Instance Mining with Class Feature Banks for Weakly Supervised Object Detection

Yufei Yin[1], **Jiajun Deng**[1], **Wengang Zhou**[1,2], **Houqiang Li**[1,2]
[1] CAS Key Laboratory of GIPAS, EEIS Department, University of Science and Technology of China
[2] Institute of Artificial Intelligence, Hefei Comprehensive National Science Center
{yinyufei, dengjj}@mail.ustc.edu.cn, {zhwg, lihq}@ustc.edu.cn

Abstract

Recent progress on weakly supervised object detection (WSOD) is characterized by formulating WSOD as a Multiple Instance Learning (MIL) problem and taking online refinement with the selected region proposals from MIL. However, MIL inclines to select the most discriminative part rather than the entire instance as the top-scoring region proposals, which leads to weak localization capability for weakly supervised object detectors. We attribute this problem to the limited intra-class diversity within a single image. Specifically, due to the lack of annotated bounding boxes, the network tends to focus on the most common parts of each class and neglect the diverse parts of objects. To solve the problem, we introduce a novel Instance Mining with Class Feature Banks (IM-CFB) framework, which includes a Class Feature Banks (CFB) module and a Feature Guided Instance Mining (FGIM) algorithm. Concretely, Class Feature Banks (CFB) consist of sub-banks for each class, which are utilized to collect diversity information from a broader view. At the training stage, the RoI features of reliable region proposals are recorded and updated in the CFB. Then, FGIM leverages the features recorded in the CFB to ameliorate the region proposal selection of the MIL branch. Extensive experiments conducted on two publicly available datasets, Pascal VOC 2007 and 2012, demonstrate the effectiveness of our method. More remarkably, our method achieves 54.3% on mAP and 70.7% on CorLoc on Pascal VOC 2007. When further re-trained by a Fast-RCNN detector, we obtain to-date the best reported mAP and CorLoc of 55.8% and 72.2%, respectively.

Introduction

With the revolution of deep learning, object detection techniques have been greatly improved in recent years (Girshick 2015; Ren et al. 2015; Liu et al. 2016; Redmon et al. 2016; Lin et al. 2017). However, many object detection methods with deep learning follow a fully supervised setting and require a large corpus of box-level annotation data, which is laborious and expensive to collect. Recently, substantial efforts (Tang et al. 2017, 2018; Wei et al. 2018; Gao et al. 2018; Wan et al. 2019; Shen et al. 2018; Arun, Jawahar, and Kumar 2019; Li et al. 2019; Diba et al. 2017; Zhang et al. 2018; Kim et al. 2020; Gao et al. 2019; Ren et al. 2020; Chen et al. 2020) have been made on weakly supervised object detection (WSOD) to alleviate the requirement of fine-grained annotation. However, there is still a large performance gap between weakly supervised and fully supervised methods.

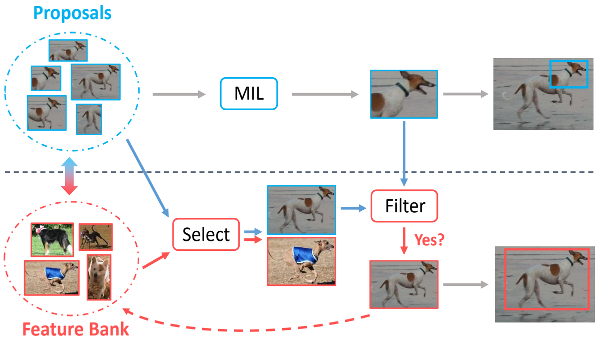

Figure 1: Illustration of our purposed Feature Guided Instance Mining (FGIM). The top part represents the MIL branch and the bottom part represents the FGIM algorithm. Benefited by the extra box-level information from the feature bank, FGIM is able to obtain tighter bounding boxes. In turn, these boxes are utilized as positive samples to update the feature bank. Best viewed in color.

Many recent WSOD approaches follow a two-stage paradigm, where Multiple Instance Learning (MIL) is applied with a CNN model, and the region proposals selected from MIL are utilized for further refinement. However, MIL inclines to locate discriminative parts rather than the entire instance, limiting the accuracy of the selected region proposals. To address this issue, Yan *et al.* (Yan et al. 2019) design an extra MIL branch to refine the initial results and Yang *et al.* (Yang, Li, and Dou 2019) utilize guided attention module to enhance features. Nevertheless, these approaches only focus on a single image, but leave the intra-class diversity uninvestigated. In particular, objects are usually diverse in appearances, (*e.g.*, shape, posture), even if they belong to the same class, while the diversity information for each class is limited in a single image. Hence, the network tends to focus on the most common parts of each class and neglect the diverse parts of objects.

To address the above problem, an intuitive idea is to absorb information from different images for each class. Enlarging batch size is a straightforward solution, however, bringing in more images will lead to more proposals maintained, hence results in the increase of GPU memory consumption and computational cost. Limited by these factors, it is difficult to get cross-image information by enlarging batch size directly. Instead, we design a Class Feature Banks (CFB) module to get rid of the limitation. Specifically, for each class, this module records and updates various box-level features online, allowing for diverse representation of each class. At the beginning of the training stage, CFB gradually absorbs RoI features of confident region proposals, and when it comes to the upper limit of its capacity, CFB applies an algorithm to update existing features using the new ones. As the training goes on, abundant informative feature samples from various images are stored by CFB, with negligible extra GPU memory utilized.

Utilizing extra box-level information from CFB, we propose Feature Guided Instance Mining (FGIM) as a supplement to the MIL branch, as shown in Figure 1. Specifically, we first calculate feature similarities between proposals and feature samples from CFB. Considering the *intra-class diversity*, we select positive proposal candidates based on the similarities between proposals and their closest feature samples. Then, we impose restrictions to filter out noisy candidates, with regard to the *intra-class similarity*. Finally, the remaining positive candidates are utilized for updating CFB and training a FGIM network to obtain more reliable detection results for region proposal selection.

In summary, the main contribution of this work is the proposal of a unified framework IM-CFB, which is able to store and utilize class-wise information for weakly supervised object detection. First, we design a simple yet efficient Class Feature Banks (CFB) module, which is performed to record and update RoI features online for each class. To our knowledge, this is the first work to introduce memory module into the WSOD task. By utilizing the extra information provided by CFB, we apply Feature Guided Instance Mining (FGIM) as a supplement to the MIL branch to ameliorate the region proposal selection. Extensive experiments on widely used datasets, *i.e.* PASCAL VOC 2007 and 2012, demonstrate the effectiveness of our method.

Related Work

In this section, we briefly review the related methods including weakly supervised object detection and memory module.

Weakly Supervised Object Detection. Weakly supervised object detection is of great interest because the demand for annotated data is growing rapidly while image-level annotations are much easier to obtain than box-level annotations. Many recent works are devoted to training end-to-end weakly supervised detection networks (Bilen and Vedaldi 2016; Tang et al. 2017). As one of the most popular frameworks for weakly supervised object detection, WSDDN (Bilen and Vedaldi 2016) first applies Selective Search (Uijlings et al. 2013) to generate large numbers of candidate proposals and then utilizes RoI pooling followed by a two-branch structure to get box-level classification scores. Finally, WSDDN obtains the image-level classification scores by a summation over all proposals and combines them with the ground truth labels to train the network. Based on WSDDN, OICR (Tang et al. 2017) adds several branches to refine the classification scores online, improving the detection performance to a great extent. However, the candidate boxes from Selective Search (Uijlings et al. 2013) are not accurate enough, hence a regression branch is added to refine the boxes (Gao et al. 2018; Yang, Li, and Dou 2019; Zeng et al. 2019). To find more credible boxes for online refinement, WSOD^2 (Zeng et al. 2019) combines bottom-up and top-down features while OIM (Lin et al. 2020) builds spatial and appearance graphs.

Memory Module. Memory module is widely used in various computer vision tasks. Wu *et al.* (Wu et al. 2018) build a large memory bank to store the feature vectors for unsupervised feature learning. Besides, Wu *et al.* (Wu et al. 2019) utilize long-term feature bank to store information about past and future scenes for video understanding. Wang *et al.* (Wang et al. 2020) construct a cross-batch memory embedding network to mine informative examples across multiple mini-batches for deep metric learning. Different from these methods, the class feature banks in our model record box-level features at the training stage to ameliorate MIL with cross-image information.

Proposed Method

In this section, we introduce our IM-CFB framework for weakly supervised object detection, which consists of four major components: MIL branch, Class Feature Banks (CFB) module, Feature Guided Instance Mining (FGIM) branch, and Online Instance Refinement (OIR) branch.

Framework Overview

For each image I, we denote its image-level label as $Y = [y_1, y_2, \cdots, y_C] \in \mathbb{R}^{C\times 1}$, where $y_c = 1$ or 0 indicates the presence or absence of class c. Let $R = \{R_1, R_2, \cdots, R_N\}$ denote the proposals extracted from Selective Search (Uijlings et al. 2013). Through ConvNet followed by RoI pooling and two fully connected layers, we obtain features $f \in \mathbb{R}^{D\times N}$ for all proposals. These features are first sent to the MIL branch to obtain proposal scores. Next, combined with these scores, FGIM mines credible positive samples utilizing the extra information from CFB. Then, these selected positive samples are both utilized to train the FGIM network and update CFB. Finally, credible region proposals are selected from the FGIM network, which are sent to OIR branch for further refinement. An overview of our framework is illustrated in Figure 2.

Multiple Instance Learning

We first introduce the MIL branch. Since we only have image-level annotations, it's hard to directly differentiate positive and negative proposals. To this end, we apply Multiple Instance Learning (MIL) in the CNN structure to detect objects following (Bilen and Vedaldi 2016).

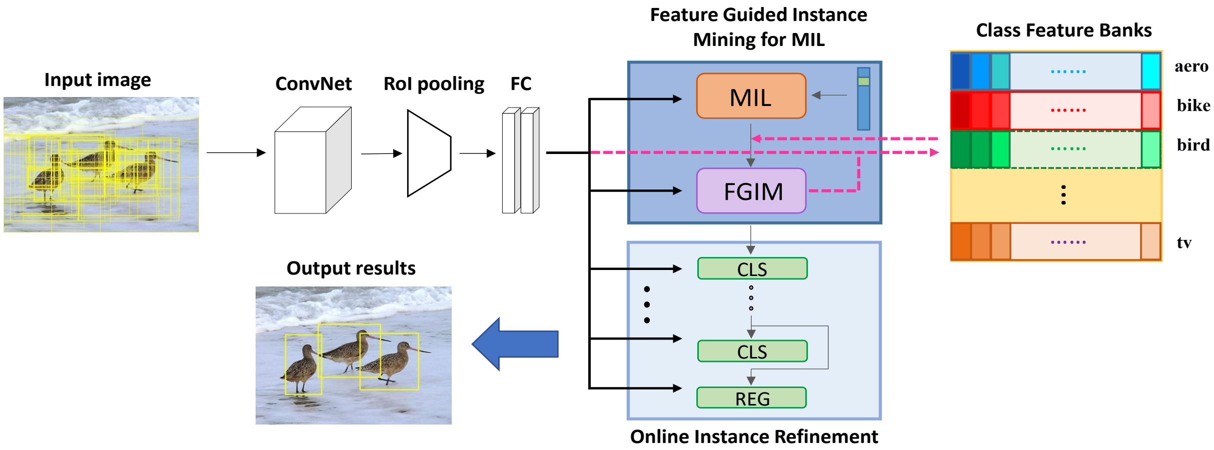

Figure 2: The overall framework of our method. MIL represents multiple instance learning and FGIM means feature guided instance mining. In Online Instance Refinement, CLS and REG represent classification refinement branch and regression branch, respectively. Through ConvNet followed by RoI pooling and two fully connected layers, features for all proposals can be obtained. These features are first sent to the MIL branch to obtain proposal scores. Next, combined with these scores, FGIM mines credible positive samples utilizing the extra information from CFB. Then, these selected positive samples are both utilized to train the FGIM network and update CFB. Finally, credible region proposals are selected from the FGIM network, which are sent to OIR branch for further refinement.

In MIL branch, the proposal features f are first fed into two sub-branches to produce two matrices X^{cls} and $X^{det} \in \mathbb{R}^{C\times N}$, which represent classification and detection scores, respectively. Then, the proposal scores are generated using element-wise product $X^{box} = X^{cls} \odot X^{det}$. Finally, we get image-level classification scores by summing the scores for all proposals of each class $X_c^{img} = \sum_{i=1}^{N} X_{c,i}^{box}$. Combined with the image-level label, we utilize cross entropy loss to train the MIL branch as follows:

$$\mathcal{L}_{mil} = -\sum_{c=1}^{C}\{y_c log X_c^{img} + (1-y_c) log(1-X_c^{img})\}. \tag{1}$$

Class Feature Banks

In different scenes, objects may vary greatly in appearances (*e.g.*, postures, shapes), even if they belong to the same class. However, in a weakly-supervised setting, deep neuronal networks may fail to learn the variations due to the lack of box-level annotations. To solve the problem, we propose a Class Feature Banks (CFB) module, which collects box-level information from different images, providing a comprehensive view for detecting objects of various appearances.

In general, CFB consists of C sub-banks, where C represents the number of categories. We choose a queue-like structure to represent our sub-bank, with a fixed length K. Formally, we denote our CFB module as $CFB = \{\Phi_1, \Phi_2, \cdots, \Phi_C\}$, where Φ_c is the sub-bank for the c-th category, which is further described as $\Phi_c = \left\{\hat{f}_1^c, \hat{f}_2^c, \cdots, \hat{f}_K^c\right\}$. We describe $\hat{f}_k^c$ as a *key* to class c, since it represents a certain type of appearance for this class. Moreover, for each key, we introduce two kinds of auxiliary information for assistance, *i.e.*, its updating iteration T_k^c and score S_k^c, representing its availability and credibility, respectively. We design CFB this way for two reasons. Firstly, each category has its particular kind of variations, hence we need to divide it into several class-wise parts. Secondly, with the network parameters updated in the training process, information stored in banks will be easily out of date. Meanwhile, we also need to control the capacity of each sub-bank since redundant information will cause much noise. Hence, a queue-like structure is an ideal choice. In addition, we choose features f as the source of box-level information. For a ground-truth class c, if a confident proposal R_{i_c} has been selected, its feature vector f_{i_c} will be absorbed into Φ_c.

With the help of the queue-like structure, the oldest features will be abandoned as new ones come in, following a First-In-First-Out (FIFO) rule. The fly in the ointment, however, is that while it can store cross-image information effectively, the information of intra-class diversity cannot be guaranteed well. To solve the problem, we design an updating algorithm to guarantee the diversity of each sub-bank. Suppose we can obtain this information from FGIM: a selected confident proposal R_{i_c}, which either has a similar appearance with a key $\hat{f}_k^c$ in Φ_c, or represents a new type of appearance. In the latter case, we continue to adopt FIFO strategy to update CFB. On the other hand, for the former case, we utilize a weighting strategy instead. Specifically, we first calculate a new feature vector by a combination of $\hat{f}_k^c$ and f_{i_c} weighted by their scores. Then, we replace $\hat{f}_k^c$ with the new feature vector and update its auxiliary information. The detailed algorithm is described in Algorithm 1.

Algorithm 1: CFB updating strategy

```
Input: Sub-bank $\Phi_c = \left\{\hat{f}_1^c, \hat{f}_2^c, \cdots, \hat{f}_K^c\right\}$ and its
       auxiliary information $T^c$, $S^c$; index of chosen
       proposal $i_c$ and its corresponding key $key_{i_c}$; feature
       vectors $f$; MIL scores $X^{box}$; current iteration $T_{cur}$
Output: Updated sub-bank $\Phi_c$
/* $key_{i_c} = -1$ represents bringing in a
   new type, utilizing FIFO strategy */
if $key_{i_c} == -1$ then
    if $|\Phi_c| < K$ then
        $k = |\Phi_c| + 1$;
    else
        $k = \arg\min_k T_k^c$;
    Update keys in $\Phi_c$ and their auxiliary information:
      $\hat{f}_k^c \leftarrow f_{i_c},\ T_k^c \leftarrow T_{cur},\ S_k^c \leftarrow X_{c,i_c}^{box}$;
/* Else, utilizing weighting strategy
   to update CFB */
else
    $k = key_{i_c}$;
    $ratio = S_k^c / (S_k^c + X_{c,i_c}^{box})$;
    Update keys in $\Phi_c$:
      $\hat{f}_k^c \leftarrow ratio \cdot \hat{f}_k^c + (1 - ratio) \cdot f_{i_c}$;
    Update auxiliary information in $\Phi_c$:
      $S_k^c \leftarrow ratio \cdot S_k^c + (1 - ratio) \cdot X_{c,i_c}^{box}$,
      $T_k^c \leftarrow T_{cur}$;
```

Algorithm 2: Feature Guided Instance Mining

```
Input: Region proposals $R = \{R_1, R_2, \cdots, R_N\}$; image
       labels $Y = [y_1, y_2, \cdots, y_C]$; feature vectors
       $F = \{f_1, f_2, \cdots, f_N\}$; MIL scores $X^{box}$; $CFB$
Output: Indexes of chosen proposals $P$ and their
        corresponding keys $KEY$; Positive Samples $PS$
for c in $[1, 2, \cdots, C]$ do
    if $y_c == 1$ then
        Fetch $\Phi_c$ from $CFB$;
        Calculate similarities between $F$ and $\Phi_c$ based on
          Eq.2;
        Choose the top-similarity proposal $\hat{i_c}$ and its
          corresponding key $key_{\hat{i_c}}$ using Eq.3 and Eq.4;
        Choose the top-scoring proposal $\widetilde{i_c}$ using Eq.5;
        Calculate the distance $d_{\hat{i_c},\widetilde{i_c}}$ using Eq.6;
        Calculate the average distance $\overline{d_{\widetilde{i_c}}}$ using Eq.7;
        $P \leftarrow \varnothing, KEY \leftarrow \varnothing, PS \leftarrow \varnothing$;
        $PS \leftarrow \widetilde{i_c}$;
        if $d_{\hat{i_c},\widetilde{i_c}} < \alpha\overline{d_{\widetilde{i_c}}}$ then
            $P \leftarrow \hat{i_c}, KEY \leftarrow key_{\hat{i_c}}, PS \leftarrow \hat{i_c}$;
        else
            $P \leftarrow \widetilde{i_c}, KEY \leftarrow -1$;
```

Feature Guided Instance Mining

In WSOD with MIL, the detection results from the MIL branch X^{box} are utilized to choose confident region proposals for refinement. However, those results are usually noisy. To this end, we apply Feature Guided Instance Mining (FGIM) as a supplement to MIL to obtain more reliable detection results.

Given an image I and a set of region proposals $R = \{R_1, R_2, \cdots, R_N\}$, we obtain their feature vectors $F = \{f_1, f_2, \cdots, f_N\}$ from the fully connected layer after RoI pooling. Meanwhile, for a ground-truth category c, we fetch the sub-bank $\Phi_c = \left\{\hat{f}_1^c, \hat{f}_2^c, \cdots, \hat{f}_K^c\right\}$ from CFB. First, for each proposal R_i and its corresponding feature vector f_i, we calculate its similarities with all keys stored in Φ_c using the cosine distance as follows,

$$S_{i,k}^c = \frac{f_i \cdot \hat{f}_k^c}{||f_i|| \cdot ||\hat{f}_k^c||}. \tag{2}$$

S^c measures the similarities between proposal features and keys from Φ_c, where the latter is a credible *priori* for c-th class, hence it can be regarded as a score that measures how likely the proposal belongs to this category. Meanwhile, considering the intra-class diversity, we denote the score of R_i as its similarity with its closest key,

$$S_i^c = \max_k S_{i,k}^c, \quad \hat{K}_i^c = \arg\max_k S_{i,k}^c. \tag{3}$$

Furthermore, the cosine distance is normalized, which allows us to compare similarity scores among the proposals. Hence, we select the top-similarity proposal as the positive sample, and find its corresponding key in Φ_c,

$$\hat{i_c} = \arg\max_i S_i^c, \quad key_{\hat{i_c}} = \hat{K}_{\hat{i_c}}^c. \tag{4}$$

Given the selected sample $\hat{i_c}$, the most direct way is to use its feature vector $f_{\hat{i_c}}$ to update $key_{\hat{i_c}}$ in CFB, as described in the previous section. However, in this way, noisy samples will be selected inevitably, especially when CFB is in the initial stages of construction. Improper selections will not only exert a negative influence on the subsequent FGIM network training, but also provide CFB with background information incorrectly, which will in turn harm the effectiveness of FGIM. Therefore, we add a constraint when utilizing similarity scores to choose positive samples, with regard to the intra-class similarity. Inspired by (Lin et al. 2020), we first calculate the feature distance between $\hat{i_c}$ and the top-scoring proposal $\widetilde{i_c}$ from MIL branch,

$$\widetilde{i_c} = \arg\max_i X_{c,i}^{box}, \tag{5}$$

$$d_{\hat{i_c},\widetilde{i_c}} = ||f_{\hat{i_c}} - f_{\widetilde{i_c}}||. \tag{6}$$

Next, we calculate the average distance between $\widetilde{i_c}$ and its surrounding proposals,

$$\overline{d_{\widetilde{i_c}}} = mean_i(||f_i - f_{\widetilde{i_c}}||), \quad s.t.\ IoU(i, \widetilde{i_c}) > \tau. \tag{7}$$

$\overline{d_{\widetilde{i_c}}}$ can be regarded as the average intra-class similarity of the c-th class in image I. Hence, if the selected sample $\hat{i_c}$ meets the condition that $d_{\hat{i_c},\widetilde{i_c}} < \alpha\overline{d_{\widetilde{i_c}}}$, we consider it credible. Otherwise, we use the top-scoring proposal $\widetilde{i_c}$ to update CFB instead, and regard it as a new type for category c, which ensures the diversity of CFB. The detailed algorithm is described in Algorithm 2.

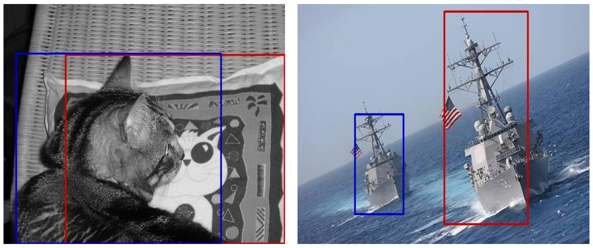

Figure 3: The examples of FGIM. Red boxes represent top-scoring proposals, while blue boxes are selected following the guidance of CFB. Both of them are utilized to train the FGIM network and update CFB in turn.

For each ground-truth class, both $R_{\tilde{i}}$ and $R_{\hat{i}}$ satisfying the constraint above will be selected as positive samples to train the FGIM network. Specifically, for each proposal R_i, we compute its maximum IoU I_i with the selected positive samples. We denote the positive proposals as $R_{pos} = \{R_i | I_i \geq 0.5\}$ and the negative proposals as $R_{neg} = \{R_i | 0.1 \leq I_i < 0.5\}$. Positive proposals are labeled as the same class as their closest positive samples, while negative proposals are labeled with $C + 1$. Therefore, for each proposal R_i, we can obtain its pseudo-label $Y_i = [y_{1,i}, y_{2,i}, \cdots, y_{C+1,i}]$.

The output of the FGIM network can be denoted as $x \in \mathbb{R}^{(C+1) \times N}$, where $x_{c,i}$ represents the confidence of the proposal i belonging to the class c. Combined with pseudo-labels, the loss for the FGIM network is denoted as:

$$\mathcal{L}_{fgim} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C+1} w_i y_{c,i} log x_{c,i}, \tag{8}$$

where the loss weight w_i for proposal R_i is calculated following (Tang et al. 2017).

As illustrated in Figure 3, $R_{\hat{i}}$ acts as a refinement or supplement for $R_{\tilde{i}}$, hence combining them together exerts a good influence on both training FGIM network and updating CFB, which will in turn help refine themselves.

Online Instance Refinement

Following the general pipelines in (Tang et al. 2017; Yan et al. 2019; Yang, Li, and Dou 2019), we add several classification refinement branches after FGIM and select the top-scoring region proposal as the initial positive seed for each ground-truth class. In contrast to the previous work, we utilize the results from the FGIM network instead. The loss for each branch has the same formulation with $\mathcal{L}_{fgim}$. Moreover, we add a regression branch sibling to the last classification branch to further refine the proposals. The regression branch is a RCNN-like structure, which contains classification and regression parts. For each proposal i, the regression part predicts offsets of the positions and shapes $t_i^c = (t_x^c, t_y^c, t_w^c, t_h^c)$ for a ground-truth class c, and its pseudo-label $\hat{t_i^c}$ can be calculated with its corresponding positive seed. We apply weighted smooth-L1 loss for regression,

$$\mathcal{L}_{regress} = -\frac{1}{N} \sum_{i=1}^{N} \sum_{c=1}^{C} \mathbb{I}(y_{c,i} = 1) w_i smooth_{L1}(t_i^c, \hat{t_i^c}). \tag{9}$$

Finally, we train the network end-to-end by combining all the losses mentioned before as in Eq. 10,

$$\mathcal{L}_{total} = \mathcal{L}_{mil} + \mathcal{L}_{fgim} + \sum_{t=1}^{T} \mathcal{L}_{ref}^{t} + \mathcal{L}_{reg}. \tag{10}$$

Experiments and Analysis

Datasets

We evaluate our proposed method on both Pascal VOC 2007 and Pascal VOC 2012 (Everingham et al. 2010) following previous WSOD works. PASCAL VOC 2007 and 2012 consist of 9,962 and 22,531 images, respectively, and both of them contain 20 categories. We train on *trainval* split (5,011 images for VOC 2007, 11,540 images for VOC 2012), and apply two kinds of metrics to evaluate the localization accuracy: 1) Average Precision (AP) and the mean of AP (mAP) on the *test* split; 2) Correct localization (CorLoc) on the *trainval* split.

Implementation Details

For the sake of fair comparison, we adopt VGG16 (Simonyan and Zisserman 2014) pre-trained on Imagenet (Deng et al. 2009) as the backbone and utilize Selective Search to generate object proposals following previous work. FGIM network consists of a fully connected layer followed by a softmax operation. The batch size, momentum, and weight decay are set to 4, 0.9, and 5×10^{-4} respectively. The learning rate is set to 1×10^{-3} for the first 60K iterations and 1×10^{-4} for the following 20K iterations. For data augmentation, we use five image scales, *i.e.*, $\{480, 576, 688, 864, 1200\}$ for the shortest side of images, and random horizontal flipping is applied. We set $K = 6$ for the VOC 2007 and $K = 8$ for the VOC 2012. We set $\tau = 0.5$ for both datasets. Considering that the stability of the network increases during the training process, we set $\alpha = 5$ at the first 40K iterations and then tighten the restriction with $\alpha = 2$. The number of refinement branches T is set to 2. For inference, we combine both proposal scores of FGIM and OIR and then calculate their average as the final scores. Our experiments are implemented based on PyTorch on NVIDIA GTX 1080Ti GPUs.

Comparison with State-of-the-arts

We compare the results of our method with other works in this subsection. Table 1 and Table 2 show the performance on VOC 2007 dataset, where FRCNN means re-training a Fast-RCNN detector utilizing the results produced by WSOD methods. Our method obtains 54.3% on mAP with a single VGG16 model, which outperforms all the other single model methods by at least 0.8% mAP. Moreover, our single model even surpasses all previous methods re-trained with FRCNN. More remarkably, after re-training,

Method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
OICR	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
PCL	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	63.0	43.5
SDCN	59.8	67.1	32.0	34.7	22.8	67.1	63.8	67.9	22.5	48.9	47.8	60.5	51.7	65.2	11.8	20.6	42.1	54.7	60.8	64.3	48.3
C-MIL	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
Yang *et al.*	57.6	70.8	50.7	28.3	27.2	72.5	69.1	65.0	26.9	64.5	47.4	47.7	53.5	66.9	13.7	29.3	56.0	54.9	63.4	65.2	51.5
C-MIDN	53.3	71.5	49.8	26.1	20.3	70.3	69.9	68.3	28.7	65.3	45.1	64.6	58.0	**71.2**	20.0	27.5	54.9	54.9	69.4	63.5	52.6
WSOD^2	65.1	64.8	**57.2**	**39.2**	24.3	69.8	66.2	61.0	29.8	64.6	42.5	60.1	71.2	70.7	21.9	28.1	**58.6**	59.7	52.2	64.8	53.6
OIM	55.6	67.0	45.8	27.9	21.1	69.0	68.3	70.5	21.3	60.2	40.3	54.5	56.5	70.1	12.5	25.0	52.9	55.2	65.0	63.7	50.1
SLV	65.6	71.4	49.0	37.1	24.6	69.6	70.3	70.6	**30.8**	63.1	36.0	61.4	65.3	68.4	12.4	**29.9**	52.4	60.0	67.6	64.5	53.5
IM-CFB	64.1	74.6	44.7	29.4	26.9	73.3	72.0	71.2	28.1	66.7	48.1	63.8	55.5	68.3	17.8	27.7	54.4	**62.7**	70.5	66.6	**54.3**
Pred Net (FRCNN)	**66.7**	69.5	52.8	31.4	24.7	**74.5**	**74.1**	67.3	14.6	53.0	46.1	52.9	69.9	70.8	18.5	28.4	54.6	60.7	67.1	60.4	52.9
C-MIL+FRCNN	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
C-MIDN+FRCNN	54.1	74.5	56.9	26.4	22.2	68.7	68.9	74.8	25.2	64.8	46.4	70.3	66.3	67.5	21.6	24.4	53.0	59.7	68.7	58.9	53.6
OIM+FRCNN	53.4	72.0	51.4	26.0	27.7	69.8	69.7	74.8	21.4	67.1	45.7	63.7	63.7	67.4	10.9	25.3	53.5	60.4	70.8	58.1	52.6
SLV+FRCNN	62.1	72.1	54.1	34.5	25.6	66.7	67.4	**77.2**	24.2	61.6	47.5	**71.6**	**72.0**	67.2	12.1	24.6	51.7	61.1	65.3	60.1	53.9
IM-CFB+FRCNN	63.3	**77.5**	48.3	36.0	**32.6**	70.8	71.9	73.1	29.1	**68.7**	47.1	69.4	56.6	70.9	**22.8**	24.8	56.0	59.8	**73.2**	64.6	**55.8**

Table 1: Comparison with the state-of-the-arts in terms of mAP (%) on the VOC 2007 *test* set.

Method	CorLoc(%)
OICR(Tang et al. 2017)	60.6
PCL(Tang et al. 2018)	62.7
SDCN (Li et al. 2019)	66.8
C-MIL (Wan et al. 2019)	65.0
Yang *et al.* (Yang, Li, and Dou 2019)	68.0
C-MIDN (Yan et al. 2019)	68.7
WSOD^2 (Zeng et al. 2019)	69.5
OIM (Lin et al. 2020)	67.2
SLV (Chen et al. 2020)	**71.0**
IM-CFB	70.7
SDCN+FRCNN (Li et al. 2019)	68.8
Pred Net (FRCNN) (Arun et al. 2019)	70.9
C-MIDN+FRCNN (Yan et al. 2019)	71.9
OIM+FRCNN (Lin et al. 2020)	68.8
SLV+FRCNN (Chen et al. 2020)	72.0
IM-CFB+FRCNN	**72.2**

Table 2: Comparison with the state-of-the-arts in terms of CorLoc (%) on the VOC 2007 *trainval* set.

Method	mAP(%)	CorLoc(%)
PCL (Tang et al. 2018)	40.6	63.2
SDCN (Li et al. 2019)	43.5	67.9
C-MIL(Wan et al. 2019)	46.7	67.4
Yang *et al.* (Yang, Li, and Dou 2019)	46.8	69.5
WSOD^2 (Zeng et al. 2019)	47.2	**71.9**
OIM (Lin et al. 2020)	45.3	67.1
SLV (Chen et al. 2020)	49.2	69.2
IM-CFB	**49.4**	69.6

Table 3: Comparison with the state-of-the-arts in terms of mAP (%) on the VOC 2012 *test* set and CorLoc (%) on the VOC 2012 *trainval* set using a single model.

our method can further achieve 55.8% on mAP and 72.2% on CorLoc, which are the new state-of-arts.

Compared with recent works, *e.g.*, (Tang et al. 2018; Yang, Li, and Dou 2019), which utilize results from MIL branch to select positive seeds directly, our work introduces extra box-level features to refine MIL branch, which enables to select more accurate seeds and outperforms (Yang, Li, and Dou 2019) by 2.8% mAP. WSOD^2 (Zeng et al. 2019) and OIM (Lin et al. 2020) also utilize features as extra information, but both of them focus on objects in a single image. In contrast, our work proposes CFB to collect class-wise cross-image information, which provides a broader view for each category, hence achieves a better performance.

We also evaluate our work on VOC 2012 dataset. Table 3 shows the mAP and CorLoc results using a single model, which validates the effectiveness of our work.

Figure 4 shows the detection results on VOC 2007 *test* set. The first two rows indicate that our method can correctly detect diverse objects, *e.g.*, "car", "dog", even if they are in some complex backgrounds. Some failure cases are shown in the last row, containing localizing the most discriminative parts, grouping multiple objects, and containing background, especially for "person" class.

Ablation Study

We conduct ablation experiments on PASCAL VOC 2007 to prove the effectiveness of our proposed network.

Effect of each component Table 4 shows the effectiveness of each component, where OIR means the online instance refinement network mentioned in Sec 3.5. We start from a basic model only containing MIL. Next, we extend the base model by adding our proposed FGIM, improving the mAP from 34.8% to 38.0%. The results indicate that, by introducing diversity information, our FGIM enhances the capacity of detection to a great degree. Furthermore, "MIL + OIR" can be seen as a stronger baseline for our work. Adding FGIM can further lead to a boost of 2.2% mAP. The results verify that, when providing more reliable seeds by utilizing FGIM (as shown in Figure 5) for further refinement, the performance of the whole detector is able to reach a higher ceiling.

Updating methods for CFB As discussed in the previous section, two methods can be applied to update CFB. Applying First-In-First-Out (FIFO) strategy is the simplest way, which is the original updating policy for queues. In order to guarantee the diversity of keys in CFB, we propose our

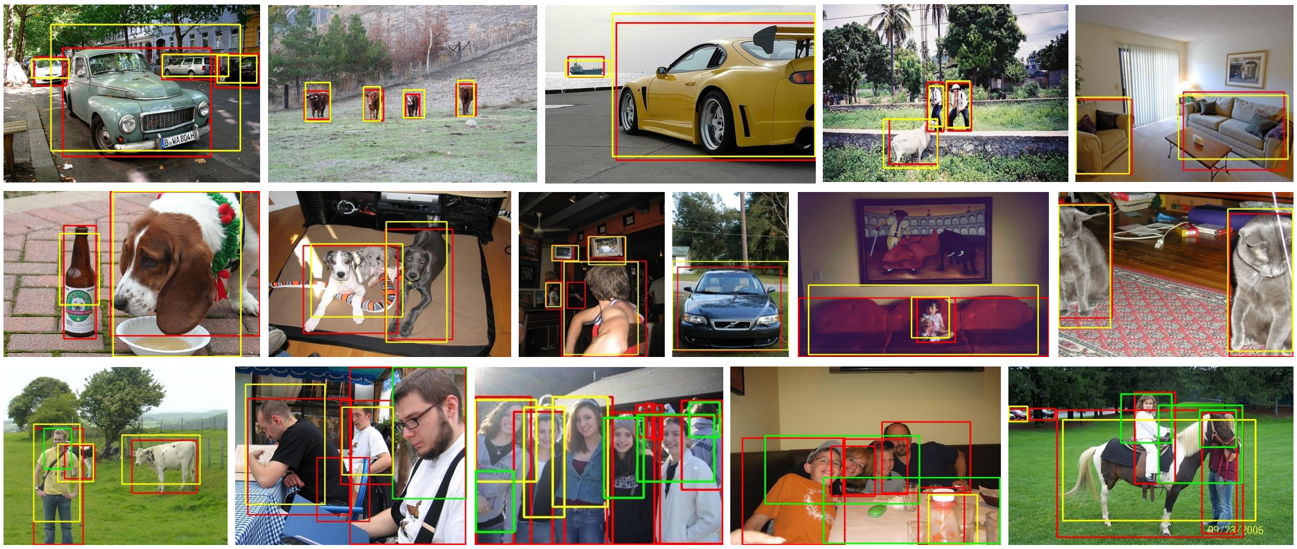

Figure 4: Detections results on VOC 2007 *test* set. Boxes in red, yellow, and green represent ground-truth boxes, successful predictions, and failure cases respectively. We apply NMS first and show all detections with scores ≥ 0.2.

MIL	OIR	FGIM	mAP(%)
✓			34.8
✓		✓	38.0
✓	✓		52.1
✓	✓	✓	**54.3**

Table 4: Ablation study of different components of our method on VOC 2007 dataset.

FIFO	Weighting	mAP(%)
		52.1
✓		52.6
	✓	**54.3**

Table 5: Impact of updating methods for CFB on VOC 2007 dataset.

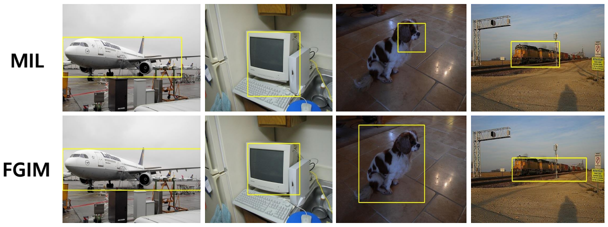

Figure 5: Seeds obtained from MIL (top part) and FGIM (bottom part) at 20k Iteration.

K	2	4	6	8	10	20
mAP(%)	51.5	52.7	**54.3**	53.1	52.3	51.3

Table 6: Impact of length K in CFB on VOC 2007 dataset.

weighting strategy. Table 5 shows a comparison between the two methods. Based on the original model, adding FIFO-based CFB brings 0.5% mAP improvement, which attributes to introduction of the extra cross-image box-level information, but the improvement is limited. Compared to FIFO, 2.2% mAP gain can be achieved by our weighting strategy, which indicates the effectiveness of saving diversity information. In addition, the weighting strategy can also suppress the impact of incorrectly bringing in noisy samples.

Length of sub-bank in CFB Table 6 indicates that $K=6$ is the optimal length for CFB. If the length is too small, diversity will be harmed, resulting in less kind of objects collected. If the length is too large, some noisy information will be absorbed and background proposals will be selected incorrectly.

Conclusion

In this paper, we present an Instance Mining with Class Feature Banks (IM-CFB) framework that enables to store and utilize class-wise information for weakly supervised object detection. Considering the intra-class diversity, the Class Feature Banks (CFB) module is designed to record and update box-level information online, bringing a broader view for each category. Leveraging the features recorded in the CFB, the Feature Guided Instance Mining (FGIM) algorithm is introduced to ameliorate the region proposal selection of the MIL branch. Extensive experiments conducted on two benchmark datasets, *i.e.* PASCAL VOC 2007 and 2012, demonstrate the effectiveness of our method.

Acknowledgements

The work of Houqiang Li was supported by NSFC under Contract 61836011. The work of Wengang Zhou was supported in part by NSFC under Contract 61822208, 61632019 and 62021001, and in part by the Youth Innovation Promotion Association CAS under Grant 2018497.

References

Arun, A.; Jawahar, C.; and Kumar, M. P. 2019. Dissimilarity coefficient based weakly supervised object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 9432–9441.

Bilen, H.; and Vedaldi, A. 2016. Weakly supervised deep detection networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2846–2854.

Chen, Z.; Fu, Z.; Jiang, R.; Chen, Y.; and Hua, X.-S. 2020. SLV: Spatial Likelihood Voting for Weakly Supervised Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 12995–13004.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A Large-Scale Hierarchical Image Database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 248–255.

Diba, A.; Sharma, V.; Pazandeh, A.; Pirsiavash, H.; and Van Gool, L. 2017. Weakly supervised cascaded convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 914–922.

Everingham, M.; Van Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2010. The pascal visual object classes (voc) challenge. *International journal of computer vision (IJCV)* 88(2): 303–338.

Gao, M.; Li, A.; Yu, R.; Morariu, V. I.; and Davis, L. S. 2018. C-wsl: Count-guided weakly supervised localization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 152–168.

Gao, Y.; Liu, B.; Guo, N.; Ye, X.; Wan, F.; You, H.; and Fan, D. 2019. Utilizing the Instability in Weakly Supervised Object Detection. *arXiv preprint arXiv:1906.06023* .

Girshick, R. 2015. Fast r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 1440–1448.

Kim, D.; Lee, G.; Jeong, J.; and Kwak, N. 2020. Tell Me What They're Holding: Weakly-Supervised Object Detection with Transferable Knowledge from Human-Object Interaction. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 11246–11253.

Li, X.; Kan, M.; Shan, S.; and Chen, X. 2019. Weakly supervised object detection with segmentation collaboration. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 9735–9744.

Lin, C.; Wang, S.; Xu, D.; Lu, Y.; and Zhang, W. 2020. Object Instance Mining for Weakly Supervised Object Detection. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 11482–11489.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2980–2988.

Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; and Berg, A. C. 2016. Ssd: Single shot multibox detector. In *European conference on computer vision (ECCV)*, 21–37.

Redmon, J.; Divvala, S.; Girshick, R.; and Farhadi, A. 2016. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 779–788.

Ren, S.; He, K.; Girshick, R.; and Sun, J. 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems (NeurIPS)*, 91–99.

Ren, Z.; Yu, Z.; Yang, X.; Liu, M.-Y.; Lee, Y. J.; Schwing, A. G.; and Kautz, J. 2020. Instance-Aware, Context-Focused, and Memory-Efficient Weakly Supervised Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 10598–10607.

Shen, Y.; Ji, R.; Zhang, S.; Zuo, W.; and Wang, Y. 2018. Generative Adversarial Learning Towards Fast Weakly Supervised Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5764–5773.

Simonyan, K.; and Zisserman, A. 2014. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv preprint arXiv:1409.1556* .

Tang, P.; Wang, X.; Bai, S.; Shen, W.; Bai, X.; Liu, W.; and Yuille, A. L. 2018. Pcl: Proposal cluster learning for weakly supervised object detection. *IEEE transactions on pattern analysis and machine intelligence (TPAMI)* 42(1): 176–191.

Tang, P.; Wang, X.; Bai, X.; and Liu, W. 2017. Multiple instance detection network with online instance classifier refinement. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2843–2851.

Uijlings, J. R.; Van De Sande, K. E.; Gevers, T.; and Smeulders, A. W. 2013. Selective search for object recognition. *International journal of computer vision (IJCV)* 104(2): 154–171.

Wan, F.; Liu, C.; Ke, W.; Ji, X.; Jiao, J.; and Ye, Q. 2019. C-mil: Continuation multiple instance learning for weakly supervised object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2199–2208.

Wang, X.; Zhang, H.; Huang, W.; and Scott, M. R. 2020. Cross-Batch Memory for Embedding Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 6388–6397.

Wei, Y.; Shen, Z.; Cheng, B.; Shi, H.; Xiong, J.; Feng, J.; and Huang, T. 2018. Ts2c: Tight box mining with surrounding segmentation context for weakly supervised object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 434–450.

Wu, C.-Y.; Feichtenhofer, C.; Fan, H.; He, K.; Krahenbuhl, P.; and Girshick, R. 2019. Long-term feature banks for detailed video understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 284–293.

Wu, Z.; Xiong, Y.; Yu, S. X.; and Lin, D. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3733–3742.

Yan, G.; Liu, B.; Guo, N.; Ye, X.; Wan, F.; You, H.; and Fan, D. 2019. C-MIDN: Coupled Multiple Instance Detection Network With Segmentation Guidance for Weakly Supervised Object Detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 9833–9842.

Yang, K.; Li, D.; and Dou, Y. 2019. Towards precise end-to-end weakly supervised object detection network. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 8372–8381.

Zeng, Z.; Liu, B.; Fu, J.; Chao, H.; and Zhang, L. 2019. Wsod2: Learning bottom-up and top-down objectness distillation for weakly-supervised object detection. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 8292–8300.

Zhang, X.; Feng, J.; Xiong, H.; and Tian, Q. 2018. Zigzag learning for weakly supervised object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4262–4270.