

Multiple Instance Detection Network with Online Instance Classifier Refinement

Peng Tang Xinggang Wang Xiang Bai Wenyu Liu*
 School of EIC, Huazhong University of Science and Technology
 {pengtang, xgwang, xbai, liuwy}@hust.edu.cn

Abstract

Of late, weakly supervised object detection is with great importance in object recognition. Based on deep learning, weakly supervised detectors have achieved many promising results. However, compared with fully supervised detection, it is more challenging to train deep network based detectors in a weakly supervised manner. Here we formulate weakly supervised detection as a Multiple Instance Learning (MIL) problem, where instance classifiers (object detectors) are put into the network as hidden nodes. We propose a novel online instance classifier refinement algorithm to integrate MIL and the instance classifier refinement procedure into a single deep network, and train the network end-to-end with only image-level supervision, i.e., without object location information. More precisely, instance labels inferred from weak supervision are propagated to their spatially overlapped instances to refine instance classifier online. The iterative instance classifier refinement procedure is implemented using multiple streams in deep network, where each stream supervises its latter stream. Weakly supervised object detection experiments are carried out on the challenging PASCAL VOC 2007 and 2012 benchmarks. We obtain 47% mAP on VOC 2007 that significantly outperforms the previous state-of-the-art.

1. Introduction

With the development of Convolutional Neural Network (CNN) [17, 19], great improvements have been achieved on object detection [12, 13, 23, 25, 26], due to the availability of large scale datasets with accurate boundingbox-level annotations [8, 11, 22]. However, collecting such accurate annotations can be very labor-intensive and time-consuming, whereas achieving only image-level annotations (i.e., image tags) is much easier, as these annotations are often available at the Internet (e.g., image search queries [21]). In this paper, we aim at the Weakly Supervised Object Detection (WSOD) problem, i.e., only image tags are available during

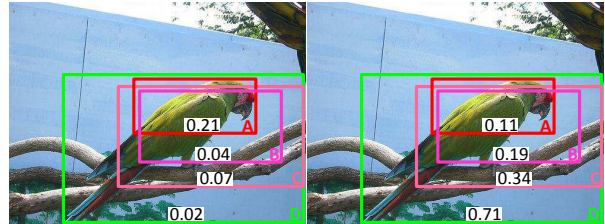


Figure 1. Detection results without/with classifier refinement (left/right). Detection scores are plotted in the bottom of the sampled proposals A, B, C, and D. In the left, the top ranking proposal A does not correctly localize the object. After instance classifier refinement, in the right, the correct proposal D is detected and more discriminative performance of instance classifier is shown.

training to indicate whether an object exists in an image.

Most of previous methods follow the Multiple Instance Learning (MIL) pipeline for WSOD [3, 4, 7, 16, 28, 30, 31]. They treat images as bags and image regions generated by object proposal methods [29, 35] as instances to train instance classifiers (object detectors) under the MIL constraints [10]. Meanwhile, recent efforts tend to combine MIL and CNN by either using CNN as an off-the-shelf feature extractor [3, 7, 28, 30, 31] or training an end-to-end MIL network [4, 16]. Here we are also along the MIL line for WSOD, and train an end-to-end network.

Though many promising results have been achieved in WSOD, they are still far from comparable to fully supervised ones [12, 13, 26]. Weakly supervised object detection only requires supervision at image category level. Bilen and Vedaldi [4] presents an end-to-end deep network for WSOD, in which final image classification score is the weighted sum of proposal scores, that is, each proposal contributes a percentage to the final image classification. The deep network can correctly classify image even only “see” a part of object, and as a result, the top ranking proposal may fail to meet the standard object detection requirement ($\text{IoU} > 0.5$ between ground truths and predicted boxes). As shown in Fig. 1 (left), the top-ranking proposal A is too small. Meanwhile, proposals B, C, and D have similar detection scores. This shows that the WSOD network is not

*Corresponding author.



Figure 2. Detection results from different stages of classifier refinement. Each row represents one stage. Green/red rectangles indicate detected boxes having high/low overlap with ground truths, and digits in the top right corner of rectangles indicate the IoU. Through multi-stage refinement, the detector can cover the whole object instead of parts gradually.

discriminative enough to correctly localize object. This is a core problem of end-to-end deep network based WSOD. To address this problem, we put forward two improvements in this paper: 1) Instead of estimating instance weights through weighted sum pooling, we propose to add some blocks in the network for learning more discriminative instance classifiers by explicitly assigning binary instance labels; 2) We propose to refine instance classifier online using spatial relation.

Our motivation is that, though some detectors only capture objects partially, proposals having high spatial overlaps with detected parts may cover the whole object, or at least contain larger portion of the object. In [4], Bilen and Vedaldi propose a spatial regulariser via forcing features of highest scoring region and its adjacent regions to be the same, which significantly improves WSOD performance. Nevertheless, forcing spatially overlapped proposals to have the same features seems too rigorous. Rather than taking the rigorous constraint, we think the features of spatially overlapped proposals are in the same manifold. Then these overlapped proposals could share similar label information. As shown in Fig. 1 (right), we expect the label information of A can propagate to B and C which has large overlap with A, and then the label information of B and C can propagate to D to correctly localize object. To implement this idea, we design some instance classifiers in the network of [4]. The labels of instance could be refined by their spatially overlapped instances. We name this new network structure Multiple Instance Detection Network (MIDN) with instance classifier.

In practice, there are two important issues. 1) How to initialize instance labels, since there is no instance-level su-

pervision in this task. 2) How to train the network with instance classifier efficiently. A natural way for classifier refinement is the alternative strategy, that is, alternatively relabelling instance and training instance classifier, while this procedure is very time-consuming, especially considering training deep networks with a huge number of Stochastic Gradient Descent (SGD) iterations. To overcome these difficulties, we propose a novel Online Instance Classifier Refinement (OICR) algorithm to train the network online.

Our method has multiple output streams for different stages: the first is the MIDN to train a basic instance classifier and others refine the classifier. To refine instance classifier online, after the forward process of SGD, we can obtain a set of proposal scores. According to these scores, for each stage, we can label the top-scoring proposal along with its spatially overlapped proposals to the image label. Then these proposal labels can be used as the supervision to train instance classifier in the next stage. Though the top-scoring proposal may only contain a part of an object, its adjacent proposals will cover larger portion of the object. Thus the instance classifier can be refined. After implementing the refinement procedure multiple times, the detector can discover the whole object instead of parts gradually, as shown in Fig. 2. But in the beginning of training, all classifiers are almost non-trained, which will result in very noisy supervision of refined classifiers, and then the training will deviate from correct solutions a lot. To solve this problem, we design a weighted loss further by assigning different weights to different proposals in different training iterations. Using this strategy, all classifier refinement procedures can thus be integrated into a single network and trained end-to-end. It can improve the performance benefiting from the classifier

refinement procedure. Meanwhile, the multi-stage strategy and online refinement algorithm is very computational efficient in both training and testing. Moreover, performance can be improved by sharing representations among different training stages.

We elaborately conduct many experiments on the challenging PASCAL VOC dataset to confirm the effectiveness of our method. Our method achieves 47.0% mAP and 64.3% CorLoc on VOC 2007 that outperforms previous best performed methods by a large margin.

In summary, the main contributions of our work are listed as follows.

- We propose a framework for weakly supervised learning that combines MIDN with multi-stage instance classifiers. With only supervision of the outputs from its preceding stage, the discriminatory power of the instance classifier can be enhanced iteratively.
- We further design a novel OICR algorithm that integrates the basic detection network and the multi-stage instance-level classifier into a single network. The proposed network is end-to-end trainable. Compared with the alternatively training strategy, we demonstrate that our method can not only reduce the training time, but also boost the performance.
- Our method achieves significantly better results over previous state-of-the-art methods on the challenging PASCAL VOC 2007 and 2012 benchmarks for weakly supervised object detection.

2. Related work

MIL is a classical weakly supervised learning problem and was first proposed in [10] for drug activity prediction. After that, many solutions have been proposed for MIL [1, 31, 33]. In MIL, a set of bags are given, and each bag is associated with a collection of instances. MIL has two constraints: 1) If a bag is positive, at least one instance in the bag is positive; 2) If a bag is negative, all instances in the bag are negative. It is natural to treat WSOD as a MIL problem. Then the problem turns into finding an instance classifier only given bag labels. Our method also follows the MIL line, and the classifier refinement is inspired by the classifier updating procedure in mi-SVM [1] to some extent. The differences are that, in mi-SVM, it uses an alternative strategy to relabel instances and retrain a classifier, while we adopt an online refinement algorithm; the mi-SVM relabel instances according to the instance score predicted by the classifier, while we select instances according to the spatial relation.

Most of the existing methods solve the WSOD problem based on MIL [3, 4, 7, 16, 24, 28, 31]. For example, Wang *et al.* [31] relaxed the MIL restraints into a differentiable loss

function and optimized it by SGD to speed up training and improve results. Cibis *et al.* [7] trained a multi-fold MIL detector by alternatively relabelling instances and retraining classifier. Recently, some researchers combined CNN and MIL to train an end-to-end network for WSOD [4, 16, 24]. Oquab *et al.* [24] trained a CNN network using the max-pooling MIL strategy to localize objects. But their methods can only coarsely localize objects regardless of their sizes and aspect ratios, our method can detect objects more accurately. Bilen and Vedaldi [4] proposed a Weakly Supervised Deep Detection Network (WSDDN), which presents a novel weighted MIL pooling strategy and combines with the proposal objectness and spatial regulariser for better performance. Based on the WSDDN, Kantorov *et al.* [16] used a contrastive model to consider the context information for improvement. We also choose the WSDDN as our basic network, but we combine it with multi-stage classifier refinement, and propose a novel OICR algorithm to train our network effectively and efficiently, which can boost the performance significantly. Different from the spatial regulariser in WSDDN [4] that forces features of highest scoring proposal and its spatially overlapped proposals to be the same, our OICR assumes features of spatially overlapped proposals are in the same manifold, which is more reasonable. Experiments on Section 4 demonstrate that our strategy can obtain more superior results.

The proposal labelling procedure is also related to the semi-supervised label propagation method [2, 34]. But in label propagation, it labels data according to the similarity among labelled and unlabelled data, while we use spatial overlap as the metric; and there are no available labelled instances for propagation, which is quite different from semi-supervised methods. Meanwhile, the sharing representation strategy in our network is similar to multi-task learning [5]. Unlike the multi-task learning that each output stream has their own relatively independent external supervision, in our method, supervision of latter streams only depends on the outputs from their preceding streams.

3. Method

The overall architecture of our method is shown in Fig. 3. Given an image, we first generate about 2,000 object proposals by Selective Search [29]. The image and these proposals are fed into some convolutional (conv) layers with Spatial Pyramid Pooling (SPP) layer [14] to produce a fixed-size conv feature map per-proposal, and then they are fed into two fully connected (fc) layers to generate a collection of proposal feature vectors. These features are branched into different streams, *i.e.*, different stages: the first one is the MIDN to train a basic instance classifier and others refine classifier. Specially, supervision for classifier refinement is decided by outputs from their preceding stages, *e.g.*, supervision of the first refined classifier

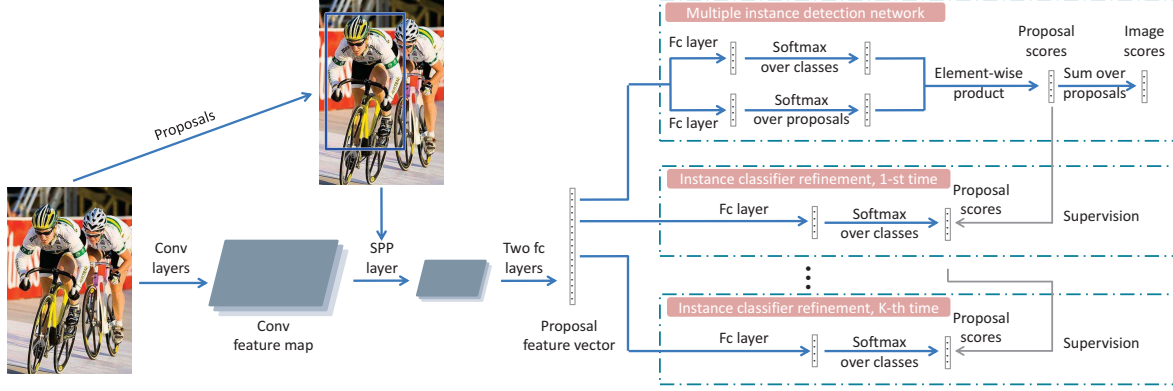


Figure 3. The architecture of MIDN with OICR. Proposal/instance feature is generated by the spatial pyramid pooling layer on the convolutional feature map of image and two fully connected layers. These proposal feature vectors are branched into many streams for different stages: the first one for the basic multiple instance detection network and others for instance classifier refinement. Supervision for classifier refinement is decided by outputs from their preceding stages. All these stages share the same proposal representations.

depends on the output from the basic classifier, and supervision of k^{th} refined classifier depends on outputs from $\{k-1\}^{\text{th}}$ refined classifier.

In this section, we will introduce the chosen basic MIDN, and explain our OICR algorithm in detail.

3.1. Multiple instance detection network

It is necessary to achieve instance-level supervision to train refined classifier, yet such supervision is unavailable. As we have stated before, the top-scoring proposal by instance classifiers and its adjacent proposals can be labelled to its image label as supervision. So we first introduce our MIDN to generate the basic instance classifier. There are many possible choices [4, 7, 16, 31] to achieve this. Here we choose the method by Bilen and Vedaldi [4] which proposes a weighted pooling strategy to obtain the instance classifier, for its effectiveness and implementation convenience. Notice that our network is independent of special MIL methods, so any method that can be trained end-to-end could be embedded into our network.

As shown in the “Multiple instance detection network” block of Fig. 3, proposal features are branched into two streams to produce two matrices $\mathbf{x}^c, \mathbf{x}^d \in \mathbb{R}^{C \times |R|}$ of image by two fc layers, where C denotes the number of image classes and $|R|$ denotes the number of proposals. Then the two matrices are passing through two softmax layer along different directions: $[\sigma(\mathbf{x}^c)]_{ij} = \frac{e^{x_{ij}^c}}{\sum_{k=1}^C e^{x_{kj}^c}}$ and $[\sigma(\mathbf{x}^d)]_{ij} = \frac{e^{x_{ij}^d}}{\sum_{k=1}^{|R|} e^{x_{ik}^d}}$. The proposal scores are generated by element-wise product $\mathbf{x}^R = \sigma(\mathbf{x}^c) \odot \sigma(\mathbf{x}^d)$. At last, image score of c^{th} class ϕ_c can be obtained by the sum over all proposals: $\phi_c = \sum_{r=1}^{|R|} x_{cr}^R$.

The interpretation of the two streams framework is as follows. The $[\sigma(\mathbf{x}^c)]_{ij}$ is the probability of proposal j be-

longing to class i . The $[\sigma(\mathbf{x}^d)]_{ij}$ is the normalized weight that indicates the contribution of proposal j to image being classified to class i . So ϕ_c is achieved by weighted sum pooling and falls in the range of $(0, 1)$. Given image label $\mathbf{Y} = [y_1, y_2, \dots, y_C]^T \in \mathbb{R}^{C \times 1}$, where $y_c = 1$ or 0 indicates the image with or without object c . We can train the basic instance classifier by standard multi-class cross entropy loss, as shown in Eq. (1), then the instance classifier can be obtained according to the proposal score \mathbf{x}^R . More details can be found in [4].

$$\mathcal{L}_b = - \sum_{c=1}^C \{y_c \log \phi_c + (1 - y_c) \log(1 - \phi_c)\}. \quad (1)$$

3.2. Online instance classifier refinement

In the last subsection, we have obtained the basic instance classifier. Here we will expound how to refine instance classifiers online. A natural way to refine classifier is an alternative strategy, that is, fixing the classifier and labelling proposals, fixing proposal labels and training the classifier. But it has some limitations: 1) It is very time-consuming as it requires training the classifier multiple times; 2) Training different classifiers in different refinement steps separately may harm the performance because it hinders the process to benefit from the shared representations. Hence, we integrate the basic MIDN and different classifier refinement stages into a single network and train it end-to-end.

The difficulty is how to obtain instance labels for refinement when there are no available labelled instances. To deal with this problem, we propose an online labelling and refinement strategy. Different from the basic instance classifier, the output score vector \mathbf{x}_j^{Rk} of proposal j for refined classifier is a $\{C+1\}$ -dimensional vector, i.e.,

$\mathbf{x}_j^{Rk} \in \mathbb{R}^{(C+1) \times 1}, k \in \{1, 2, \dots, K\}$, where the k is for k^{th} time refinement, K is the total refinement times, and the $\{C + 1\}^{\text{th}}$ dimension is for background (here we represent the proposal score vector from the basic classifier as $\mathbf{x}_j^{R0} \in \mathbb{R}^{C \times 1}$). The $\mathbf{x}_j^{Rk}, k > 0$ is obtained by passing the proposal feature vector through a single fc layer and a softmax over classes layer, as shown in the “Instance classifier refinement” block of Fig. 3.

Suppose the label vector for proposal j is $\mathbf{Y}_j^k = [y_{1j}^k, y_{2j}^k, \dots, y_{(C+1)j}^k]^T \in \mathbb{R}^{(C+1) \times 1}$. In each training iteration, after the forward process of SGD, we can get a set of proposal scores $\mathbf{x}^{R(k-1)}$. Then we can obtain the supervision of refinement time k according to $\mathbf{x}^{R(k-1)}$. There are many possible methods to obtain instance labels using $\mathbf{x}^{R(k-1)}$, e.g., labeling an instance as positive if its score exceeds a threshold, otherwise as negative, as the mi-SVM [1]. But in our case, the score for each instance is changed during each training iteration, and for different classes, using the same threshold may not be suitable, thus it is hard to settle a threshold. Here we choose a different strategy, inspired by the fact that highly spatially overlapped instances should have the same label. Suppose an image has class label c , we first select proposal j_c^{k-1} with highest score for $\{k-1\}^{\text{th}}$ time as in Eq. (2), and label it to class c , i.e., $y_{cj_c}^{k-1} = 1$ and $y_{c'j_c}^{k-1} = 0, c' \neq c$. As different proposals always have overlaps, and proposals with high overlap should belong to the same class, we can label proposal j_c^{k-1} and its adjacent proposals to class c for k^{th} refinement, i.e., if proposal j have a high overlap with proposal j_c^{k-1} , we label proposal j to class c ($y_{cj}^k = 1$), otherwise we label proposal j as background ($y_{(C+1)j}^k = 1$). Here we label proposal j to class c if the IoU between proposal j and j_c^{k-1} greater than a threshold I_t which is determined by experiments. Meanwhile, if there is no object c in the image, we set all $y_{cj}^k = 0$. Using this supervision, we can train the refined classifier based on the loss function in Eq. (3). Through multiple times of refinement, our detector can detect larger parts of objects gradually.

$$j_c^{k-1} = \arg \max_r x_{cr}^{R(k-1)}. \quad (2)$$

$$\mathbf{L}_r^k = -\frac{1}{|R|} \sum_{r=1}^{|R|} \sum_{c=1}^{C+1} y_{cr}^k \log x_{cr}^{Rk}. \quad (3)$$

Actually the acquired supervision for refining classifier is very noisy, especially in the beginning of training, which will result in unstable solutions. To solve this problem, we change the loss in Eq. (3) to a weighted version, as in Eq. (4).

$$\mathbf{L}_r^k = -\frac{1}{|R|} \sum_{r=1}^{|R|} \sum_{c=1}^{C+1} w_r^k y_{cr}^k \log x_{cr}^{Rk}, \quad (4)$$

Algorithm 1 Online instance classifier refinement

Input: Image X and its proposals; image label vector $\mathbf{Y} = [y_1, \dots, y_C]$; refinement times K .

Output: Loss weights w_r^k ; proposal label vectors $\mathbf{Y}_r^k = [y_{1r}^k, \dots, y_{(C+1)r}^k]^T$. Where $r \in \{1, \dots, |R|\}$ and $k \in \{1, \dots, K\}$.

- 1: Feed X and its proposals into the network to produce proposal score matrices $\mathbf{x}^{Rk}, k \in \{0, \dots, K-1\}$.
 - 2: **for** $k = 0$ **to** $K-1$ **do**
 - 3: Set all elements in $\mathbf{I} = [I_1, \dots, I_{|R|}]^T$ to $-\text{inf}$.
 - 4: Set all $y_{cr}^{k+1} = 0, c \in \{1, \dots, C\}$ and $y_{(C+1)r}^{k+1} = 1$.
 - 5: **for** $c = 1$ **to** C **do**
 - 6: **if** $y_c = 1$ **then**
 - 7: Choose the top-scoring proposal j_c^k by Eq. (2).
 - 8: **for** $r = 1$ **to** $|R|$ **do**
 - 9: Compute IoU I_r' between proposal r and j_c^k .
 - 10: **if** $I_r' > I_r$ **then**
 - 11: Set $I_r = I_r'$ and $w_r^{k+1} = x_{cj_c}^{Rk}$.
 - 12: **if** $I_r > I_t$ **then**
 - 13: Set $y_{c'r}^{k+1} = 0, c' \neq c$ and $y_{cr}^{k+1} = 1$.
-

where w_r^k is the loss weight and can be acquired by the 11th line of Algorithm 1. The explanation of such choice is as follows. In the beginning of training, the w_r^k is small, hence, the loss is also small. As a consequence, the performance of the network will not decrease a lot though good positive instances cannot be found. Meanwhile, during the training procedure, the network can achieve positive instances with high scores easily for easy bags, and these positive instances are always with high scores, i.e., w_r^k is large. On the contrary, it is difficult to get positive instances for difficult bags, as a result, these positive instances are always very noisy. Nevertheless, the refined classifier will not deviate from the correct solution a lot, because the scores of these noisy positive instances are relatively low, i.e., w_r^k is small.

To make the OICR algorithm more clear, we summarize the process to obtain supervision in Algorithm 1, where I_r indicates the maximum IoU between proposal r and the top-scoring proposal. After obtaining supervision and loss for training refined classifiers, we can get the loss of our overall network by combining Eq. (1) and Eq. (4), as Eq. (5). Through optimizing this loss function, we can integrate the basic network and different classifier refinement stages into a single network, and share representations among different stages.

$$\mathbf{L} = \mathbf{L}_b + \sum_{k=1}^K \mathbf{L}_r^k. \quad (5)$$

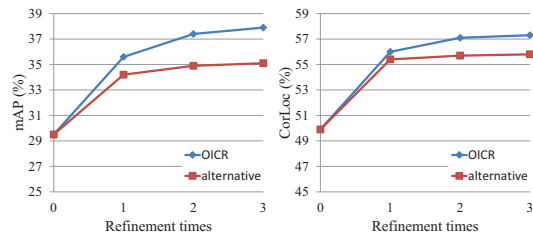


Figure 4. Results on VOC 2007 for different refinement times and different training strategies, where “OICR” indicates our OICR training strategy, “alternative” indicates the alternative strategy.

Loss	mAP (%)	CorLoc (%)
unweighted	32.8	50.6
weighted	37.9	57.3

Table 1. Results on VOC 2007 for different losses.

4. Experiments

4.1. Experimental setup

In this section we will perform thorough experiments to analyse our OICR and its components for weakly supervised object detection.

Datasets and evaluation measures We evaluate our method on the challenging PASCAL VOC 2007 and 2012 datasets [11] which have 9,962 and 22,531 images respectively for 20 object classes. These two datasets are divided into train, val, and test sets. Here we choose the trainval set (5,011 images for 2007 and 11,540 for 2012) to train our network. As we focus on weakly supervised detection, only image-level labels are utilized during training. For testing, there are two metrics for evaluation: mAP and CorLoc. Average Precision (AP) and the mean of AP (mAP) is the evaluation metric to test our model on the testing set, which follows the standard PASCAL VOC protocol [11]. Correct localization (CorLoc) is to test our model on the training set measuring the localization accuracy [9]. All these two metrics are based on the PASCAL criteria, *i.e.*, $\text{IoU} > 0.5$ between ground truths and predicted boxes.

Implementation details Our method is built on two pre-trained ImageNet [8] networks: VGG_M [6] and VGG16 [27], each of which has some conv layers with max-pooling layer and three fc layers. We replace the last max-pooling layer of the two models by SPP layer, and the last fc layer and softmax loss layer by the layers described in Section 3. To increase the feature map size from the last conv layer, we replace the penultimate max-pooling layer and its subsequent conv layers by the dilated conv layers [32]. The new added layers are initialized using Gaussian distributions with 0-mean and standard deviations 0.01. Biases are initialized to 0. During training, the mini-batch size for SGD is set to 2, and the learning rate is set to 0.001 for the

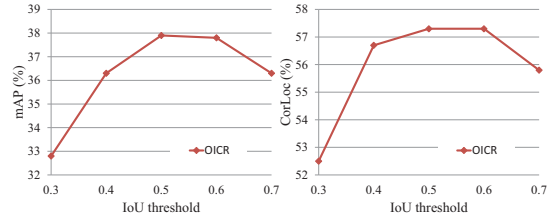


Figure 5. Results on VOC 2007 for different IoU threshold I_t .

first 40K iterations and then decrease to 0.0001 in the following 30K iterations. The momentum and weight decay are set to 0.9 and 0.0005 respectively.

As we have stated in Section 3, Selective Search (SS) [29] is adopted to generate about 2,000 proposals per-image. For data augmentation, we use five image scales {480, 576, 688, 864, 1200} (resize the shortest side to one of these scales) and cap the longest image side to less than 2000 with horizontal flips for both training and testing. We refine instance classifier three times, *i.e.*, $K = 3$ in Section 3.2, so there are four stages in total. The IoU threshold I_t in the 12th line of Algorithm 1 is set to 0.5. During testing, the mean output of these three refined classifiers is chosen. We also follow the [18, 20] to train a supervised object detector by choosing top-scoring proposals given by our method as pseudo ground truths to further improve our results. Here we train a Fast RCNN (FRCNN) [12] detector using the VGG16 model and the same five image scales (horizontal flips only in training). SS is also chosen for proposal generation to train the FRCNN. Non-maxima suppression (with 30% IoU threshold) is applied to compute AP and CorLoc.

Our experiments are implemented based on the Caffe [15] deep learning framework. All of our experiments are running on a NVIDIA GTX TitanX GPU. Codes for reproducing the results are available at <https://github.com/ppengtang/oicr>.

4.2. Ablation experiments

We first conduct some ablation experiments to illustrate the effectiveness of our training strategy, including the influence of classifier refinement, OICR, weighted loss, and the IoU threshold I_t . Without loss generality, we only perform experiments on VOC 2007 and use the VGG_M model.

4.2.1 The influence of instance classifier refinement

As in the blue line of Fig. 4, we can observe that compared with the basic network, even just refining instance classifier one time can boost the performance a lot (mAP from 29.5 to 35.6 and CorLoc from 49.9 to 56.0), which confirms the necessity of refinement. If we refine the classifier multiple times, the results can be improved further. But when re-

Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
WSDDN-VGG-F [4]	42.9	56.0	32.0	17.6	10.2	61.8	50.2	29.0	3.8	36.2	18.5	31.1	45.8	54.5	10.2	15.4	36.3	45.2	50.1	43.8	34.5
WSDDN-VGG-M [4]	43.6	50.4	32.2	26.0	9.8	58.5	50.4	30.9	7.9	36.1	18.2	31.7	41.4	52.6	8.8	14.0	37.8	46.9	53.4	47.9	34.9
WSDDN-VGG16 [4]	39.4	50.1	31.5	16.3	12.6	64.5	42.8	42.6	10.1	35.7	24.9	38.2	34.4	55.6	9.4	14.7	30.2	40.7	54.7	46.9	34.8
WSDDN+context [16]	57.1	52.0	31.5	7.6	11.5	55.0	53.1	34.1	1.7	33.1	49.2	42.0	47.3	56.6	15.3	12.8	24.8	48.9	44.4	47.8	36.3
OICR-VGG-M	53.1	57.1	32.4	12.3	15.8	58.2	56.7	39.6	0.9	44.8	39.9	31.0	54.0	62.4	4.5	20.6	39.2	38.1	48.9	48.6	37.9
OICR-VGG16	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	41.2
WSDDN-Ens. [4]	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	39.3
OM+MIL+FRCNN [20]	54.5	47.4	41.3	20.8	17.7	51.9	63.5	46.1	21.8	57.1	22.1	34.4	50.5	61.8	16.2	29.9	40.7	15.9	55.3	40.2	39.5
OICR-Ens.	58.5	63.0	35.1	16.9	17.4	63.2	60.8	34.4	8.2	49.7	41.0	31.3	51.9	64.8	13.6	23.1	41.6	48.4	58.9	58.7	42.0
OICR-Ens.+FRCNN	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

Table 2. Average precision (in %) for different methods on VOC 2007 test set. The upper part shows results using a single model. The lower part shows results of combing multiple models.

Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mean
WSDDN-VGG-F [4]	68.5	67.5	56.7	34.3	32.8	69.9	75.0	45.7	17.1	68.1	30.5	40.6	67.2	82.9	28.8	43.7	71.9	62.0	62.8	58.2	54.2
WSDDN-VGG-M [4]	65.1	63.4	59.7	45.9	38.5	69.4	77.0	50.7	30.1	68.8	34.0	37.3	61.0	82.9	25.1	42.9	79.2	59.4	68.2	64.1	56.1
WSDDN-VGG16 [4]	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
WSDDN+context [16]	83.3	68.6	54.7	23.4	18.3	73.6	74.1	54.1	8.6	65.1	47.1	59.5	67.0	83.5	35.3	39.9	67.0	49.7	63.5	65.2	55.1
OICR-VGG-M	81.7	72.9	56.5	31.4	36.3	75.6	81.6	57.0	7.3	74.7	47.1	46.0	78.2	88.8	12.2	46.2	66.0	56.7	65.8	64.9	57.3
OICR-VGG16	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
OM+MIL+FRCNN [20]	78.2	67.1	61.8	38.1	36.1	61.8	78.8	55.2	28.5	68.8	18.5	49.2	64.1	73.5	21.4	47.4	64.6	22.3	60.9	52.3	52.4
WSDDN-Ens. [4]	68.9	68.7	65.2	42.5	40.6	72.6	75.2	53.7	29.7	68.1	33.5	45.6	65.9	86.1	27.5	44.9	76.0	62.4	66.3	66.8	58.0
OICR-Ens.	85.4	78.0	61.6	40.4	38.2	82.2	84.2	46.5	15.2	80.1	45.2	41.9	73.8	89.6	18.9	56.0	74.2	62.1	73.0	77.4	61.2
OICR-Ens.+FRCNN	85.8	82.7	62.8	45.2	43.5	84.8	87.0	46.8	15.7	82.2	51.0	45.6	83.7	91.2	22.2	59.7	75.3	65.1	76.8	78.1	64.3

Table 3. CorLoc (in %) for different methods on VOC 2007 trainval set. The upper part shows results using a single model. The lower part shows results of combing multiple models.

finement is implemented too many times, the performance tends to be saturated (the improvement from 2 times to 3 times is small). Maybe this is because the network tends to converge so that the supervision of 3rd time is similar to 2nd time. In the rest of this paper we only refine the classifier 3 times. Notice that in Fig. 4, the “0 time” is similar to the WSDDN [4] using SS as proposals. Our result is a little worse than theirs (30.9 mAP in their paper), due to the different implementing platform and details.

4.2.2 The influence of OICR

Fig. 4 compares the results of different refinement times and different training strategies for classifier refinement. As we can see, whether for our OICR algorithm or the alternative strategy, results can be improved by refinement. More importantly, compared with the alternatively refinement strategy, our OICR can boost the performance consistently and significantly, which confirms the necessity of sharing representations. Meanwhile, our method can also reduce the training time a lot, as it only requires to train a single model instead of training $K + 1$ models for K times refinement in the alternative strategy.

4.2.3 The influence of weighted loss

We also study the influence of our weighted loss in Eq. (4). So here we train a network based on the Eq. (3). From Table 1, we can see that using the unweighted loss, the improvement from refinement is very scant, and the perfor-

Method	mAP (%)	CorLoc (%)
WSDDN+context [16]	34.9	56.1
OICR-VGG-M	34.6	60.7
OICR-VGG16	37.9	62.1
OICR-Ens.	38.2	63.5
OICR-Ens.+FRCNN	42.5	65.6

Table 4. Results for different methods on VOC 2012. Detailed per-class results can be found in Table 1 and Table 2 of the Supplementary Material.

mance is even worse than the alternative strategy. Using the weighted loss can achieve much better performance, which confirms our theory in Section 3.2.

4.2.4 The influence of IoU threshold

In previous experiments, we set the IoU threshold I_t in the 12th line of Algorithm 1 to 0.5. Here we conduct experiments to analyse the influence of I_t . As in Fig. 5, $I_t = 0.5$ outperforms other choices, and the results are not very sensitive to the I_t : when changing I_t from 0.5 to 0.6, the performance only drops a little (mAP from 37.9 to 37.8, CorLoc maintains 57.3). Here we set I_t to 0.5 in other experiments.

4.3. Comparison with other methods

We report our results for each class on VOC 2007 and 2012 in Table 2, Table 3, and Table 4. Compared with other methods, our method achieves the state-of-the-art performance using single model, and even outperforms the results by combining multiple different models [4, 20]. Spe-



Figure 6. Some detection results for class bicycle, bus, cat, chair, dog, motorbike, person, and train. Green rectangle indicates success cases ($\text{IoU} > 0.5$), and red rectangle indicates failure cases ($\text{IoU} < 0.5$).

cially, our methods achieves much better performance than the method by Bilén and Vedaldi [4] using the same CNN model. Notice that [4] not only uses the weighted pooling as we stated in Section 3.1, but also combines the objectness measure of EdgeBoxes [35] and the spatial regulariser, which is much complicated than our basic MIDN. We believe that our performance can be improved by choosing better basic detection network, like the complete network in [4] and using the context information [16]. As reimplementing their method completely is trivial, here we only choose the simplest architecture in [4]. Even in this simplified case, our method can achieve very promising results. We also show some visualization comparisons among the WSDDN [4], the WSDDN+context [16], and our method in Fig. 1 of the Supplementary Material.

Our results can also be improved by combining multiple models. As shown in the tables, if we simply sum up the scores produced by the VGG_M model and VGG16 model (OICR-Ens. in tables), there is little improvement. Also, as mentioned in Section 4.1, we train a FRCNN detector using top-scoring proposals produced by OICR-Ens. as ground truths (OICR-Ens.+FRCNN in tables). As we can see, the performance can be improved further.

Though our method significantly outperforms other methods for some class, like “bicycle”, “bus”, “motorbike”, etc, the performance is poor for classes like “cat”, “dog”, and “person”. For analysis, we visualize some success and failure detection results on VOC 2007 trainval by OICR-

Ens., as in Fig. 6. We can observe that, our method is robust to the size and aspect of objects, especially for rigid objects. The main failures for these rigid objects are always due to overlarge boxes that not only contain objects, but also include their adjacent similar objects. For non-rigid objects like “cat”, “dog”, and “person”, they are always with great deformation, while there is less deformation of their most representative parts (like head), so our detector is still inclined to find these parts. An ideal solution is yet wanted because there is still room for improvement.

5. Conclusion

In this paper, we present a novel algorithm framework for weakly supervised object detection. Different from traditional approaches in this field, our method integrates a basic multiple instance detection network and multi-stage instance classifiers into a single network. Moreover, we propose an online instance classifier refinement algorithm to train the proposed network end-to-end. Experiments show substantial and consistent improvements by our method. Our learning algorithm is potential to be applied in many other weakly supervised visual learning tasks. In the future, we will explore other cues such as instance visual similarity for performing instance classifier refinement better.

Acknowledgements This work was partly supported by NSFC (No. 61503145, No. 61572207, No. 61573160) and the CAST Young Talent Supporting Program.

References

- [1] S. Andrews, I. Tsochanaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *NIPS*, pages 561–568, 2002. 3, 5
- [2] S. Bai, X. Bai, and Q. Tian. Scalable person re-identification on supervised smoothed manifold. In *CVPR*, 2017. 3
- [3] H. Bilen, M. Pedersoli, and T. Tuytelaars. Weakly supervised object detection with convex clustering. In *CVPR*, pages 1081–1089, 2015. 1, 3
- [4] H. Bilen and A. Vedaldi. Weakly supervised deep detection networks. In *CVPR*, pages 2846–2854, 2016. 1, 2, 3, 4, 7, 8
- [5] R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997. 3
- [6] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman. Return of the devil in the details: Delving deep into convolutional nets. In *BMVC*, 2014. 6
- [7] R. G. Cinbis, J. Verbeek, and C. Schmid. Weakly supervised object localization with multi-fold multiple instance learning. *TPAMI*, 39(1):189–203, 2017. 1, 3, 4
- [8] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009. 1, 6
- [9] T. Deselaers, B. Alexe, and V. Ferrari. Weakly supervised localization and learning with generic knowledge. *IJCV*, 100(3):275–293, 2012. 6
- [10] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89(1):31–71, 1997. 1, 3
- [11] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 88(2):303–338, 2010. 1, 6
- [12] R. Girshick. Fast r-cnn. In *ICCV*, pages 1440–1448, 2015. 1, 6
- [13] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Region-based convolutional networks for accurate object detection and segmentation. *TPAMI*, 38(1):142–158, 2016. 1
- [14] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. *TPAMI*, 37(9):1904–1916, 2015. 3
- [15] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM MM*, pages 675–678, 2014. 6
- [16] V. Kantorov, M. Oquab, M. Cho, and I. Laptev. Contextlocnet: Context-aware deep network models for weakly supervised localization. In *ECCV*, pages 350–365, 2016. 1, 3, 4, 7, 8
- [17] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012. 1
- [18] K. Kumar Singh, F. Xiao, and Y. Jae Lee. Track and transfer: Watching videos to simulate strong human supervision for weakly-supervised object detection. In *CVPR*, pages 3548–3556, 2016. 6
- [19] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 1
- [20] D. Li, J.-B. Huang, Y. Li, S. Wang, and M.-H. Yang. Weakly supervised object localization with progressive domain adaptation. In *CVPR*, pages 3512–3520, 2016. 6, 7
- [21] Q. Li, J. Wu, and Z. Tu. Harvesting mid-level visual concepts from large-scale internet images. In *CVPR*, pages 851–858, 2013. 1
- [22] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755, 2014. 1
- [23] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed. SSD: single shot multibox detector. In *ECCV*, pages 21–37, 2016. 1
- [24] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Is object localization for free?-weakly-supervised learning with convolutional neural networks. In *CVPR*, pages 685–694, 2015. 3
- [25] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pages 779–788, 2016. 1
- [26] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015. 1
- [27] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 6
- [28] H. O. Song, Y. J. Lee, S. Jegelka, and T. Darrell. Weakly-supervised discovery of visual pattern configurations. In *NIPS*, pages 1637–1645, 2014. 1, 3
- [29] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013. 1, 3, 6
- [30] C. Wang, W. Ren, K. Huang, and T. Tan. Weakly supervised object localization with latent category learning. In *ECCV*, pages 431–445, 2014. 1
- [31] X. Wang, Z. Zhu, C. Yao, and X. Bai. Relaxed multiple-instance svm with application to object discovery. In *ICCV*, pages 1224–1232, 2015. 1, 3, 4
- [32] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. In *ICLR*, 2016. 6
- [33] Q. Zhang and S. A. Goldman. EM-DD: An improved multiple-instance learning technique. In *NIPS*, pages 1073–1080, 2001. 3
- [34] X. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation. Technical report, CMU-CALD-02-107, 2002. 3
- [35] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, pages 391–405, 2014. 1, 8