Distilling Knowledge from Refinement in Multiple Instance Detection Networks

Luis Felipe Zeni
luis.zeni@inf.ufrgs.br

Claudio R. Jung
crjung@inf.ufrgs.br

Institute of Informatics, Federal University of Rio Grande do Sul, Brazil

Abstract

Weakly supervised object detection (WSOD) aims to tackle the object detection problem using only labeled image categories as supervision. A common approach used in WSOD to deal with the lack of localization information is Multiple Instance Learning, and in recent years methods started adopting Multiple Instance Detection Networks (MIDN), which allows training in an end-to-end fashion. In general, these methods work by selecting the best instance from a pool of candidates and then aggregating other instances based on similarity. In this work, we claim that carefully selecting the aggregation criteria can considerably improve the accuracy of the learned detector. We start by proposing an additional refinement step to an existing approach (OICR), which we call refinement knowledge distillation. Then, we present an adaptive supervision aggregation function that dynamically changes the aggregation criteria for selecting boxes related to one of the ground-truth classes, background, or even ignored during the generation of each refinement module supervision. Experiments in Pascal VOC 2007 demonstrate that our Knowledge Distillation and smooth aggregation function significantly improves the performance of OICR in the weakly supervised object detection and weakly supervised object localization tasks. These improvements make the Boosted-OICR competitive again versus other state-of-the-art approaches.

1. Introduction

Supervised object detection has been achieving increasingly better results in terms of accuracy and speed along the past years [13, 11]. The main drawback of these methods is the need for annotated bounding boxes, which is a tedious, error-prone, time-consuming, and expensive task. The annotation cost directly impacts the viability of deployment of these detectors in real-world applications, particularly when starting from scratch for a specific application. One approach that researchers are exploring to alleviate the annotation cost is Weakly Supervised Object Detection (WSOD), where the object detector is trained using only image category annotations (presence or absence of interest classes in the image), which is much easier and faster to generate.

Most WSOD methods [2, 1, 19, 5, 18, 23] follow the Multiple Instance Learning (MIL) pipeline [6] to train detectors using only image category level annotations. In the adaptation of MIL to the WSOD task, each image is considered a bag of positive and negative object proposals generated by object proposal methods such as Selective Search [22] or Edge Boxes [27]. The training process in the MIL framework encompasses two steps: (i) to train an instance selector to compute the object score of each object proposal; (ii) to select the proposal with the highest score and use it to mine positive instances and train detector estimators. The majority of recent methods explore features extracted by Convolutional Neural Networks (CNN) as an off-the-shelf feature extractor [2, 10] or train an end-to-end Multiple Instance Detection Network (MIDN) [1].

The lack of localization supervision during the training process, as expected, makes detection accuracy of WSOD methods worse than its supervised counterparts. However, the promise of a lower annotation cost attracted the efforts of many researchers to WSOD, and significant improvements were achieved in recent years exploring a variety of strategies [2, 19, 20, 18, 23].

In this paper, we focused on the instance mining step of MIL-based methods, and used a modification of an existing baseline approach as a proof-of-concept. More precisely, we propose improvements to boost the performance of OICR, which we call Boosted-OICR (BOICR). We first observed that it is possible to extract extra information from the refinement modules to boost the detection mAP of OICR, which we call refinement knowledge distillation. We also propose an adaptive supervision aggregation function that dynamically changes the IoU threshold to select boxes that will be aggregated as belonging to one of the ground-truth class, background, or ignored during the generation of each refinement module supervision. The selection process follows the principle that at the beginning of the training is better to aggregate boxes with small IoU (since the best instance is typically small and comprehends a small

portion of the object, such as the face for a person or cat). To avoid an overgrowth of the object-related proposals, the IoU threshold is tightened as the training phase advances. We also embedded an adapted version of the "trick" proposed in [20], which ignores boxes with small intersection in the refinement losses. We evaluate our method in Pascal VOC 2007, and our approach presents competitive state-of-art results both in detection mAP and CorLoc mAP.

Our main contributions in this paper are the introduction of: i) a module to distill extra knowledge from refinement agents; and ii) an adaptive supervision aggregation function to mine candidate instances. Next, we present the state-of-the-art on WSOD, and then describe the proposed methodology with the experimental results and conclusions.

2. Related Work

There is a considerable number of WSOD works that precede the CNN era [14, 16, 17]. However, we focus on CNN- based methods as all state-of-the-art methods rely on CNN architectures. The adoption of CNN features was not immediate, and initial works started combining the CNN features with features extracted by other kinds of feature descriptors. Cinbis et. al. [2] proposed a multi-fold multiple instance learning training procedure, which splits the positive instances in K training folds. The method combines the Fisher Vector with CNN features as descriptors, and an objectness refinement is proposed to improve localization accuracy. Since a pre-trained CNN is only used as a feature extractor, its weights are not fine-tuned, which can lead to lower accuracy. Li et al. [10] introduced a two-stage adaptation algorithm. The first stage fine-tunes the network to collect class-specific object proposals with higher precision; the second uses confident object candidates to optimize the CNN representations to turn image classifiers into object detectors gradually. A drawback of the method is the need for individually forwarding each region proposal into CNN to extract features, making the whole process very slow. This problem is solved in more recent methods using Spatial Pyramidal Pooling (SPP) [9].

Bilen et al. [1] proposed a two-stream method, where one stream performs classification and the other detection. The output of both streams is combined into a global scoring matrix by taking the Hadamard product of the two streams. The classification scores are calculated by summing the values in the proposals dimension of this matrix. Tang et al. [19] improved the smoothed version of MIL proposed by [1] using an online instance classification refinement that utilizes cascaded refinement modules to increase the detection performance, where each refinement steep makes the detector able to detect larger objects parts during training gradually. In [18], the refinement process of [19] is further improved, adding proposal clusters to select one or more supervision boxes during the training. Selecting more than one supervision box is interesting because, usually, objects can have multiple parts and also have multiple instances present in the image. However, a limitation of the clustering process is that it increases the computational cost making the whole training process slower. Our Boosted-OICR has a better mAP result than [18] without using the clustering process.

Diba et. al. [5] proposed a three-stage cascaded method that mines boxes from Class Activation Maps (CAM). The first stage is inspired by [26], which uses a fully convolutional CNN with global average pooling (GAP) to create the CAMs in conjunction with the classification scores. The second stage uses the CAM from the first stage as supervision to generate a segmentation map that is used to select a set of candidate bounding boxes using the connective algorithm from [26]. Finally, the features of the candidate boxes are extracted by an SPP layer [9], and a MIL algorithm is applied to select the best candidate boxes for each class. In the same direction, Wei et al. [25] introduced a method that uses CAMs to mine tight object boxes by exploiting segmentation confidence maps. The segmentation confidence maps are employed to evaluate the objectness scores of proposals according to two properties – purity and completeness –, and the detection process is based on [19]. Although the idea of using CAMs to guide the selection of the supervision boxes is interesting, the training process of [5, 25] is overly complex.

Wan et al. [24] proposed a min-entropy latent model to measure the randomness of object localization. The learning process operates with two network branches. The first branch is designated for discovering objects using a global min-entropy layer that defines the distribution of object probability. This discovery process targets at finding candidate object cliques, which is a proposal with high object confidence. The second branch is designated to localize objects using a local min-entropy layer and a softmax layer. The local min-entropy layer classifies the object candidates in a clique into pseudo objects and hard negatives by optimizing the local entropy.

Non-convexity is also a common problem in multiple instance learning, which might lead to sub-optimal results. Wan et al. [23] introduced a continuation optimization method that uses a series of smoothed loss functions to approximate the target (desired) loss, claiming that this smoothed process alleviates the non-convexity problem in MIL. The authors also propose a parametric strategy, for instance, subset partition, which is combined with a deep neural network to activate a full object extent. In contrast, Tang et al. [20] proposed a two-stage region proposal network that explores the responses in mid-layers of a network to create object proposals. The process creates coarse proposals using an objectness score metric and sliding window boxes. Later, the coarse proposals are refined proposals us-

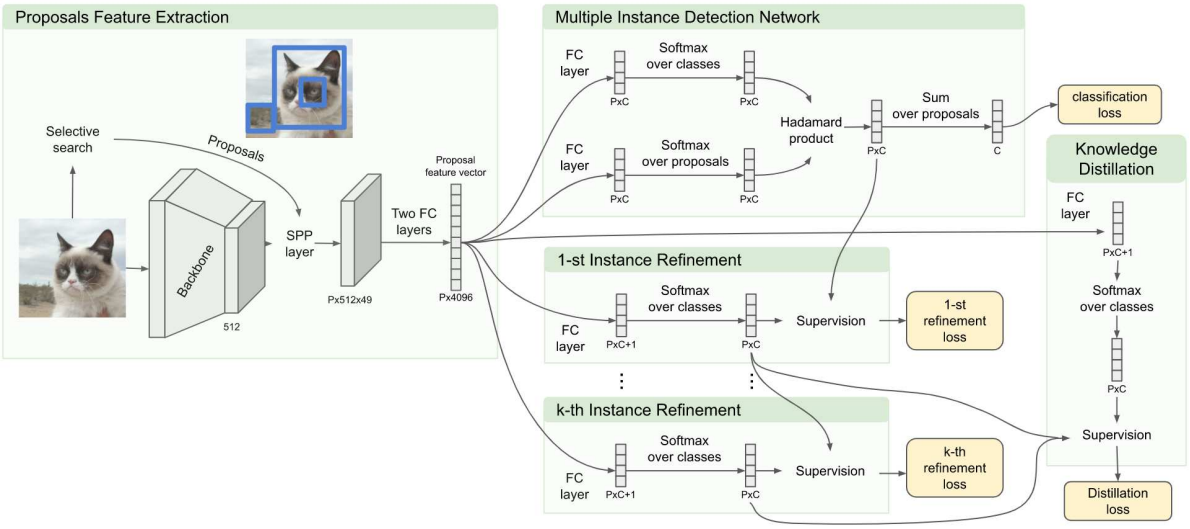

Figure 1: The proposed architecture and its four modules. The proposals feature extraction module uses an SSP layer to extract features from proposals generated by selective search. The multiple instance detection network module learns to select the best proposal instance and generates an image classification score. The instance refinement modules have k instances, and each one learns to refine instances from its predecessor result. Finally, the knowledge distillation module aggregates all the knowledge learned by all the K refinement agents.

ing a region-based CNN classifier, which are used to train the network proposed in [19].

In summary, existing WSOD approaches vary regarding the selection of candidate proposals, the strategy for mining instances, and the underlying classification network that guides the supervision, which leads to different levels of complexity for both implementation and training times. This paper focuses mostly on the instance selection part, and we used the continuation function proposed in [23] as inspiration to adaptively select positive and negative instances. We also present and additional step to the refinement supervision of [19]. The proposed method is presented next.

3. The Proposed Approach

Since we propose improvements to boost OICR's pipeline [19], we will try to follow the same notation of the original paper, and Fig. 1 shows a high-level diagram of all stages of the proposed architecture. The first stage aims to extract feature vectors from a given image, and candidate proposals are extracted using selective search [22]. The image and the extracted proposals feed a CNN backbone with SPP to produce a fixed-size feature map to each proposal. The proposals feature maps are converted to proposal feature vectors using two fully connected (fc) layers, which are branched into three different stages. The two first stages are similar to [19] stages, where the first one trains a basic instance classifier, and the second stage trains a set of K refinement agents. The k^{th} refinement agent uses as supervision the output from the previous agent $\{k-1\}$, and the supervision for the 1^{st} refinement agent ($k=1$) comes from the instance classifier branch. The third state, proposed by us, utilizes the knowledge of all K refinement agents to train a new agent. We call this process knowledge distillation as it aims to extract extra knowledge during the refinement process.

In this section, we will explain all the employed stages in detail. Also, in section 3.4, we explain the adaptive supervision aggregation function that is employed by all refinement agents during the learning process.

3.1. Instance selection

Following [19], we use the method proposed by [1] because of its effectiveness and implementation convenience. The instance selection works by branching the proposal feature vectors into two streams, and each stream starts with an fc layer to produce two matrices $\mathbf{x}^c$, $\mathbf{x}^d \in \mathbb{R}^{C\times|R|}$, where C is the number of classes and $|R|$ is the number of proposals. A softmax function is applied to both matrices along different dimensions, yielding

$$\sigma(\mathbf{x}^c)]_{ij} = \frac{e^{x^c_{ij}}}{\sum_{k=1}^{C} e^{x^c_{kj}}}, \quad \sigma(\mathbf{x}^d)]_{ij} = \frac{e^{x^d_{ij}}}{\sum_{k=1}^{|R|} e^{x^c_{ik}}}. \quad (1)$$

The two streams are then combined to generate proposal scores using Hadamard (element-wise) matrix product, yielding $\mathbf{x}^R = \sigma(\mathbf{x}^c) \odot \sigma(\mathbf{x}^d)$. Finally, the classifica-

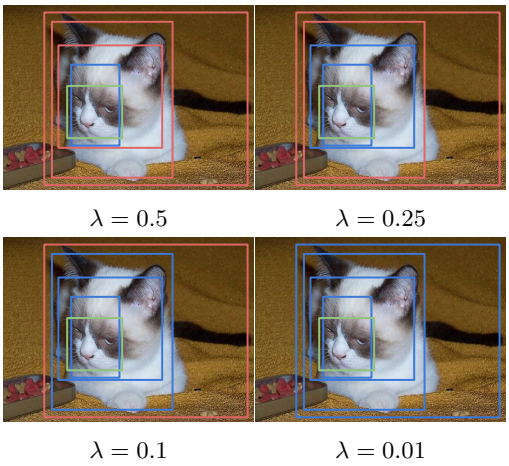

Figure 2: Effect of changing the IoU threshold λ for instance selection. Green boxes are denote the supervision, blue boxes pass the threshold (selected) and red boxes fail (not selecetd).

tion score $\phi_c \in (0, 1)$ for class c is obtained by by summing over proposal dimensions, i.e,. $\phi_c = \sum_{r=1}^{|R|} \mathbf{x}_{cr}^R$. We train the instance classifier using multi-class cross entropy loss, defined as

$$L_{class} = -\sum_{c=1}^{C} y_c \log \phi_c + (1 - y_c) \log(1 - \phi_c), \quad (2)$$

where $y_c = \in 0, 1$ indicates if the image contains any instance of class c in the image. More details can be found in [1, 19]

3.2. Classifier refinement agents

To refine the outputs of the instance classifier, we use the online labeling and refinement strategy proposed by [19]. Here we refer to each k^{th} refinement pass as k^{th} refinement agent. In contrast with the instance classifier, each refinement agent outputs an additional dimension for background in its score vector $\mathbf{x}_j^{Rk} \in \mathbb{R}^{(C+1)\times 1}$, $k \in 1, 2, ..., K$, where the k is the index of the agent, K is the total of agents, and the $C + 1^{th}$ dimension relates to the background. The score vector from the instance classifier is represented here as $\mathbf{x}_j^{R0} \in \mathbb{R}^{C\times 1}$, and is used to initialize the refinements. To obtain $\mathbf{x}_j^{Rk}$ for $k > 0$, the feature vector related to the proposals is passed through a single fc layer, and a softmax layer is applied over class dimension.

Each agent needs some kind of supervision to learn how to separate the proposals related to the background from those related to ground-truth classes. Thus, the supervision for agent k is obtained from the previous agent $\mathbf{x}^{R(k-1)}$ and a supervision label vector is created for each proposal j in the format $\mathbf{Y}_j^k = [y_{1j}^k, y_{2j}^k, \cdots, y_{(C+1),j}^k]^T \in \mathbb{R}^{(C+1)\times 1}$.

To build $\mathbf{Y}_j^k$, first the proposal with highest score is selected from the agent $k - 1^{th}$ supervision, sa given in Eq. (3).

$$j_c^{k-1} = \arg\max_r x_{cr}^{R(k-1)}. \quad (3)$$

The highest score proposal is labeled as belonging to class c, i.e., $y_{cj_c^{k-1}}^k = 1$ and $y_{c'j_c^{k-1}}^k = 0$, $c' \neq c$. Next, proposals with high overlap with j_c^{k-1} are labeled as belonging to the same class of j_c^{k-1}, otherwise the adjacent proposals are labeled as background. More precisely, this assignment is given by

$$c^{*k}_j = \begin{cases} c, & \text{if } IoU(j_c^{k-1}, j_{cj}^k) \geq \lambda \\ C + 1, & \text{otherwise} \end{cases}, \quad (4)$$

where λ is the IoU threshold. We claim in this work that selecting a fixed value for λ might not be the best choice, and present our dynamic threshold in Section 3.4. Each y_{cj}^k is updated using c^{*k}_j, that is, $y_{c^{*k}_j j}^k = 1$. Meanwhile, if there is no object c in the image, all values are set to zero, i.e., $y_{cj}^k = 0$.

Now that y_{cj}^k is ready it can be used as supervision to train the k^{th} refine agent using the loss function in Eq. 5.

$$L_{agent}^K = -\frac{1}{|R|} \sum_{r=1}^{|R|} \sum_{c=1}^{C+1} w_r^k y_{cr}^k \log x_{cr}^{Rk}, \quad (5)$$

where w_r^k is a weight term introduced to reduce noise during the supervision and is obtained as $w_r^k = x_{cj_c^{k-1}}^{Rk-1}$. More details can be found in [19].

3.3. Knowledge distillation module

The motivation behind cascading K refinement agents in [19] is that it allows the detector to gradually learn larger parts of objects, starting from the best instance only. However, we can observe that the supervision generated by a k^{th} agent will not be directly used by the $k + 2^{th}$ agent. This happens because agent $k + 1$ will learn with the supervision k and will pass its own supervision to the next agent $k + 2$. In other words, during the agent supervision process, some knowledge could be lost between the connections of the agents. We try to recover this information loss using our knowledge distillation module. The distillation agent is a special kind of agent that learns using all the K outputs as supervision. In reality, this agent only differs in the supervision part when compared with a standard refinement agent.

The distillation agent also outputs a score vector in the format $\mathbf{x}_j^{Dk} \in \mathbb{R}^{(C+1)\times 1}$. To obtain $\mathbf{x}_j^{Dk}$, the proposals-related feature vector is passed through a single fc layer, and a softmax layer is applied over the class dimension.

The supervision process of the distillation agent, instead of getting the supervision from a previous agent, uses all

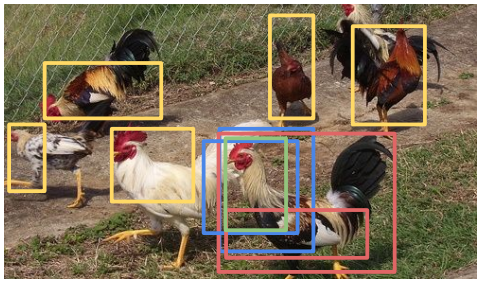

Figure 3: A visual example of instance mining for "chicken" class, where the green box is the best instance. Boxes in blue present large IoU, in red present small (but not zero) IoU, and in yellow, the IoU is zero.

refinement agents outputs as supervision. More precisely, it is computed by averaging the outputs of the K refinement agents outputs:

$$\mathbf{x}_{cj}^{D} = \frac{1}{K}\sum_{k=1}^{K}\mathbf{x}_{cj}^{Rk}. \tag{6}$$

Using $\mathbf{x}_{cj}^{D}$ as the input to the supervision, the remaining process is similar to the described in section 3.2 and the loss function $L_{destill}$ is the same as the weighted softmax loss in Eq. (5).

3.4. Adaptive supervision aggregation function

In [19], the authors experimentally chose $\lambda = 0.5$ as the proposal selection scheme in Eq. (4) to create the supervision matrices w_r^k and y_{cr}^k. The interpretation of this value is that only boxes with $IoU > 0.5$ w.r.t. the best overall proposal are selected as belonging to the ground-truth class c. The problem with using a fixed value is that at the beginning of training, the instance selection module tends to select only small boxes as top score proposals, typically related to discriminant features of the objects (e.g., the face of a person or animal, as shown in Fig. 2). As a consequence, only other small boxes will have $IoU > 0.5$ w.r.t. this box, and hence only small boxes will be considered as belonging to the class c. Figure 2 shows the effect of changing λ, where green denotes the best proposal, and blue the similar proposals according to the selected threshold.

Although the goal of refinement agents is to gradually improve the detectors to find larger parts of objects, starting with a larger value for λ causes each agent to highlight only small boxes in beginning of the process, and in some cases, the optimization will be stuck in small boxes during all training (especially for deformable objects). Relaxing λ alleviates this issue, but it also tends to include proposals that are not related to the correct class.

Instead of using a fixed value for λ, we use an adaptive supervision function that changes λ during the training process. The function should be monotonically increasing, such that more candidates are aggregated in the beginning and less at the end. During our experiments, we evaluated a set of different adaptive supervision aggregation functions, and the best results were archived using the following function, also explored by C-MIL in a different context [23]:

$$\lambda = \frac{1}{2}\frac{\log(s + l_b) - \log l_b}{\log(S + l_b) - \log l_b}, \tag{7}$$

where s is the current training step, S is the total of training steeps, and l_b defines the velocity that the curve grows.

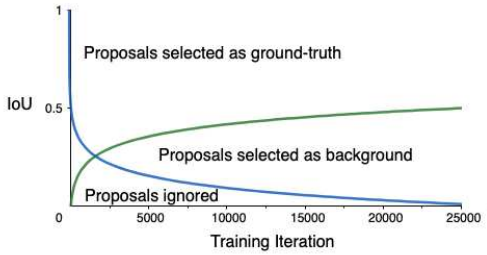

Figure 4: A visual interpretation of the proposed adaptive supervision aggregation function. X-axis shows the iteration step number, and Y-axis shows the IoU with the box of the highest score.

Another deficiency of the supervision selection approach given by Eq. (4) is that when more than one instance of a class is present in the image, it will obligatorily include all other instances as background in during the supervision (since their IoU with the best instance is small – in general, null). This is a bad decision, as we do not want to lower the scoring of these instances. In Fig. 3, we present a visual example of this problem, considering the "chicken" class. In the figure, the rectangles are the candidate proposals, with the best one shown in green. Boxes shown in blue indicate proposals considered similar to the best one, according to Eq. (4), which leaves several proposals related to the chicken class (in yellow) marked as background, which is not desirable.

One solution to solve the penalization of other instances in the loss is to include the "trick" proposed by [20], where a threshold value λ_{ign} is used to ignore boxes with a low IoU w.r.t. j_c^{k-1} in the loss. With the trick, all the instances of Fig. 3 in yellow would be ignored, and the ones in red would be marked as background.

In contrast to [20], where λ_{ign} has a fixed value, we propose to use an adaptive value similar to the scheme used for mining positive instances. Although the choice for λ_{ign} could be independent from λ, we propose a "complementary" threshold selection scheme given by

$$\lambda_{ign} = \lambda_{max} - \lambda, \tag{8}$$

where λ_{max} defines the starting point of the adaptive trick.

Fig. 4 presents the visual interpretation of λ and λ_{ign} during the supervision process. Thus, we can adapt Eq. (4) to include the trick as is defined in Eq. (9), leading to

$$c^{*k}_{j} = \begin{cases} c, & \text{if } IoU(j^{k-1}_{c}, j^{k}_{cj}) \geq \lambda, \\ C+1, & \text{if } IoU(j^{k-1}_{c}, j^{k}_{cj}) \geq \lambda_{ign}, \\ -1, & \text{otherwise} \end{cases} \quad (9)$$

where -1 defines indices to be ignored in the agent loss functions.

3.5. Final loss function

The classification, refinement and distillations modules present individual loss functions. However, we train our model using a single loss that combine the individual loss functions given by

$$L = L_{class} + L_{distill} + \sum_{k=1}^{K} L^{k}_{agent}. \quad (10)$$

.

4. Experiments

Boosted-OICR was evaluated on the challenging PASCAL VOC 2007 and 2012 datasets [7]. Although the ground truth bounding box annotations are present in these datasets, we only use the (weak) classification annotations (presence or absence of a class in the given image). The performed evaluation is based on the two standard metrics in WSOD, that is, mean average precision (mAP) [7] and correct localization (CorLoc) [4]. The former provides a measure of how well the detector adapts to all instances, while the latter indicates if the best detection is a good match. Both metrics utilizes PASCAL criteria of $IoU > 0.5$ between ground truths and predicted boxes.

4.1. Implementation Details

All experiments were performed using PyTorch 1.2 [12][1]. Our method uses VGG16 [15] pre-trained on ImageNet [3] as backbone. We replaced the last max-pooling layer by the SPP layer, and the last FC layer and softmax loss layer by the layers described in Section 3. The new layers are initialized using Gaussian distributions with 0-mean and standard deviations 0.01. Biases are initialized to 0. The object proposals are extracted using Selective Search [22]. For data augmentation, the input images were re-sized into five scales $\{480, 576, 688, 864, 1200\}$ concerning the smallest image dimension. During training time, the scale of the image was randomly selected, and the image was randomly horizontal flipped, which is a standard approach among WSOD methods [23, 19, 24, 18] and creates a total of ten augmented images. The learning process was done using the SGD algorithm with momentum 0.9, weight decay $5e^{-4}$, and batch size 2. We set $l_b = 100$ and $\lambda_{max} = 0.51$. The learning rate is set to 0.001 for the first 30K and 60K iterations and then decreases to 0.0001 in the following 20K and 30K iterations, respectively, for pascal VOC 2007 and 2012. During test time, all ten images are passed in the network, and the outputs are averaged. As an additional result, we also trained a supervised object detector by choosing top-scoring proposals as ground truth labels, as done in [19, 18, 23]. To make a fair comparison, we also trained a Fast RCNN (FRCNN) [8] detection network using the five image scales. The supervision boxes are chosen by its score (larger than 0.3) and using non-maxima suppression (with 30% IoU threshold).

[1]Source code available at: http://github.com/luiszeni/Boosted-OICR

ID	K	λ	λ_{ign}	distillation	mAP
1	3	0.5	0	No	42.3
2	3	adaptive	0	No	41.6
3	3	adaptive	adaptive	No	46.6
4	3	adaptive	adaptive	Yes	**49.7**
5	4	adaptive	adaptive	No	48.1

Table 1: Ablation study performance (%) on the VOC 2007.

4.2. Ablation experiments

We conduct some ablation experiments to illustrate the effectiveness of the proposed improvements over the baseline method OICR [19].

We first study the impact of using the adaptive supervision aggregation function instead of fixed IoU thresholds for proposal mining. We display the different scenarios in Table 1. The experiment with ID$=1$ presents the results using the standard OICR pipeline. In the experiment ID$=2$ we replace the fixed λ value by the proposed adaptive aggregation function defined in Eq. (7), in this experiment all boxes with $IoU < \lambda$ are considered as background. As the experiment suggests, using the adaptive supervision aggregation function alone without the adaptive trick makes the results worse than the OICR's baseline. However, adding the adaptive trick (experiment ID=3) leads to an improvement of 4.3% in the final mAP, suggesting that using our adaptive supervision aggregation function can boost the OICR detection mAP significantly.

We also evaluated the effect of including the distillation refinement module. In fact, one could argue that using such a module could produce the same result as cascading one more refinement agent. To show the difference, we tested our method using $K = 4$ (and no distillation) vs. $K = 3$ with distillation, and results with distillation were considerably better (see experiments ID$=4$ vs. ID$=5$ in Table 1). As we can see, adding the knowledge distillation improves

Network	Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
VGG16	WSDDN [1]	46.4	58.3	35.5	25.9	14.0	66.7	53.0	39.2	8.9	41.8	26.6	38.6	44.7	59.0	10.8	17.3	40.7	49.6	56.9	50.8	39.2
	OICR [19]	58.0	62.4	31.1	19.4	13.0	65.1	62.2	28.4	24.8	44.7	30.6	25.3	37.8	65.5	15.7	24.1	41.7	46.9	64.3	62.6	42.0
	WCCN [5]	49.5	60.6	38.6	29.2	16.2	70.8	56.9	42.5	10.9	44.1	29.9	42.2	47.9	64.1	13.8	23.5	45.9	54.1	60.8	54.5	42.8
	TS2C [25]	59.3	57.5	43.7	27.3	13.5	63.9	61.7	59.9	24.1	46.9	36.7	45.6	39.9	62.6	10.3	23.6	41.7	52.4	58.7	56.6	44.3
	WeakRPN [21]	57.9	**70.5**	37.8	5.7	21.0	66.1	69.2	59.4	3.4	57.1	**57.3**	35.2	**64.2**	68.6	**32.8**	**28.6**	50.8	49.5	41.1	30.0	45.3
	PCL [18]	54.4	69.0	39.3	19.2	15.7	62.9	64.4	30.0	**25.1**	52.5	44.4	19.6	39.3	67.7	17.8	22.9	46.6	57.5	58.6	63	43.5
	MELM [24]	55.6	66.9	34.2	29.1	16.4	68.8	68.1	43.0	25.0	**65.6**	45.3	53.2	49.6	68.6	2.0	25.4	**52.5**	56.8	62.1	57.1	47.3
	C-MIL [23]	62.5	58.4	49.5	**32.1**	19.8	70.5	66.1	**63.4**	20.0	60.5	52.9	53.5	57.4	68.9	8.4	24.6	51.8	**58.7**	66.7	63.5	**50.5**
	Ours	**68.6**	62.4	**55.5**	27.2	**21.4**	**71.1**	**71.6**	56.7	24.7	60.3	47.4	**56.1**	46.4	**69.2**	2.7	22.9	41.5	47.7	**71.1**	**69.8**	49.7
FRCNN Re-train	OICR [19]	65.5	67.2	47.2	21.6	22.1	68.0	68.5	35.9	5.7	63.1	49.5	30.3	64.7	66.1	13.0	25.6	50.0	57.1	60.2	59.0	47.0
	TS2C [25]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	48.0
	PCL [18]	63.2	**69.9**	47.9	22.6	27.3	71.0	69.1	49.6	12.0	60.1	51.5	37.3	63.3	63.9	15.8	23.6	48.8	55.3	61.2	62.1	48.8
	WeakRPN [21]	63.0	69.7	40.8	11.6	**27.7**	70.5	**74.1**	58.5	10.0	**66.7**	**60.6**	34.7	**75.7**	**70.3**	**25.7**	**26.5**	**55.4**	56.4	55.5	54.9	50.4
	C-MIL [23]	61.8	60.9	**56.2**	28.9	18.9	68.2	69.6	**71.4**	18.5	64.3	57.2	66.9	65.9	65.7	13.8	22.9	54.1	**61.9**	68.2	**66.1**	**53.1**
	Ours	**65.8**	58.6	55.0	**32.4**	19.5	**74.2**	71.4	70.9	**19.2**	54.8	46.2	**67.5**	57.0	65.6	1.4	16.7	40.4	53.0	**69.5**	61.1	50.0

Table 2: Detection performance (%) on the VOC 2007 test set. Comparison to the state-of-the-arts.

Network	Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
VGG16	WSDDN [1]	65.1	58.8	58.5	33.1	39.8	68.3	60.2	59.6	34.8	64.5	30.5	43.0	56.8	82.4	25.5	41.6	61.5	55.9	65.9	63.7	53.5
	OICR [19]	81.7	80.4	48.7	49.5	32.8	81.7	85.4	40.1	40.6	79.5	35.7	33.7	60.5	88.8	21.8	57.9	76.3	59.9	75.3	81.4	60.6
	WCCN [5]	83.9	72.8	64.5	44.1	40.1	65.7	82.5	58.9	33.7	72.5	25.6	53.7	67.4	77.4	26.8	49.1	68.1	27.9	64.5	55.7	56.7
	TS2C [25]	84.2	74.1	61.3	52.1	32.1	76.7	82.9	66.6	42.3	70.6	39.5	57.0	61.2	88.4	9.3	54.6	72.2	60.0	65.0	70.3	61.0
	WeakRPN [21]	77.5	81.2	55.3	19.7	44.3	80.2	86.6	**69.5**	10.1	**87.7**	**68.4**	52.1	**84.4**	**91.6**	**57.4**	**63.4**	**77.3**	58.1	57.0	53.8	63.8
	PCL [18]	79.6	**85.5**	62.2	47.9	37.0	**83.8**	83.4	43.0	38.3	80.1	50.6	30.9	57.8	90.8	27.0	58.2	75.3	**68.5**	75.7	78.9	62.7
	MELM [24]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	61.4
	C-MIL [23]	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	65.0
	Ours	**86.7**	73.3	**72.4**	**55.3**	**46.9**	83.2	**87.5**	64.5	**44.6**	76.7	46.4	**70.9**	67.0	88.0	9.6	56.4	69.1	52.4	**79.8**	**82.8**	**65.7**

Table 3: Localization performance (%) on the VOC 2007 trainval set. Comparison to the state-of-the-arts.

Method	mAP	Corloc
WCCN [5]	37.9	-
OICR [19]	37.9	62.1
TS2C [25]	40	64.4
WeakRPN [21]	40.8	64.9
PCL [18]	40.6	63.2
MELM [24]	42.4	-
C-MIL [23]	**46.6**	**67.4**
Ours	*	66.3

Table 4: Detection (test set) and localization (trainval set) performance (%) on the VOC 2012 dataset using VGG16.

the results in 1.6% mAP more than adding an extra refinement agent. We select the model utilized in the experiment ID=4 as default to the next experiments.

4.3. Comparison with state-of-the-art

We compare our results with other state-of-the-art (SOTA) methods in the Pascal VOC 2007 and 2012 datasets. Table 2 shows a comparison of detection performance of our method and SOTA in the Pascal VOC 2007 test set. It can be seen that Boosted-OICR improves the original OICR paper [19] in 7.7% mAP and outperformed other approaches such as WCCN [5] (6.9%), TS2C [25] (5.4%), WeakRPN [21] (4.4%), PCL [18] (6.2%), and MELM [24] (2.4%). Boosted-OICR was only inferior to C-MIL [23] by a small value (0.8% mAP). However, Boosted-OICR presented the highest AP results in 9 of the total 20 classes (`aeroplane`, `bird`, `bottle`, `bus`, `car`, `dog`, `motorbike`, `train` and `tv`). Figure 5 presents some results generated by our WSOD method. We also re-trained an Fast-RCNN detector using the learned pseudo objects as ground-truth, and achieved 50% mAP, as shown in Table 2, which improved our method by 0.3%.

Table 3 presents a comparison in localization performance of our method and SOTA in the Pascal VOC 2007 train-val set. Boosted-OICR outperformed OICR [19] (5.1%), WCCN [5] (9.0%), TS2C [25] (4.7%), WeakRPN [21] (1.9%), PCL [18] (3.0%), MELM [24] (4.3%), and C-MIL [23] (0.7%). The better corloc result of our method in comparison with C-MIL suggests that C-MIL is just a little better dealing with images with more than one instance (which impacts the final detection mAP). We also compare the localization performance of our method in pascal VOC 2012[2]. in Table 4. Boosted-OICR presents a competitive corloc in VOC 2012 outperforming OICR [19] (4.2%), TS2C [25] (1.9%), WeakRPN [21] (1.4%) and PCL [18] (3.1%), being inferior to C-MIL [23] by 1.1% mAP.

5. Conclusions

In this paper, we propose two improvements to boost the online instance classifier refinement. First, we propose a knowledge distillation methodology that extracts extra knowledge from the refinement agents. Second, we propose an adaptive supervision aggregation function that improves the way that each refinement agent learns to separate

[2] We submitted our results for VOC 2012 to the evaluation server, but still did not get the feedback. The anonymous submission link is `http://host.robots.ox.ac.uk:8080/anonymous/E7JSMD.html`

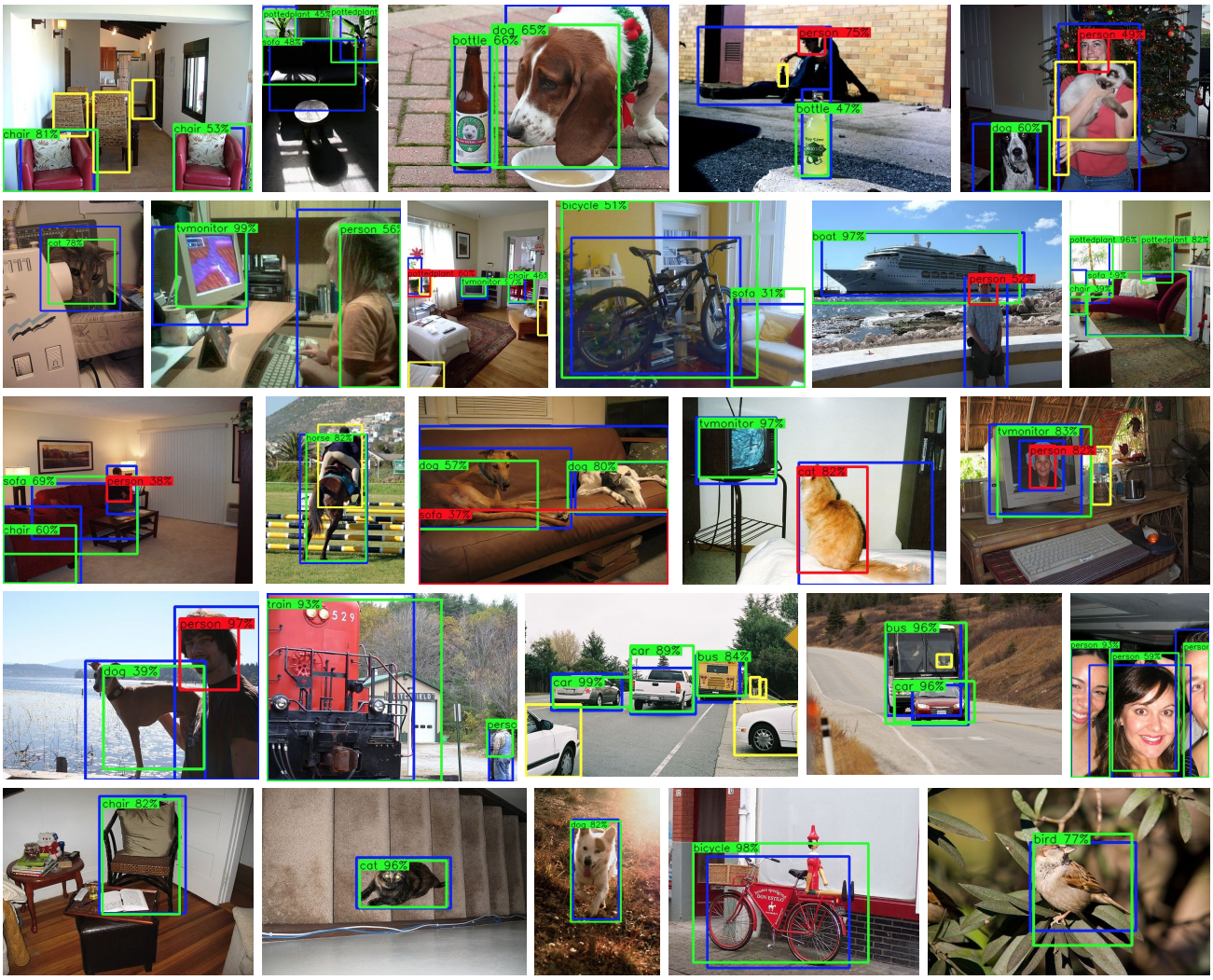

Figure 5: Detection examples for Pascal VOC 2007 dataset. Blue rectangles are ground-truth boxes that have at least one detection with $IoU > 0$, and yellow ones are ground-truth with no detection intersection. Green boxes are correct detections ($IoU > 0.5$ with ground truth), and red boxes are wrong detections. The label in each detection box is the class label and confidence score of the detection.

class-related instances, background instances, and which instances ignore. Both contributions were built using OICR as a baseline approach, and the proposed contributions were able to provide a 7.4 mAP boost over the OICR baseline method. Boosted-OICR presents competitive SOTA results on Pascal VOC 2007 dataset, being inferior only to [23] by a small margin (0.8% mAP). Also, Boosted-OICR presents the highest AP results in 9 of the 20 classes, such as `airplane`, `bird`, `bottle`, and `train`. Although Boosted-OICR has the best performance in these classes, it fails in deformable objects such as `person` class. In fact, the `person` class is very challenging, since the GT annotations might contain only the face or upper body (when there are occlusions), or the whole body.

In the future, we intend to explore improvements that make WSOD methods to not focus on the most discriminated part of deformable objects such as the human face. We further plan to explore mid-layers of the network and class activation maps to create object proposals as an alternative to the selective search module.

Acknowledgments

The authors would like to thank Brazilian funding agencies CNPq and CAPES (Finance Code 001), as well as NVIDIA Corporation for the donation of a Titan Xp Pascal GPU used for this research.

References

[1] Hakan Bilen and Andrea Vedaldi. Weakly Supervised Deep Detection Networks. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-Decem:2846–2854, 2016. 1, 2, 3, 4, 7

[2] Ramazan Gokberk Cinbis, Jakob Verbeek, and Cordelia Schmid. Weakly supervised object localization with multi-fold multiple instance learning. *IEEE transactions on pattern analysis and machine intelligence*, 39(1):189–203, 2016. 1, 2

[3] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 6

[4] Thomas Deselaers, Bogdan Alexe, and Vittorio Ferrari. Weakly supervised localization and learning with generic knowledge. *International journal of computer vision*, 100(3):275–293, 2012. 6

[5] Ali Diba, Vivek Sharma, Ali Pazandeh, Hamed Pirsiavash, and Luc Van Gool. Weakly supervised cascaded convolutional networks. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-Janua:5131–5139, 2017. 1, 2, 7

[6] Thomas G. Dietterich, Richard H. Lathrop, and Tomas Lozano-Perez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71, 1997. 1

[7] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. *International journal of computer vision*, 88(2):303–338, 2010. 6

[8] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 6

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1904–1916, 2015. 2

[10] Dong Li, Jia-Bin Huang, Yali Li, Shengjin Wang, and Ming-Hsuan Yang. Weakly Supervised Object Localization with Progressive Domain Adaptation. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3512–3520, 2016. 1, 2

[11] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 1

[12] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017. 6

[13] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017. 1

[14] Olga Russakovsky, Yuanqing Lin, Kai Yu, and Li Fei-Fei. Object-centric spatial pooling for image classification. *Computer Vision–ECCV 2012*, pages 1–15, 2012. 2

[15] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 6

[16] Parthipan Siva, Chris Russell, Tao Xiang, and Lourdes Agapito. Looking beyond the image: Unsupervised learning for object saliency and detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3238–3245, 2013. 2

[17] Hyun Oh Song, Yong Jae Lee, Stefanie Jegelka, and Trevor Darrell. Weakly-supervised discovery of visual pattern configurations. In *Advances in Neural Information Processing Systems*, pages 1637–1645, 2014. 2

[18] Peng Tang, Xinggang Wang, Song Bai, Wei Shen, Xiang Bai, Wenyu Liu, and Alan Loddon Yuille. Pcl: Proposal cluster learning for weakly supervised object detection. *IEEE transactions on pattern analysis and machine intelligence*, 2018. 1, 2, 6, 7

[19] Peng Tang, Xinggang Wang, Xiang Bai, and Wenyu Liu. Multiple instance detection network with online instance classifier refinement. *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017-January:3059–3067, 2017. 1, 2, 3, 4, 5, 6, 7

[20] Peng Tang, Xinggang Wang, Angtian Wang, Yongluan Yan, Wenyu Liu, Junzhou Huang, and Alan Yuille. Weakly supervised region proposal network and object detection. In *Proceedings of the European conference on computer vision (ECCV)*, pages 352–368, 2018. 1, 2, 5

[21] Peng Tang, Xinggang Wang, Angtian Wang, Yongluan Yan, Wenyu Liu, Junzhou Huang, and Alan Yuille. Weakly Supervised Region Proposal Network and Object Detection. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11215 LNCS:370–386, 2018. 7

[22] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013. 1, 3, 6

[23] Fang Wan, Chang Liu, Wei Ke, Xiangyang Ji, Jianbin Jiao, and Qixiang Ye. C-MIL: Continuation Multiple Instance Learning for Weakly Supervised Object Detection. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1:2199–2208, 2019. 1, 2, 3, 5, 6, 7, 8

[24] Fang Wan, Pengxu Wei, Zhenjun Han, Jianbin Jiao, and Qixiang Ye. Min-Entropy Latent Model for Weakly Supervised Object Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2019. 2, 6, 7

[25] Yunchao Wei, Zhiqiang Shen, Bowen Cheng, Honghui Shi, Jinjun Xiong, Jiashi Feng, and Thomas Huang. Ts2c: Tight box mining with surrounding segmentation context for weakly supervised object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 434–450, 2018. 2, 7

[26] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, pages 2921–2929. IEEE, 2016. 2

[27] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *European Conference on Computer Vision*, pages 391–405. Springer, 2014. 1