an image for the WSOD problem.

Earlier methods [3]–[5] follow conventional multiple instance learning (MIL) paradigm for WSOD task. In MIL framework, images are treated as a set of positive and negative instances bags, and classifier utilizes these instances set as labels. High confidence proposals can be extracted by applying MIL, which is also a suitable solution to localize objects with image-level labels. However, MIL learns the most discriminative part of the target object instead of the complete object [6]. Moreover, MIL has certain constraints, such as positive bags contains at least one positive instance and the negative bag contains all negative instances. Another major drawback of MIL is the most likely positives are predicted using existing classifier which can result in faulty learning in case of false-positive predictions, as classifier explicitly cannot deduct true positives in the given image [7].

Notable progress has been made in WSOD with the advancement of CNNs, since many methods combine MIL and CNNs [8], [9]. Recent studies have revealed that even better results for WSOD can be achieved by training MIL network in the standard end-to-end manner [4], [10], or a...
We adopt the same training approach motivated by [11], [12] a variant of end-to-end training. The approaches which integrate the CNN with MIL for WSOD have shown performance improvement by CNN as feature extractor compared to conventional hand-crafted features [13].

Although a considerable advancement has been accomplished in WSOD literature and these methods have delivered significant results, the existing approaches [6], [13], [14] are less efficient in detecting tight boxes to cover the entire object. In this paper, we propose a framework for WSOD to tackle the main problems with existing WSOD methods, specifically, 1) partial object detection and 2) occluded object detection. This paper is an advanced version based on our previous work [7]. In particular, a three-fold refinement strategy is presented to amend the detected proposals by leveraging the localization maps. The proposed deep network for WSOD consists of three main branches, classification branch for extracting localization information, multiple instance classifier (MIC) (detection branch), and a refinement branch. We present a robust proposal refinement module (PRM) to rectify the proposals to be learned by object detector network by retraining through the instance-level supervisions generated by PRM. This study aims to detect complete and tight bounding boxes around an object instance.

We utilized the class activation maps (CAMs) to obtain localization maps from the fused complementary network (FuCN) [7], which is a classification network. A CAM signifies the distinctive image areas which CNN utilizes to classify an object as an instance of a particular class. Therefore, to cope with the problem of partial object detection and loose detection, we leverage localization maps from FuCN to refine the object detection outputs as shown in Figure 1. For each image localization maps are generated corresponding to each object category. The generated localization maps have the same spatial resolution as of the input image.

In PRM, proposals are first refined based on activated regions within the bounding box with respect to the corresponding localization map. The coordinates of the bounding box are adjusted according to the activated region (pixels). This step compacts the loose proposals, and then further fine-tuning of bounding box is performed by the surrounding region to expand the too-tight proposals which do not cover the complete object, thus attains a complete and tight bounding box for each instance using within the bounding box and the surrounding region. Afterward, the activations for each refined proposal are erased from localization map iteratively. In the second step, missing detections are inferred if there exists any closed region (regions) of activations in the localization map after refining all proposals. These refinement operations result in learning complete objects and improves detection performance. Concisely, the proposed PRM first refines the detections and then investigates the detection completeness for all instances in the entire image by phase I and phase II, respectively.

As summary, our contributions are listed as follows:

1) We present a three-fold refinement procedure after the MIC branch to learn complete instances in the PRM branch. We leverage the localization maps as contextual information for proposal refinement.
2) In the proposed PRM, first, we perform refinement within bounding boxes by the activated pixels in the localization region, then additional fine-tuning is achieved by investigating the fixed neighborhood region of bounding boxes. Lastly, we leverage the connected localization regions to infer missing detections.
3) In phase I of PRM, we determine the precise association between bounding boxes and localization regions by position-aware and class-specific localization maps corresponding to bounding boxes, and elucidate the conflicting regions encompassed by multiple boxes.
4) We propose a cascade of efficient light-weight reinforcement learning (RL) networks based on policy gradient method to further learn complete proposals by training decisive-agent and rectifying-agent concurrently. Lastly, the object detector is retrained based on these refined proposals as instance-level supervision.
5) We evaluate our method on PASCAL VOC2007 and VOC2012 datasets in terms of average precision (AP) and correct localization (CorLoc) to measure accuracy and inference time in frames per second (FPS).

The rest of the paper is organized as follows. Section II is devoted to related work. In Section III we present our proposed method. Section IV presents the experimental results and discussion. Finally, in Section V we conclude the paper.

II. RELATED WORK

WSOD problem has been investigated since last decade, yet the performance of WSOD methods is not close to fully supervised object detection methods. The majority of existing works in WSOD conform to MIL as a baseline framework [3], [11]. However, MIL is non-convex optimization problem [11] and also suffers from the erroneous learning process due to predicted false positives. Several studies have proposed to regularize optimization by reforming the MIL initialization strategy. Cinbis et al. [8] studied a multi-fold MIL strategy, this method was effective in averting the performance collapse in object localization.

Recent approaches combine MIL with CNNs [4], [10], [11], [13]. Bilen et al. [11] proposed an end-to-end framework, weakly supervised deep detection network (WSDDN) for WSOD. Their method is also an adaptation of MIL approach. The authors used the set of pre-computed proposals [15] to obtain candidate boxes that may contain objects, perform feature extraction of these proposals, and classify each proposal. The spatial regularizer is employed for improving performance. Sangineto et al. [16] proposed self-paced learning approach and trained with Fast RCNN [17]. During network training, the same network at different progression stages is used to predict object localization of positive samples. At each stage, a subset of images is selected whose pseudo-ground truth is the most reliable. Online instance classifier refinement (OICR) [4] is proposed to refine the instances in a fully supervised approach on results from WSDDN [11].

Some methods [18], [19] aimed at a proposal-free framework by utilizing deep features. Tang et al. [18] proposed a two-stage region proposal network for WSOD. The authors focused on proposals generation within-end-to-end framework by exploiting the deep feature maps. The hidden object location information generated by early CNNs layers is utilized to generate the proposals and refine the proposals by region-based CNN classifier. Shen et al. [19] proposed an end-to-end WSOD network with generative adversarial learning approach, they used single shot multibox detector (SSD) [20] as baseline detector. Cheng et al. [21] merge selective search [15] and a gradient-weighted CAM based technique to produce proposals that can better enclose the whole objects. An adversarial erasing (AE) approach is proposed by Wei et al. [22] to localize integral object regions. The authors trained the accompanying classification networks with partially erased discriminative object regions from input images. Zhang et al. [23] employed adversarial learning approach motivated by [22], they trained two classifiers to learn distinct features. This strategy boosts object localization performance. We adapt [23] to learn integral object localization cues intended for proposal refinement. Jie et al. [24] proposed a self-taught learning approach for learning object location evidences to train detector. The detector is progressively learned to localize positive samples.

Tang et al. [4] presented multiple instance classifiers approach to learn the features of the whole object iteratively. They performed the instance refinement by multiple supervised stages. However, in their refinement procedure, high confidence proposals neighbouring (intersection-over-union (IoU) based) are considered with the highest score proposal or by using graphs of top-ranking proposals to make clusters which also contain partial object proposals. This approach can result in learning falsely objects with discriminative parts and ineffective to learn whole object detection, hence this method does not guarantee to predict the whole object. Wei et al. [6] proposed tight box mining strategy with a segmentation context for WSOD. They adapted OICR [4] as their detection branch. This method exploits the segmentation context to evaluate proposals by the aid of segmentation maps per-pixel scores to confine the boxes with object parts and obtain the high-quality boxes to learn instance classifier. Their network has an additional segmentation branch. Different from [6], in our WSOD framework we directly refine the proposals from localization maps to reduce the computations. They [6] trained the segmentation branch by pseudo supervisions of object localization acquired from the classification network. Hitherto, a single classification network learns the discriminative features of the object category and cannot provide good localization cues for supervising the segmentation network. It can be trapped in local minimums, consequently, partial object detection problem also resides in this method as well. Shen et al. [25] studied the multi-task learning for WSOD. They treated object detection together with semantic segmentation as a joint learning problem to overcome the failure patterns of segmentation and object detection which are typically encountered in other MIL based self-enforcement methods [4], [8], [26] trained with single-task learning.

Li et al. [27] studied WSOD jointly with segmentation task in a collaborative loop. They trained each task with the supervision of the accompanying task. Lately, spatial likelihood voting method is proposed by Chen et al. [28] to converge proposal localization for object detection with image-level supervision using multi-task learning. Recently, Zhang et al. [14] proposed region-searching paradigm for WSOD with reinforcement learning approach under weak supervisions. They extracted region correspondence maps to use as pseudo-target regions for training the agent. A teacher-student learning approach through multiple instance self-training is used by Ren et al. [29]. They trained the student network with the pseudo-labels from teacher network.
They also used a dropblock to zero out the discriminative parts of objects, a similar idea to [7].

These methods [27]–[29] have achieved suitable performance, however, these approaches still have the problems of missing detection in case of occluded objects and wrong detections due to objects cluster. Since the training image is decomposed into thousands of proposals, and each approximately correct training instance is flooded with many incorrect training instances. Such weak supervision results in inaccurate predictions as it inevitably involves a lot of uncertainty due to noisy training instances. Therefore, to address the aforementioned limitations in WSOD and to improve the robustness of WSOD, we rectify the proposals predicted by MIC through class-specific localization maps, and retrain the object detection network by optimizing the instance-level objective function to learn the refined instances.

### III. PROPOSED METHOD

We propose a proposal refinement module to the WSOD network in [7] to efficiently tackle the problem of partial object detection, loose detections, and objects cluster detection. In particular, our method not only generate a complete and tight box by revising each predicted bounding box but also infer missing instances (if any) from the corresponding class-specific localization maps. The same procedure is applied to all predicted boxes by the MIC branch. The overall architecture of the proposed approach is shown in Figure 2.

VGG [30] is used as a backbone, which further has three branches, i.e., FuCN, MIC (object detection), and PRM. We utilize the high-level features of classification networks learned in complementary manner to make WSOD network learns the whole object representations. We fused the CAMs from two complementary classifier networks to generate a localization map to cover whole object activation regions using Score-CAM [31]. These localization cues from complementary classification network are used in the PRM. The spatial resolution of localization maps is made identical to the input image by upsampling (interpolation). In PRM, we determine the pairs of bounding boxes and the corresponding activation regions in class-specific localization map by coordinated positions of bounding boxes. First, we perform refinement with the inner region of the bounding box with respect to activated pixels in the associated localization region. Then we again exploit the localization maps for contextual information within the fixed region surrounding the box by scaling it with

### TABLE 1. Notations.

| Notation | Description |
|----------|-------------|
| $I$      | set of input images |
| $Y_i$    | label-vector of $i^{th}$ image |
| $I_c$    | subset of class-specific images |
| $\delta_{IoU}$ | IoU threshold |
| $\delta_{a}$ | activations threshold |
| $x^C$    | score matrix from classification branch |
| $x^D$    | score matrix from detection branch |
| $x_f$    | final score matrix |
| $I_{erase}$ | input image with erase detected regions from phase I |
| $L_c$    | class-specific localization map of $I_c$ |
| $L'_c$   | class-specific localization map of $I_{erase}$ |
| $a_c$    | connected region of activations in $L_c$ |
| $a'_c$   | connected region of activations in $L'_c$ |
| $b_c$    | class-specific bounding box from network $C$ |
| $b'_c$   | class-specific bounding box from phase I |
| $b_e$    | temporary bounding box for $a_c$ |
| $b_{c, e}$ | extended to $b_e$ as temporary bounding box |
| $b_c^f$  | final bounding box from phase II |
| $e$      | episode length |
| $B_c$    | set of class-specific bounding boxes from network $C$ |
| $B'_c$   | set of bounding boxes from phase I |
| $B''_c$  | set of bounding boxes from phase II |
M. Awan, J. Shin: Robust Context-Aware Proposal Refinement Method for Weakly Supervised Object Detection

FIGURE 3. (a) CAM overlay with bounding boxes from MIC, (b) refined bounding boxes by PRM with class activation values ≥ δa.

A. FUSED COMPLEMENTARY NETWORK (FuCN)

FuCN aims to learn the whole object features as well as localization through discriminative but complementary feature learning approach inspired by [22] and [23]. FuCN includes two complementary classifiers trained with distinct inputs. In particular, image features are extracted by pre-trained VGG [30]. Then, we add four additional convolutional layers (convs) followed by a fully connected (FC) layer in both classifiers. Feature erasing is performed by thresholding the discriminative features of network A on its heatmaps, afterward, these regions are removed from the entire feature maps of the input image. Removed regions are replaced by zeros. Network B is then encouraged to learn the complementary features for the target object from these erased feature maps. The CAMs from both classifiers are extracted and fused to obtain complete localization maps of objects for each class separately. We upsample these localization maps to the same resolution as the image by interpolation. This network is optimized while training with binary cross entropy (BCE) loss function.

B. DETECTION NETWORK (MIC)

Instead of focusing only on discriminative features which results in partial object detection, in the object detection network features are concatenated with complementary features from FuCN [7] to learn the whole object features. The object detection network has two further branches for classification and detection as in [11]. A set of about 2000 region proposals from selective search [15] is employed in both the training and test phase. Single-level spatial pyramid pooling (SPP) [32] is then applied to produce fixed-size feature maps for these different sized proposals. The discriminative feature learning approach used in [7] drives the network to learn the whole objects and results in improved detections. For image-level classification, we used class-specific max-pooling of regions score from classification branch only.

C. PROPOSAL REFINEMENT MODULE (PRM)

PRM is proposed as a rectification scheme to aim at learning complete and tight object detections. Although whole objects can be learned by object detector as in [7] through concatenated features from FuCN branch, yet in some cases, it can result in loose bounding boxes. Therefore, further refinement is required to obtain tight bounding boxes. We use spatial information within the input image to achieve the refined tight and complete proposals with only image-level supervision. PRM consists of two refinement phases, 1) In phase I, pixel activations from localization maps are leveraged to rectify the detected bounding boxes, 2) phase II includes two policy networks trained with reinforcement learning for proposal refinement. The overall flow of PRM with phase I and phase II rectification is demonstrated in Figure 4. In subsequent subsections, each module is described in detail.
1) PHASE I: PROPOSAL INWARD AND OUTWARD CAM BASED REFINEMENT

The spatial resolution of localization maps obtained from FuCN is made identical to the input image by upsampling (interpolation). We select the pixels from class-specific localization maps with values greater than or equal to $\delta_a$ as foreground objects. Inferring the precise association between bounding boxes and localization regions is very critical especially in case of multiple instances of the same objects in an image. The association between bounding boxes and the activated localization maps is based on the position of bounding box coordinates and corresponding position in the localization map. Bounding box coordinates are transformed according to localization map activations within the box and also to a fixed proportion of its surrounding context ($s_c$).

Algorithm 1 shows the proposal rectification procedure.

As a first step, we check whether the activated pixels ($a_e$) in localization map ($L_c$) each belong to a single bounding box or multiple bounding boxes. If $a_e$ exist in multiple bounding boxes, we call these pixels ‘conflicting pixels’. Afterward, we separate the conflicting pixels from non-conflicting pixels. We calculate the area of bounding boxes which contains conflicting pixels. Then, we assign conflicting pixels to the smallest bounding box to ensure the compactness. When a pixel is assigned to a particular bounding box, we change its activation to zero in the localization map for other proposals during inward refinement. Consequently, there remains no incongruity while associating the activated pixels in the map to bounding boxes. Inward refinement involves the adjustment of region proposal according to the closed activated pixels in the corresponding localization map using image coordinates. Lastly, outward refinement is performed if region proposal has activations in its neighboring region $s_c$.

2) PHASE II: REINFORCEMENT LEARNING NETWORKS TOWARDS COMPLETE REGIONS

We design a cascade of policy networks to learn whole objects under weakly supervised detection settings. Since there are no GT bounding boxes in WSOD, consequently, regression cannot be applied. We, therefore, train RL agents to further rectify the proposals in the weakly supervised setting. To the best of our knowledge, this is the first effort to apply RL based policy algorithms to learn the actions for complete or incomplete proposals and then further rectify proposals with contextual connections for WSOD. The overall procedure of phase II rectification is summarized in Algorithm 2. In subsequent
subsections, we describe the decisive and rectifying agents for WSOD.

\textbf{a: DECISIVE NETWORK}

The objective of decisive-agent for WSOD is to learn the actions for input images as complete or incomplete object detections. We mask out all detected bounding box regions with zeroes correspondingly in input image feature maps. The decisive-agent is designed to discover the undetected object regions in the mask-out image to further improve detections of phase I. For an image if the decision of decisive-agent is incomplete, then the rectifying network decides about the detected region as a new bounding box (i.e., a missing detection from preceding detection procedure) or part of the existing proposals.

A policy gradient reinforcement strategy is adopted to learn policy function to maximize the expected reward. To input to decisive network, we mask out all detected bounding box regions obtained from phase I with zeroes in input image feature maps. The policy network predicts complete and incomplete action likelihood for the mask-out image ($I_{\text{erase}}$). This method corresponds as the detections are complete or there exists some missing objects or parts undetected.

The state is the input image features with the position-aware masked-out feature maps of detected regions from phase I. Action to the current state is the binary decision ($d \in \{\text{incomplete, complete}\}$). Decisive-agent inspects whether the mask-out image further have any undetected object(s) regarded as incomplete detections, or detections are complete such that no remaining activated regions are found in the masked-out image (complete detections). The policy gradient method directly models and optimizes the policy ($\pi_\theta$) with respect to parameters ($\theta$), $\pi_\theta(a|s)$ ($a$ is action and $s$ is state) to learn actions for the states.

The proposed decisive network consists of 3 FC layers followed by a softmax layer as shown in Figure 2. We use 256+1 (+1 is for additional input for action history), and 128 sized vectors for first FC and hidden FC layers, respectively. To avoid catastrophic forgetting, we input the agent’s experience replay $[(s, a, r), \ldots, (s_t, a_t, r_t)]$ to the last FC layer.

\textbf{REWARD FUNCTION:} In RL, designing an appropriate reward function for the specific decision problem is of great importance. The reward function $\{r \in (-1, +1)\}$ for decisive-agent is defined based on the confidence score ($s_{\text{erase}}$) of mask-out image from a maximum of both classifiers (classifier $A$ and $B$). The agent learns the correct decisions by positive and negative rewards given for particular state as defined below in (1) and (2):

\begin{align}
 r_{\text{complete}} &= \begin{cases} +1, & \text{if } s_{\text{erase}} = 0 \\ -1, & \text{else} \end{cases} \quad (1) \\
 r_{\text{incomplete}} &= \begin{cases} +1, & \text{if } s_{\text{erase}} > 0 \\ -1, & \text{else} \end{cases} \quad (2)
\end{align}

\textbf{b: RECTIFYING NETWORK}

We propose a context-aware approach by training the RL agent to amend the region proposals ($B'_c$ from phase I) based on the contextual connections of bounding boxes with CAMs ($L'_c$). The proposals neighboring to the CAMs ($a'_1, a'_2, \ldots, a'_n$) in the mask-out image are investigated and each bounding box ($b'_c$) is modified according to the decision of rectifying-agent. The architectural design of rectifying network is same as decisive network (Figure 2).
Closed activated regions \( (a'_c) \) from localization map \( (L'_c) \) are computed as activated values within threshold \( (\delta_a) \) with 8-connected neighbors. We define the criteria to decide about the discovered activated regions \( (a'_c) \) by the decisive network. This mainly includes dual considerations: 1) Is \( a'_c \) a missing detection or part of an already detected proposal (phase I detected proposals), 2) explore the neighboring context of each \( a'_c \) \( (a'_c \in L'_c) \) to determine action for it.

We construct a temporary bounding box \( (b_t) \) around each \( a'_c \) as a closed foreground region based on activated locations as thresholded regions. Then we perform a distance measure between each \( b_t \) and \( b'_c \). As a first step, we calculate the IoU between \( b'_c \) and \( b'_t \) \( (b'_{ct} \) is temporary or hypothetical bounding box generated as \( b'_c \) extended to \( b_t \)). If IoU between \( b'_c \) and \( b'_{ct} \) is greater or equal to \( \delta_{IoU} \) then further pixel-based distance measure comparison is performed.

Additional spatial information is fed to the network by computing spatial relations within an image by exploring the neighboring context of activated areas in \( L'_c \). We calculate the connectivity of each \( a'_c \) with \( a_c \) in detected proposals \( (B'_c) \) as a distance measure between both activated regions. The pixel distance is computed between activated regions within the bounding boxes \( b_t \) and \( b'_c \) by the most upper-left, upper-right, bottom-left, and bottom-right pixel location of the activated regions. The pixel distance of each outermost pixel in \( a'_c \) is computed to all four outermost pixels of the neighboring region \( a_c \) to estimate the nearest region.

The neighboring bounding box with the least distance to \( b_t \) is selected to modify with \( b'_c \). However, if IoU between \( b'_c \) and all \( b'_{ct} \) is less than \( \delta_{IoU} \), the particular temporary box is considered as a separate new bounding box that was a missed detection. For training the rectifying-agent, these two-level contextual relationship details between the bounding boxes \( b_t \) and \( b'_c \) are fed to the network.

The color smoothness information between each \( b_t \) and their neighboring proposals is also fed to the rectifying network. Color smoothness matching further guides the agent about the degree of similarity between two regions. To compute smoothness over the color image regions, the Laplacian operator (second derivative along pixels) is applied on each channel and then take an average for all three channels. For each region proposal, smoothness information is feed to the network.

We feed the rectifying network with the following state information, 1) convolutional features, 2) fused CAMs \( (L'_c) \) from both classifiers, 3) rectified proposals \( (B'_c) \) with scores, and 4) contextual details. Action history along with \( \alpha \) (tolerance coefficient) history for the entire image including all bounding boxes rectification actions is also fed to the network. The transformations to the bounding box(es) are applied according to the action of rectifying-agent w.r.t. \( L'_c \). Figure 6 presents the procedure of rectification network.

REWARD FUNCTION: we define reward function for rectifying-agent by computing the overall score \( (h) \) for amended boxes \( (B'_c \) and \( B''_c) \) with high weightage to entropy score than category confidence score. The score \( (h(b'_c)) \) of the amended bounding box \( b''_c \) (phase II) is compared to the score \( (h(b'_c)) \) of the previous same bounding box \( b'_c \) (phase I). If a large gain in the score is calculated then the decision of rectifying-agent is considered as wrong and negative reward is given and vice versa. A little gain in \( h(b''_c) \) is considered tolerable. For rectifying-agent rewards are defined in (3) and (4) as follows;

\[
\begin{align*}
    r &= -1, & \text{if } & h(b''_c) \neq \alpha \ast h(b'_c) \\
    r &= +1, & \text{if } & h(b''_c) == \alpha \ast h(b'_c)
\end{align*}
\]

\( \alpha \) is the tolerance coefficient that is learned by the rectifying-agent. To optimize the network performance \( \alpha \) coefficient is learned by the rectifying-agent. \( \alpha \) value is
updated according to learning rate based on the reward for the current action given a particular state, with current \( \alpha \) value, and \( \alpha \) history with action history. Therefore rectifying-agent is trained based on reinforcement learning as well as self-learning approach at the same time.

### D. TRAINING PROPOSED WSOD FRAMEWORK

Overall training of the proposed WSOD method is performed epoch-wise. Once the network is trained for all epochs including all episodes for policy networks, the whole network is trained again with continuous rewards for each image regardless of object categories groups. After completing the network training for each epoch MIC network is retrained to learn whole instances.

1) TRAINING DETECTION NETWORK (MIC)

After feature extraction from pretrained VGG [30], the network is branched into three further networks, \( A, B, \) and \( C \). Network \( A \) and network \( B \) extract discriminative features which are then concatenated for input to network \( C \) (MIC network).

For \( N \) multi-label images in trainval set, label-vector for \( i \)th image is \( y_i = \{y_{i1}, y_{i2}, \ldots, y_{iC}\} \), where \( y_{ij} = 1(j = 1, \ldots, C) \), if \( j \)th class object is present in image and \( y_{ij} = 0 \) otherwise, and \( C \) is the total number of categories. For detection network, image-level classification scores are computed by class-specific max-pooling on all regions \( R, R = (r_1, r_2, \ldots, r_T) \), here \( T \) is the total number of proposals. For \( i \)th image, the \( j \)th class score is calculated by maximum of all regions \( j \)th class softmax probabilities, \( p_{ij} = \max(p_{r1}', p_{r2}', \ldots, p_{rT}') \). Thus, the prediction vector for \( i \)th image is computed as \( p_i = [p_{i1}, p_{i2}, \ldots, p_{iC}] \).

We use BCE loss function with stochastic gradient descent with momentum 0.9 and weight decay \( 5 \times 10^{-4} \) for optimization. The overall loss function for the network is defined in (5) as;

\[
L = L_A + L_B + L_C
\]

\[
L_A = L_B = - \sum_{j=1}^{C} (y_{ij} \cdot \log (p_{ij}) + (1-y_{ij}) \cdot \log (p_{ij}))
\]

\[
L_C = - \sum_{j=1}^{C} (y_{ij} \cdot \log (p_{ij}) + (1-y_{ij}) \cdot \log (p_{ij})) - \sum_{s=1}^{s} (p_s \cdot \log (p_s))
\]

where, \( L \) is the loss function for WSOD network [7], \( L_A \) and \( L_B \) are the loss functions of networks \( A \) and \( B \), respectively. \( L_C \) is the loss function of MIC network. In (7), \( s \) is the individual state in the probability distribution (\( s \in S \)), where \( S \) is the number of discrete states.

2) TRAINING POLICY NETWORKS

Training settings for both decisive and rectifying networks are same except reward functions. We set episode length to the number of images in the entire subset of class-specific images (\( e = I_c \)) for training the network on PASCAL VOC2007 dataset. Whereas, for PASCAL VOC2012 episode length is half of each class-specific image subset (\( e = I_c/2 \)). For rectifying-agent, episode length is dynamic depending on the number of discovered instances in each image.

For training policy networks, random images with same category are used in an episode. softmax in action preferences is used for policy parameterization such that the actions with the highest preferences in each state are given the highest probabilities of being selected [35]. Furthermore, parameterizing policies according to the softmax in action preferences with discrete action space enables optimizing the approximate policy as a deterministic policy which is imperative for WSOD scenario.

We use gradient ascend with RMSprop [36] optimizer for policy optimization with same learning rate as in detection network. The objective function \( J(\theta) \) for each episode with a set of parameters (\( \theta \)), current state at time \( t \) (\( s_t \)) and action \( (a_t) \) with discount factor (\( \gamma = 1.0 \)) using Monte-Carlo policy differentiation with reinforcement method is defined in (8) as;

\[
J(\theta) \approx \sum_{t}^{e} \gamma \cdot r(s_t, a_t)
\]

We use 0.4 dropout for the last hidden layer during training as regularization to avoid overfitting. For each epoch, 20 separate episodes are executed corresponding to each PASCAL VOC category.

### IV. EXPERIMENTAL RESULTS

#### A. BENCHMARK DATA

The proposed method is evaluated on PASCAL VOC2007 and VOC2012 datasets with 20 object categories which are widely used as benchmarks for object detection. We use trainval sets with 5011 images for VOC2007 and 11540 images for VOC2012 for training. Only image-level labels are used for training. The test sets with 4952 images for VOC2007 and 10991 images for VOC2012 are used for evaluation of the proposed WSOD framework.

#### B. PERFORMANCE EVALUATION METRICS

We use two performance measures for detection. The first metric is AP with 0.5 IoU between detected boxes and ground-truths and mean of AP (mAP). Furthermore, we use CorLoc to evaluate the localization accuracy. Both AP and CorLoc metrics measure the quantitative performance of the object detector based on the PASCAL criteria with IoU \( \geq 0.5 \).

#### C. IMPLEMENTATION DETAILS

VGG16 [30] trained on ImageNet [2] is used as a backbone network. In our experiments, FuCN is trained with two complementary networks with same implementation settings as in [7]. We employ pretrained FuCN to train the proposed network. Score-CAM [31] is used for computing class activations.
TABLE 2. Comparison (AP in %) on PASCAL VOC2007 test set.

| Method          | mAP  |
|-----------------|------|
| Cubis et al. [8]| 39.3 |
| Li et al. [33]  | 49.7 |
| Bielen et al. [11] | 42.9 |
| Tang et al. [4] | 38.0 |
| Tang et al. [13] | 34.4 |
| Zhang et al. [34] | 36.4 |
| Cheng et al. [21] | 65.6 |
| Ren et al. [29] | 79.6 |
| Our MIC+PI     | 62.8 |
| Our MIC+PI-II  | 62.9 |
| Our MIC (retrain)| 68.5 |

The learning rate is set to $10^{-3}$ for the first 30 epochs and then to $10^{-4}$ for the remaining 40 epochs. The number of episodes for training policy networks is same as the number of epochs. We initialize $\alpha$ with 1.3 value for optimizing rectifying-agent.

It is always crucial to select an optimal threshold. Nevertheless, depending on the objective for a specific task at a particular stage we use threshold values in our experiments which yields maximum performance for that specific task. In our experiments $\delta_a$ is 0.5 for phase I refinement and 0.7 for phase II refinement. We use a more constricted activation threshold for phase II since we aim to extract highly confident but mistakenly unexploited regions. Using low activation values in phase II refinement can cause ambiguity and may result in incorrect detections. Surrounding context $(s_c)$ is set to 1.2 [6] during the refinement of proposals in phase I. The threshold $\delta_{IoU}$ is set to 0.75 in rectifying network. A loose $\delta_{IoU}$ can even wrongly already correctly detected regions instead of improving the detections, and too tight $\delta_{IoU}$ may not aid in rectifying a region even if it has its true part in an activated neighboring region. The overall score $h$ is computed by 0.7 weight to entropy score and 0.3 weight to category confidence score. All experiments are conducted on NVIDIA GeForce TITAN XP 4 parallel GPUs.

**D. COMPARISON WITH STATE-OF-THE-ART METHODS**

We compare the proposed WSOD method with state-of-the-art methods on PASCAL VOC2007 and VOC2012 datasets. We reported experimental results for both PRM phases (PI and PII) separately, and retrained MIC. Table 2 and Table 3 illustrate the comparison of the results on PASCAL VOC2007 dataset in terms of AP, mean AP for test set and CorLoc for trainval set, respectively.

It can be observed that our method with each module outperforms all compared methods for both test and trainval sets. In Table 2, from all compared methods, [29] has the highest mAP (54.9%), however, our method (MIC+PI+PII) has an enormous improvement with 6.7% mAP to [29]. From Table 3, Chen et al. [28] have the highest mean Corloc 71.0% among all other compared methods. In comparison to [28], the proposed method has a significantly high score with 6.0% gain in mean CorLoc.

Table 2 and Table 3 illustrate the results by proposed and compared methods on PASCAL VOC2012 test and trainval sets in terms of mAP and CorLoc, respectively. A similar...
TABLE 4. Comparison (AP in %) on PASCAL VOC2012 test set.

| Method            | aero | bike | bird | boat | bottle | bus | car  | cat  | chair | cow  | table | dog  | horse | mbike | person | plant | sheep | sofa  | train | tv  | mAP  |
|-------------------|------|------|------|------|--------|-----|------|------|------|------|------|------|-------|--------|-------|-------|-------|-------|-----|------|
| Cheng et al. [21] | 68.3 | 60.0 | 47.4 | 26.4 | 20.6   | 61.5| 59.9 | 82.1 | 23.7 | 20.1 | 78.8 | 52.7 | 67.7  | 2.6   | 21.5  | 43.8  | 50.1  | 67.2 | 60.5 | 48.3 |
| Chen et al. [28]  | -    | -    | -    | -    | -      | -   | -    | -    | -    | -    | -    | -    | -     | -     | -     | -     | -     | -    |    | 49.2 |
| Ren et al. [29]   | -    | -    | -    | -    | -      | -   | -    | -    | -    | -    | -    | -    | -     | -     | -     | -     | -     | -    |    | 52.1 |
| Tang et al. [13]  | 58.2 | 66   | 41.8 | 24.8 | 27.2   | 55.7| 55.2 | 28.5 | 16.6 | 51   | 17.5 | 28.6 | 49.7  | 70.5  | 7.1   | 25.7  | 47.5  | 36.6 | 44.1 | 59.2 | 40.6 |
| Li et al. [27]    | -    | -    | -    | -    | -      | -   | -    | -    | -    | -    | -    | -    | -     | -     | -     | -     | -     | -    | -   | 43.5 |
| Our MIC+PI        | 56.5 | 65.1 | 44.3 | 32.8 | 26.1   | 67.9| 68.3 | 66.3 | 25.7 | 48.7 | 49.1 | 59.5 | 48.7  | 70.4  | 43.1  | 54.8  | 45.8  | 56.3 | 55.7 | 49.3 | 51.7 |
| Our MIC+PI+PII    | 61.1 | 67.2 | 50.8 | 42.4 | 30.5   | 69.4| 35.3 | 55.7 | 53.2 | 62.9 | 52.3 | 71.8 | 48.3  | 57.2  | 51.1  | 58.4  | 57.1  | 53.6 | 55.9 | -    |      |
| Our MIC (retrain) | 60.8 | 66.4 | 49.3 | 43.5 | 31.2   | 68.7| 68.2 | 67.8 | 33.8 | 32.5 | 51.9 | 61.3 | 51.1  | 70.7  | 47.2  | 55.6  | 50.5  | 57.2 | 53.5 | 52.8 | 54.8 |

TABLE 5. Comparison (CorLoc in %) on PASCAL VOC2012 trainval set.

| Method            | aero | bike | bird | boat | bottle | bus | car  | cat  | chair | cow  | table | dog  | horse | mbike | person | plant | sheep | sofa  | train | tv  | mAP  |
|-------------------|------|------|------|------|--------|-----|------|------|------|------|------|------|-------|--------|-------|-------|-------|-------|-----|------|
| Cheng et al. [21] | 85.5 | 81.1 | 69.2 | 54.3 | 37.6   | 86.7| 81.7 | 84.0 | 44.6 | 83.3 | 45.8 | 80.2 | 84.2  | 87.2  | 11.5  | 52.1  | 78.9  | 63.9 | 81.0 | 80.9 | 68.7 |
| Chen et al. [28]  | -    | -    | -    | -    | -      | -   | -    | -    | -    | -    | -    | -    | -     | -     | -     | -     | -     | -    | -   | 69.2 |
| Ren et al. [29]   | -    | -    | -    | -    | -      | -   | -    | -    | -    | -    | -    | -    | -     | -     | -     | -     | -     | -    | -   | 70.9 |
| Tang et al. [13]  | 77.2 | 83   | 62.1 | 55   | 49.3   | 83  | 75.8 | 37.7 | 43.2 | 81.6 | 46.8 | 42.9 | 73.3  | 90.3  | 21.4  | 56.7  | 84.4  | 55   | 62.9 | 82.5 | 63.2 |
| Li et al. [27]    | -    | -    | -    | -    | -      | -   | -    | -    | -    | -    | -    | -    | -     | -     | -     | -     | -     | -    | -   | 67.9 |
| Our MIC+PI        | 78.8 | 86.6 | 67.2 | 58.2 | 51.3   | 80.0| 78.2 | 77.5 | 55.1 | 79.7 | 59.8 | 70.6 | 78.4  | 89.4  | 65.2  | 64.4  | 85.6  | 66.2 | 66.9 | 78.9 | 71.9 |
| Our MIC+PI+PII    | 79.1 | 87.3 | 69.8 | 59.2 | 53.7   | 83.4| 79.6 | 79.8 | 55.7 | 82.9 | 60.6 | 72.7 | 79.7  | 89.8  | 68.5  | 64.5  | 85.7  | 68.3 | 67.7 | 79.6 | 73.4 |
| Our MIC (retrain) | 78.8 | 86.8 | 68.2 | 59.0 | 52.2   | 81.5| 78.8 | 77.8 | 55.3 | 81.8 | 59.9 | 71.8 | 81.1  | 88.7  | 67.1  | 64.1  | 85.5  | 66.7 | 67.3 | 78.8 | 72.4 |

FIGURE 7. Class-specific detection results by the proposed method with final localization maps.

performance pattern as PASCAL VOC2007 is observed for PASCAL VOC2012 on both test and trainval sets. All modules of the proposed method have surpassed all compared methods with a substantial performance boost.

Our method has numerous advantages compared to all previous state-of-the-art methods. Unlike [28], we investigate the outside neighboring region of the detected box but on a fixed scale, which avoids our method to erroneously include activated regions of the same category other than the specific instance. SLV approach [28] wrongly detect whole objects cluster with same category as a single detection and individual instances remained undetected. Partial object detection problem also still occur in [29] and [28]. Furthermore, mostly WSOD methods have declined performance on classes with relatively smaller size objects particularly “bottle” class. It is also observed that “chair” class has lower AP by all compared methods. The results in Table 2 and Table 3 illustrate that our method has achieved significantly improved AP for these categories compared to state-of-the-art methods. Qualitative results (Section IV: F) have further verified that small object detection is significantly improved by phase II refinement which is mostly missed in many other WSOD methods [21], [28], [29].

Overall the proposed PRM and MIC retraining results in enhanced performance for WSOD. CAM based dual-step rectification and complete or incomplete detection investigation supervise the WSOD network to learn high-quality, refined, tight, and complete regions. Furthermore, RL lightweight policy networks have an imperative contribution in enhancing the detection performance by further discovering missed instances.

E. INFERENCE TIME

We report the inference time of our method incrementally for each module with VGG16 [30] backbone network in Table 6. The inference time for our method (MIC) is same as [7] with much improved detections. Since PRM involves a series of refinement stages the inference time for phase I is only 0.70 FPS including MIC branch, and inference time for phase II including MIC and phase I is 0.67 FPS for PASCAL VOC2007. Phase I with MIC has inference time 0.82 FPS, and phase II with MIC and phase I takes 0.79 FPS for PASCAL VOC2012 dataset.
TABLE 6. Comparison of inference time and accuracy for each module incrementally in terms of FPS and mAP (%), respectively on PASCAL VOC2007 and VOC2012 test sets.

| Method | VOC2007 |  | VOC2012 |  |
|--------|---------|--------|---------|--------|
| MIC [7] | PI | PII | MIC (retrain) | mAP | FPS | mAP | FPS |
| ✔ |  |  |  | 52.6 | 0.72 | 45.4 | 0.84 |
| ✔ | ✔ |  |  | 60.3 | 0.70 | 51.7 | 0.82 |
| ✔ | ✔ | ✔ |  | 61.6 | 0.67 | 55.9 | 0.79 |
| ✔ | ✔ | ✔ |  | 60.7 | 0.72 | 54.8 | 0.84 |

Moreover, it is practical to use MIC during inference as it is trained with supervision from PRM and attains high accuracy with reasonable inference time (0.72 FPS on PASCAL VOC2007 and 0.84 FPS on VOC2012 dataset).

F. QUALITATIVE RESULTS
Figure 7 shows the final qualitative detection and localization results of the proposed method. Our proposed method has overcome the problems of 1) partial object detection, 2) same category object cluster detection, 3) loose detections, and 4) missing detections effectively. However, there are certain failure cases for some images as shown in Figure 8. These failure cases are mainly due to images with less effective initial proposals because of high similarity in texture or color between neighboring instances, or due to occlusions.

Duplicate detections for the same object are effectively tackled by our method since phase I refinement procedure avoids the double detections by assigning a particular activated region only to a single bounding box. Very few false detections are observed as objects cluster or partial object detection. Such failure cases can be overcome by employing a more sophisticated proposal generator in the first place. Moreover, the qualitative results in Figure 4, Figure 6, Figure 7, and Figure 8 have shown that small object detection, loose detection, and object cluster detection is significantly improved by proposed PRM. The final detections inferred by the proposed WSOD method have validated substantially better-quality outputs in most cases as presented in Figure 7 and Figure 8.

V. CONCLUSION
This paper proposed an effective framework for WSOD to achieve exhaustive refinement after instance-level classification. Two main rectification stages are proposed, phase I for rectifying the detected regions to obtain close-fitted and whole object bounding boxes. Phase II goals to detect missed instances which are mostly small objects. Since the majority of WSOD methods suffer from problems of partial object detection, objects cluster detection, and loose detections. Therefore, this study has efficaciously accomplished the aim of tight and complete object detections in weakly supervised settings. Qualitative results have demonstrated high-quality detection output by the proposed method on PASCAL VOC2007 and VOC2012 benchmarks. Correspondingly, the quantitative results have shown enhanced performance by the proposed WSOD method compared to the state-of-the-art WSOD methods.

REFERENCES
[1] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” Proc. IEEE, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2009, pp. 248–255.

[3] X. Wang, Z. Zha, C. Yao, and X. Bai, “Relaxed multiple-instance SVM with application to object discovery,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 1224–1232.

[4] P. Tang, X. Wang, X. Bai, and W. Liu, “Multiple instance detection network with online instance classifier refinement,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 2843–2851.

[5] Z. Zhang, S. Qiao, C. Xie, W. Shen, B. Wang, and A. L. Yuille, “Single-shot object detection with enriched semantics,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 5813–5821.

[6] Y. Wei, Z. Shen, B. Cheng, H. Shi, J. Xiong, J. Feng, and T. Huang, “TSC2: Tight box mining with surrounding segmentation context for weakly supervised object detection,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 434–450.

[7] M. Awan and J. Shin, “Weakly supervised object detection using complementary learning and instance clustering,” IEEE Access, vol. 8, pp. 103419–103432, 2020.

[8] R. G. Cinbis, J. Verbeek, and C. Schmid, “Weakly supervised object localization with multi-fold multiple instance learning,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 39, no. 1, pp. 189–203, Jan. 2017.

[9] M. Shi, H. Caesar, and V. Ferrari, “Weakly supervised object localization using things and stuff transfer,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 3381–3390.

[10] V. Kantorov, M. Oquab, M. Cho, and I. Laptev, “ContextLocNet: Context-aware deep network models for weakly supervised localization,” in Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer, 2016, pp. 350–365.

[11] H. Bilen and A. Vedaldi, “Weakly supervised deep detection networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 2846–2854.

[12] A. Diba, V. Sharma, A. Pazandeheh, H. Pirsiavash, and L. Van Gool, “Weakly supervised cascaded convolutional networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR). Jul. 2017, pp. 914–922.

[13] P. Tang, X. Wang, S. Bai, W. Shen, X. Bai, W. Liu, and A. Yuille, “PCL: Proposal cluster learning for weakly supervised object detection,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 42, no. 1, pp. 170–191, Jan. 2020.

[14] D. Zhang, J. Han, L. Zhao, and T. Zhao, “From discriminant to complete: Reinforcement searching-agent learning for weakly supervised object detection,” IEEE Trans. Neural Netw. Learn. Syst., early access, Feb. 21, 2020, doi: 10.1109/TNNLS.2020.2969483.

[15] J. R. R. Uijlings, K. E. A. van de Sande, T. Gevers, and A. W. M. Smeulders, “Selective search for object recognition,” Int. J. Comput. Vis., vol. 104, no. 2, pp. 154–171, Sep. 2013.

[16] E. Sangineto, M. Nabi, D. Cublierk, and N. Sebe, “Self paced deep learning for weakly supervised object detection,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 41, no. 3, pp. 712–725, Mar. 2019.

[17] R. Girshick, “Fast R-CNN,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Dec. 2015, pp. 1440–1448.

[18] P. Tang, X. Wang, A. Wang, Y. Yan, W. Liu, J. Huang, and A. Yuille, “Weakly supervised region proposal network and object detection,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2018, pp. 352–368.

[19] Y. Shen, R. Ji, S. Zhang, W. Zuo, and Y. Wang, “Generative adversarial learning towards fast weakly supervised detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 5764–5773.

[20] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “SSD: Single shot multibox detector,” in Proc. Eur. Conf. Comput. Vis. Cham, Switzerland: Springer, 2016, pp. 21–37.

[21] G. Cheng, J. Yang, D. Gao, L. Guo, and J. Han, “High-quality proposals for weakly supervised object detection,” IEEE Trans. Image Process., vol. 29, pp. 5794–5804, 2020.

[22] Y. Wei, J. Feng, X. Liang, M.-M. Cheng, Y. Zhao, and S. Yan, “Object region mining with adversarial erasing: A simple classification to semantic segmentation approach,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 1568–1576.

[23] X. Zhang, Y. Wei, J. Feng, Y. Yang, and T. Huang, “Adversarial complementary learning for weakly supervised object localization,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 1325–1334.

[24] Z. Iie, Y. Wei, X. Jin, J. Feng, and W. Liu, “Deep self-taught learning for weakly supervised object localization,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 1377–1385.

[25] Y. Shen, R. Ji, Y. Wang, Y. Wu, and L. Cao, “Cyclic guidance for weakly supervised joint detection and segmentation,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2019, pp. 697–707.

[26] W. Ren, K. Huang, D. Tao, and T. Tan, “Weakly supervised large scale object localization with multiple instance learning and bag splitting,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 2, pp. 405–416, Feb. 2016.

[27] X. Li, M. Kan, S. Shan, and X. Chen, “Weakly supervised object detection with segmentation collaboration,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 9735–9744.

[28] Z. Chen, Z. Fu, R. Jiang, Y. Chen, and X.-S. Hua, “SLV: Spatial likelihood voting for weakly supervised object detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2020, pp. 12995–13004.

[29] Z. Ren, Z. Yu, X. Yang, Y.-M. Liu, Y.-J. Lee, A. G. Schwing, and J. Kautz, “Instance-aware, context-focused, and memory-efficient weakly supervised object detection,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2020, pp. 10598–10607.

[30] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014, arXiv:1409.1556. [Online]. Available: http://arxiv.org/abs/1409.1556.

[31] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. Mardziel, and X. Hu, “Score-CAM: Score-weighted visual explanations for convolutional neural networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jun. 2020, pp. 24–25.

[32] K. He, X. Zhang, S. Ren, and J. Sun, “Spatial pyramid pooling in deep convolutional networks for visual recognition,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 37, no. 9, pp. 1904–1916, Sep. 2015.

[33] D. Li, J.-B. Huang, Y. Li, S. Wang, and M.-H. Yang, “Weakly supervised object localization with progressive domain adaptation,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 3512–3520.

[34] D. Zhang, J. Han, L. Zhao, and D. Meng, “Leveraging prior-knowledge for weakly supervised object detection under a collaborative self-paced curriculum learning framework,” Int. J. Comput. Vis., vol. 127, no. 4, pp. 363–380, Apr. 2019.

[35] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction. Cambridge, MA, USA: MIT Press, 2018.

[36] G. Hinton, N. Srivastava, and K. Swersky, “Neural networks for machine learning lecture 6a overview of mini-batch gradient descent,” Cited On, vol. 14, no. 8, 2012.