Few-Example Object Detection with Model Communication

Xuanyi Dong, Liang Zheng, Fan Ma, Yi Yang, Deyu Meng

Abstract—In this paper, we study object detection using a large pool of unlabeled images and only a few labeled images per category, named “few-example object detection”. The key challenge consists in generating trustworthy training samples as many as possible from the pool. Using few training examples as seeds, our method iterates between model training and high-confidence sample selection. In training, easy samples are generated first and, then the poorly initialized model undergoes improvement. As the model becomes more discriminative, challenging but reliable samples are selected. After that, another round of model improvement takes place. To further improve the precision and recall of the generated training samples, we embed multiple detection models in our framework, which has proven to outperform the single model baseline and the model ensemble method. Experiments on PASCAL VOC’07, MS COCO’14, and ILSVRC’13 indicate that by using as few as three or four samples selected for each category, our method produces very competitive results when compared to the state-of-the-art weakly-supervised approaches using a large number of image-level labels.

Index Terms—few-example learning, object detection, convolutional neural network

1 INTRODUCTION

This paper considers the problem of generic object detection with very few training examples (bounding boxes) per class, named “few-example object detection (FEOD)”. Existing works on supervised/semi-supervised/weakly-supervised object detection usually assume much more annotations than this paper. Specifically, we annotate all the bounding boxes in such a number of images that each class will only have 3-4 annotated examples. This task is extremely challenging due to the scarcity of labels which leads to the difficulty in label propagation and model training.

We provide a brief discussion on the relationship between FEOD and other types of supervisions, excluding the methods using strong labels [13], [14], [15], [16], [17], [18]. First, strictly speaking, FEOD is a semi-supervised task. But to the best of our knowledge, most works on semi-supervised object detection (SSOD) assume around 50% of all the labeled bounding boxes [4], [5], [6]. These methods assume that some classes have strong bounding box labels, while other classes have weak image-level labels [4], [5], [6], [19]. Therefore, FEOD is distinctive from SSOD in terms of the small number of required labels. Second, weakly supervised object detection (WSOD) usually relies on image-level labels [13], [14], [15], [16], [17], a type of supervision that is distinct from bounding box level labels as used in FEOD. An advantage of FEOD over WSOD is that the labeling effort of FEOD is much smaller. In this paper, we mainly compare our method with the state-of-the-art WSOD works. The third category leverages tracking to mine labels from videos [2], [3]. Usually, these methods focus on moving objects, e.g., car and bicycle, which can be tracked based on their motions along time. So a potential problem of this category of methods is its effectiveness on stationary objects, e.g., table and sofa, for which tracking may be infeasible.

Table 1 presents a brief summary of the types of supervision used in previous related object detection methods.

Therefore, comparing with the other types of supervision listed in Table 1, the advantage of FEOD is mainly fourfold. First, FEOD reduces the labeling effort by using only several annotated bounding boxes per class. Second, FEOD provides robust supervision to rare classes such as Dugong, where only a few training images can be found. For these classes, image-level supervision on the limited number of images is always not enough to train a good detector. Third, FEOD can deal with stationary objects, so that it has a larger application scope. Fourth, FEOD provides accurate annotations to crowded objects, while models trained with image-level labels usually perform poorly on the crowded objects, such as people and bottle. In comparison, using a few images with bounding box annotations, FEOD can enhance the detector to be robust to such crowded objects. This can be seen in our experiments. Table 2 evidently shows that the best weakly-supervised algorithm can only achieve 24.7% mAP on the class of person, but MSPLD achieves 40.1% mAP. In this paper, we explore the setting in which there is no motion information and no image-level supervision, and there are only several instance-level annotations. Under this setting, FEOD is extremely challenging due to the lack of labels. Addressing this challenging yet interesting task is the focus of this paper.

To be specific, the major challenges are: (1) generating reliable pseudo-annotated samples (high precision), and (2) finding possibly many newly annotated samples (high recall). Specifically, on the one hand, the training samples should
be generated with high confidence, i.e., a high precision
to guarantee sound guidance for detector training in the
following process. On the other hand, since more training
samples benefit a more discriminative detector, we speculate
that the generated training samples should have high recall
to provide sufficient knowledge for detector amelioration.
A trade-off clearly exists between the precision and recall
requirements.

In this paper, two seamlessly integrated solutions, self-
paced learning and multi-modal learning, are used to
achieve high precision and recall during training sample
generation. In a nutshell, with the training iterations, the
selected training images go from “easy” (with relatively
high confidence) to “hard”, and the object detector is gradu-
ally promoted. First, a self-paced learning (SPL) framework,
in its optimization process, selects “easy” training samples
and avoids noisy instances. Second, we embed multi-modal
learning in the SPL. Multiple detection models are incor-
porated in the learning process. Learning from multiple
models accomplishes two goals. (1) It helps alleviate the
local minimum issue of the model training, and (2) it im-
proves the precision and recall of training sample generation
due to knowledge compensation between multiple models.
Note that, since the multiple detection models are jointly
optimized, our experiments show that multi-modal learning
is far superior to model ensembles. In addition, prior
knowledge, i.e., confidence filtration and non-maximum
suppression, can be injected into this learning scheme to
further improve the quality of selected training samples.

The major points of this work are outlined below:

(1) We address object detection from a new perspective: using very few annotated bounding boxes per class. We propose to alternate between detector improvement and reliable sample generation, thereby gradually obtaining a stable yet robust detector.

(2) To ameliorate the trade-off between precision and recall in training sample generation, we embed multiple detection models in a unified learning scheme. In this manner, our method fully leverages the mutual benefit between multiple features and the corresponding multiple detectors.

(3) Our proposed algorithm is capable of producing competitive accuracy to state-of-the-art WSOD algorithms, which require much more labeling efforts.

2 RELATED WORK

2.1 Supervised object detection

Object detection methods based on convolution neural
networks (CNNs) can be divided into two types: proposal-
based and proposal-free [14], [15], [16], [17], [20], [21].
The road-map of proposal-based methods starts from R-
CNN [20] and is improved by SPP-Net [22] and Fast R-
CNN [14] in terms of accuracy and speed. Later, Faster
R-CNN [17] uses the region proposal network to quickly
generate object regions, achieving a high recall compared
to previous methods [23], [24]. Many methods directly predict
bounding boxes without generating region proposals [15],
[16], [25]. For example, YOLO [16] uses the whole feature
map from the last convolution layer. SSD [15] makes
improvements by leveraging default boxes of different aspect
ratios for multiple feature maps. These supervised methods
require strong supervision, which is relatively expensive to
obtain in practice.

2.2 Semi-supervised object detection

Current SSOD literature usually uses both the image-
level labels and some of the bounding box labels. For example, Yang et al. [26] design methods to learn video-specific features to boost detection performance. Liang et al. [1] propose an elegant method by integrating prior knowledge modeling, exemplar learning and video context learning for the SSOD task. They utilize around 350k images with bounding box annotations to provide a good initialization for fine-tuning the detection model on PASCAL VOC. Besides, they use a negative dataset (without the 20 classes on VOC) as well as around 20k labeled videos. In comparison, our algorithm only requires 3-4 bounding boxes of the target
classes, e.g., 20 classes on PASCAL VOC, and do not use
any outsider dataset. Misra et al. [3] start training with
some instance-level annotations and iteratively learn more
instances by fusing detection and tracking information. In
[2], discriminative visual regions are assigned with pseudo-
labels by matching and retrieving technique. Compared
with them, we do not need any extra supervised auxiliary
knowledge and the required amount of given annotations is kept at a extremely low level.

2.3 Weakly supervised object detection

The WSOD setting is to utilize the image-level label of each image to train object detectors. Some works employ off-the-shelf CNN models [7, 27, 28, 29, 30, 31]. For example, Shi et al. [11] employ multiple instance learning (MIL) to train support vector machine (SVM) classifiers in the order of object sizes. Others design new CNN architectures to obtain object information from the classification loss and leverage this classification model to derive object detectors [8, 9, 10, 32]. Bilen et al. [8] propose a weakly supervised detection network using selective search (SS) to generate proposals and train image-level classification based on regional features. Li et al. [9] train an image-level classifier to adapt detection results through a mask-out strategy and MIL. Tang et al. [33] integrate a multiple instance detection network and multi-stage instance classifiers in a single network, in which the results of one stage can be used as supervision for the next stage. Ge et al. [34] propose a weakly supervised curriculum pipeline to jointly optimize recognition, detection, and segmentation, so that multi-task learning enhances the detection performance. The aforementioned methods depart from our method in that image-level labels are used, which are still expensive to collect when compared with our scheme.

2.4 Object detection from few examples

A limited number of previous works can be classified into our settings. Wang et al. [12] propose to generate a large number of object detectors from few samples by model recommendation. However, they use 10-100 training samples per class, and their initial detectors are required to be trained on other large-scale detection datasets. Compared to previous methods [1], [12], [35], our approach only requires 2-4 examples per class without any extra training datasets.

Here we also briefly introduce and contrast few-shot learning and semi-supervised learning with the few-example learning setting. On the one hand, few-shot learning [36], [37], [38], [39], [40] aims to learn a model based on a few training examples without unlabeled data. In contrast, learning from few samples [12], [35], [41] usually learns an initial model based on the few labeled data, and then progressively ameliorate the initial model on unlabeled data. An important difference between few-example and few-shot learning is whether to use the unlabeled data. On the other hand, semi-supervised learning [1], [26] also leverages a portion of the annotations, which is similar to few-shot learning and few-example learning. However, semi-supervised learning can use a relatively large number of annotations (e.g., 50% of the full annotations), which is different from few-example learning and few-shot learning. We also note that semi-supervised learning can also use only a few annotations. In this scenario, few-example learning is a special case of semi-supervised learning.

2.5 Webly supervised learning for object detection

It can also reduce the annotation cost by leveraging web data. Chen et al. [42] propose a two-step approach to initialize the CNN models from easy sample first, and then adapt it to more realistic images. Divvala et al. [43] propose a fully-automated approach for learning extensive models for a wide range of variations via webly supervised learning, while their system requires lots of collection and training time. Besides, the algorithm can not obtain a good detection model even with 10 million automatically annotated images. Co-localization algorithms [44] localize the objects of the same class across a set of distinct images. They usually leverage the Internet images and are also able to detect objects, but require a strong prior that the image set contains objects with the same class. Some researchers [45], [46] propose an unsupervised algorithm to discover the common objects from large image collections via the Internet search. They usually assume the clean algorithm, but for most object classes, this assumption is unrealistic in real-world settings.

2.6 Zero-shot Object Detection

Zero-shot object detection (ZSD) [47], [48], [49] aims to locate object instances belonging to novel categories without any training examples. Rahman et al. [47] propose a deep network to model the interplay between visual and semantic domain information jointly. Bansal et al. [48] adopt visual-semantic embeddings for ZSD, and provide novel splits and baseline experiments on MSCOCO and Visual Genome [50]. ZDS is a very challenging task and has many potential research possibilities. The focus of this paper is not on detecting the new categories of objects like ZDS, while on extracting detectors from extremely few training samples for each class of objects. Thus their purposes are different.

2.7 Model ensemble

Ensemble methods are widely used. Dai et al. [51] ensemble multiple part detectors to form sub-structure detectors, which further constitute the final object detector. Their ensemble model can only handle a specific class and needs a relatively long training time, e.g., more than 400 hours on PASCAL VOC 2007 [52]. The algorithm of [53] is based on the linear SVM classifier, which is limited to using the off-the-shelf features. Bilen et al. [8] first train three detection models with different architectures and then averagely fuse them. Many previous detection methods [8], [51], [53] employ model ensemble as post-processing. However, without considering the multiple models in training, these methods may not fully utilize the complementary nature of different detection models. In this paper, we jointly optimize multiple detection models during training to further improve each model.

2.8 Progressive paradigm

Our method adapts a progressive strategy to iteratively optimize the multiple detection models, which is related to curriculum learning [54] and self-paced learning [55]. Bengio et al. [54] first propose a learning paradigm in which organizing the examples in a meaningful order significantly improves the performance. Kumar et al. [55] propose to determine the training sample order by how easy they are. Wang et al. [56] propose an approach to learn novel
3 The Proposed Method

As our framework combines self-paced learning and multi-modal learning, we call it multi-modal self-paced learning for detection (MSPLD). We first introduce some basic notations in Sec.3.1, and demonstrate the detailed algorithm description in Sec.3.4. Lastly, we show the whole optimization method in Sec.3.3. Then, we describe the optimization method in Sec.3.3. Consequently, we obtain a significantly improvement in object detection from few examples.

3.1 Preliminaries

We choose Fast R-CNN [14] and R-FCN [13] as the basic detectors. Both networks achieve the state-of-the-art performance when provided with strong supervisions. The Fast R-CNN network uses the RoI pooling layer and multi-task loss to improve the efficiency and effectiveness. The R-FCN optimizes the Fast R-CNN with the position-sensitive score maps, and all the computations are shared over the entire image instead of being split for each proposal. Each detector has a different architecture and thus reflects different, but complementary, intrinsic characteristics of the underlying samples. As for the region proposal, we use unsupervised methods, such as SS [23] and edge box [24]. We denote the proposal generation as function B, which takes an image I as input. For simplification, we denote the detector (Fast R-CNN and R-FCN) as function F. Therefore, the generation of region proposals can be formalized as:

\[
\text{rectangle} = (\text{up}, \text{left}, \text{bottom}, \text{right}),
\]

\[
B(I) = \{\text{rectangle}_i | 1 \leq i \leq n\},
\]

where each proposal is a rectangle in the image and \((\text{up}, \text{left})\) and \((\text{bottom}, \text{right})\) represent the coordinates of the upper left corner and the bottom right corner of this rectangle. The generated proposals are likely to be the true objects. We then have

\[
F(I, B(I)) = \{(\text{rectangle}, \text{score})_{(i,j)} | 1 \leq i \leq n, 1 \leq j \leq C\},
\]

where \(C\) is the number of object classes, \(\text{score}\) represents the confidence score for the corresponding proposal.

Some other algorithms [67], [68] can generate more robust and high-quality detection proposals. However, these algorithms usually require human annotations for training. This is not applicable to handling the situation of few annotations. Therefore, we leverage SS [23], an unsupervised method, to generate proposals in our experiments by default.

3.2 The MSPLD Model

Suppose we have \(l\) labeled images in which all the object bounding boxes are annotated. Note that, when we randomly annotate approximately four images for each class, an image may contain several objects, and we annotate all the object bounding boxes. We denote the labeled images as \(y_i \subseteq \mathbb{R}^4, C\), \(i = 1, \ldots, l\). We also have \(u\) unlabeled images \(y^u_i \subseteq \mathbb{R}^4, C\), \(i = 1, \ldots, u\). The unlabeled bounding boxes will be assigned labels, or discarded during each training iteration. We also assume there are \(m\) detection models. In technical terms, our method integrates multi-modal learning into the SPL framework. Our model can be formulated as Eq. (4), Eq. (5), Eq. (6) and Eq. (7).

\[
E(w^j, v^j_{i,c}, y^{u}_{i,j}, \lambda, \Psi) = \sum_{j=1}^{l} \sum_{i=1}^{m} L^i_s(y_{i,j}, B(I_i), w^j)
\]

\[
+ \sum_{j=1}^{l} \sum_{i=1}^{m} C v^j_{i,c} L^i_j(y^{u}_{i,j}, B(I_i), w^j)
\]

\[- \sum_{j=1}^{l} \sum_{i=1}^{m} \sum_{c=1}^{C} \lambda^j_{c v} w^j_{i,c} - \sum_{j=1}^{l} \sum_{i=1}^{m} \gamma^{j_1, j_2}(V^{j_1})^T V^{j_2}
\]

\[
\text{s.t.} \sum_{c=1}^{C} v^j_{i,c} \leq 1 \text{ for } 1 \leq j \leq m \text{ & } 1 \leq i \leq u,
\]

\[
v^j_{i,c} \in \{0, 1\} \text{ & } v \in \Psi_v,
\]

\[
y^{u}_{i,j} \subseteq F^*(I_i, B(I_i), w) \text{ and } y^{u}_{i,j} \in \Psi_y \text{ for } 1 \leq i \leq u,
\]

In Eq. (4), \(w^j\) denotes the parameters of the \(j^\text{th}\) basic detector. \(v^j_{i,c}\) encodes whether the bounding boxes in the \(i^\text{th}\) image are determined as the \(c^\text{th}\) class to train the \(j^\text{th}\) model. Thus, \(v^j_{i,c}\) can only be 0 or 1. \(y^{u}_{i,j}\) is the generated pseudo bounding boxes for the unlabeled images from the \(j^\text{th}\) detector. \(i, j, c\) are the indexes of images, models, and classes, respectively. \(V^j\) is a \(u \times C\) matrix and denotes all the \(v^j_{i,c}\) for the \(j^\text{th}\) detection model. \(\lambda\) is the parameter for the SPL regularization term, which enables the possibly selection of high confidence images during optimization. \(\gamma\) is the parameter for the multi-modal regularization term.

Note that an inner product regularization term \((V^j)^T V^{j_2}\) has been imposed on each pair of selection weights \(V^j\) and \(V^{j_2}\). This term delivers the basic assumption that different detection models share common knowledge of pseudo-annotation confidence for images, i.e., an unlabeled image is labeled correctly or incorrectly simultaneously for both models. This term thus encodes the relationship between multiple models. It uncovers the shared information and leverages the mutual benefits among all the models.

In Eq. (4), \(L_s\) represents the original multi-task loss of the supervised object detection [14], [17], [20]. The loss function for the unlabeled images \(L_u\) is defined as

\[
L_u = \begin{cases} L_s & \text{if the } \epsilon^\text{th} \text{ class appears in } y_i, \\ \infty & \text{if otherwise} \end{cases}
\]

Given the constraints in Eq. (5) and Eq. (6), it is guaranteed that \(L_s = \sum_{c=1}^{C} v^j_{i,c} L_c\) if the \(i^\text{th}\) image is selected as the
training data by the $j^{th}$ detection model. As the distribution of the confidence/loss can be different for different classes, this class-specific loss function helps the selected images cover as many classes as possible. $F^*$ indicates the fused results from multiple models, which contains $n \times C$ bounding boxes and, thus, has too many noisy objects. We use some empirical procedures to select the faithful pseudo-objects, and incorporate prior knowledges into a curriculum regime $y_i^u \in \Psi_y$. Similar to $\Psi_y$, some specially designed processes for discarding the unreliable images is denoted as $v \in \Psi_v$. The detailed steps of $\Psi_y$ and $\Psi_v$ will be discussed in the next section.

3.3 Optimization

Update $v^j$: This step aims to update the training pool of the $j^{th}$ detection model. We can calculate the derivative of Eq. (4) with respect to $v_{i,c}^j$ as:

$$\frac{\partial E}{\partial v_{i,c}^j} = L_c(y_{i,c}^u, I_i, B(I_i), w^j) - \lambda_c^j - \sum_{k=1; k \neq j}^m \gamma_{i,k} v_{i,c}^j. \tag{9}$$

Then the closed-form solution is

$$v_{i,c}^j = \begin{cases} 
1 & \text{if } L_{i,c}^j < \lambda_c^j + \sum_{k=1; k \neq j}^m \gamma_{i,k} v_{i,c}^k \\
0 & \text{if } L_{i,c}^j \geq \lambda_c^j + \sum_{k=1; k \neq j}^m \gamma_{i,k} v_{i,c}^k \end{cases} \tag{10}$$

for the unlabeled images. Due to the limitation of $\sum_{c=1}^C v_{i,c}^j \leq 1$, if there are multiple $v_{i,c}^j = 1$ for the same $(i,j)$ indicating the same image, we only choose the one with the lowest corresponding loss value $L_{i,c}^j$. The item $\gamma$ and $v_{i,c}^j$ uncover the shared information. Because if $v_{i,c}^k = 1$ (indicate the $i^{th}$ image is selected by the $k^{th}$ model) the threshold in Eq. (10) will become higher, and this image will become easier to be selected by the current detector.

Update $w^j$: We will train the basic detector of the $j^{th}$ model, given $v$ and $y^u$. The training data is the union set of initial annotated images and the selected images ($v_{i,c}^j = 1$) with the pseudo boxes $y^u$. Due to the limitation of $\sum_{c=1}^C v_{i,c}^j \leq 1$ and $v_{i,c}^j \in \{0, 1\}$, our selected images are unique. Finally, this step can be solved by the standard process, described as [13], [14].

Update $y^u$: Fixing $v$ and $w$, $y^u$ should be solved by the following minimization problem:

$$y_{i,j}^u = \arg \min_{y_{i,j}^u} \sum_{j=1}^m \sum_{c=1}^C v_{i,c}^j L_c(y_{i,j}^u, I_i, B(I_i), w^j), \tag{11}$$

s.t. $y_{i,j}^u \in F^*(I_i, B(I_i), w)$ for $1 \leq i \leq u$

It’s almost impossible to directly optimize $y_{i,j}^u$, because

---

**Algorithm 1 Alternative Optimization Algorithm for Solving MSPLD**

**Input:** $L = \{(x_i^l, y_i^l)\}$ and $U = \{(x_i^u)\}$

1. $m$ basic detectors with parameters $W$
2. $\lambda, \gamma, \Psi_y, \Psi_v$ and max iteration
3. initialize $W$ trained by $L$
4. initialize $V_j = O$ for $1 \leq j \leq m$
5. for $iter = 1$; $iter \leq \max; iter++$ do
6.   for $j = 1; j \leq m; j++$ do
7.     Clean up the unlabeled data via curriculum $\Psi_v$
8.     Generate the pseudo labels $y_i^u$ via Eq. (11)
9.     Compute loss $L_j^j$ by $j^{th}$ detector [13], [14]
10.    Update $V_j$ according to Eq. (10)
11.   Update $y_{i,j}^u$ and $V_j$ via the prior knowledge
12.   Retrain $w_j$ via training pool $\{(x_i^u, y_i^u)\} \cup \mathbb{L}$
13. end for
14. Update $\lambda, \gamma$ to select more images in the next round
15. end for

**Output:** detectors’ parameters $W = \{w_j|1 \leq j \leq m\}$
Eq. (7). The prior constrains to filter unreliable images, corresponding to Eq. (4). At the instance level, the current detector may either correct or directly use the previous results. For example, the green box of the plant is better aligned by the 2nd model compared to the 1st model; the blue box of the car detected by the 1st model is directly used by the 2nd model. At the image level, the previously selected images will be assigned higher priority in the next round, see Eq. (10). Besides, the probability of the unselected images remains unchanged.

The multi-modal mechanism pulls the self-paced baseline out of the local minimum by significantly improving the precision and recall of training objects and images. In Figure 3, we show the details of precision/recall using the ResNet-101 model and compare it to the method without multi-modal. We observe that, as the model iterates, the recall of the training data improves, while the precision decreases, which clearly demonstrates the trade-off between precision and recall. Meanwhile, the mean average precision (mAP) of object detection keeps increasing and remains stable when precision and recall reach convergence. Compared with the baseline (no multi-modal), the precision of images (denoted as “Img/P”) and instances (denoted as “Ins/P”) is improved by about 6% and 13% using multi-modal; the recall of generated objects and selected images is improved by more than 5%. These observations suggest that the multi-modal mechanism obtains a better trade-off between precision and recall.

An alternative optimization strategy (AOS) can be adopted to solve Eq. (4), and is summarized in Algorithm 1. The parameters are iteratively updated by the sequence \( y_{i1}, v^1, w^1, \ldots, v^j, w^j, y_{i2}, \ldots \) until there are no more available unlabeled data or the maximum iteration number is reached. In Algorithm 1, the 7th/11th steps are prior constraints to filter unreliable images, corresponding to Eq. (7). The 8th and 12th steps are the solution for updating \( y_i^1 \) and \( W \), respectively (see the second and third paragraphs in Sec.3.3). The 9th/10th steps are used to update \( V \) via the SPL and multi-modal regularization terms. Later, we illustrate this optimization process in Figure 1 and Figure 2.

Figure 1 illustrates a special case of our MSPLD with only one detection model, which means the case of \( m=1 \) in Eq. (4). We initialize the detector with few annotated bounding boxes. In the 1st round, we generate pseudo boxes with high confidences from some of the unlabeled images and retrain the detector by combining the strongly-labeled and the newly-labeled bounding boxes. In the next round, with the improved detector, we are able to generate more reliable pseudo boxes, such as the green boxes generated in round 2. Therefore, the process iterates between instance-level label generation and detector updates. Through these iterations, our approach gradually generates more bounding boxes with reliable labels, from “easy” to “hard”, shown in Figure 1, and we can, therefore, obtain a more robust detector with these newly labeled training data.

Since this method only uses very few training samples per category, a simple self-paced strategy may be trapped by local minimums. To avoid this problem, we incorporate multi-modal learning into the learning process, which corresponds to the case of \( m > 1 \) in Eq. (4). In Figure 2, we observe that the three detection models are complementary to each other. These different models can communicate with each other by the multi-modal regularization term. Each detector can communicate with each other by the effect of \( \gamma \) and the prior knowledge in Eq. (4). At the instance level, the current detector may either correct or directly use the previous results. For example, the green box of the plant is better aligned by the 2nd model compared to the 1st model; the blue box of the car detected by the 1st model is directly used by the 2nd model. At the image level, the previously selected images will be assigned higher priority in the next round, see Eq. (10). Besides, the probability of the unselected images remains unchanged.

The multi-modal mechanism pulls the self-paced baseline out of the local minimum by significantly improving the precision and recall of training objects and images. In Figure 3, we show the details of precision/recall using the ResNet-101 model and compare it to the method without multi-modal. We observe that, as the model iterates, the recall of the training data improves, while the precision decreases, which clearly demonstrates the trade-off between precision and recall. Meanwhile, the mean average precision (mAP) of object detection keeps increasing and remains stable when precision and recall reach convergence. Compared with the baseline (no multi-modal), the precision of images (denoted as “Img/P”) and instances (denoted as “Ins/P”) is improved by about 6% and 13% using multi-modal; the recall of generated objects and selected images is improved by more than 5%. These observations suggest that the multi-modal mechanism obtains a better trade-off between precision and recall.

There are two regularization parameters, \( \lambda \) and \( \gamma \), in our objective function Eq. (4). We show how \( \lambda \) changes during the training procedures in Figure 3. As \( \lambda \) is related to how many images are used during the training procedure. Therefore, we should use the appropriate parameter \( \lambda \) to guarantee the images in the training pool can stably increase over the training iterations. \( \gamma \) is usually fixed as 0.2/(m-1). More details can be found in experiments.

**Injecting prior knowledge.** In Eq. (6) and Eq. (7), prior knowledge \( \Psi_v \) and \( \Psi_y \) are leveraged to filter out some

1. “Image-level label” denotes which objects appear in an image.
2. “instance-level label” denotes (1) the type of the object instance and (2) the instance’s location (coordinates) in terms of a rectangular bounding box.
Fig. 3: The change of $\lambda$, precision, recall and mAP for the first four training iterations of MSPLD. “mv” and “no” denote using and not using multi-modal learning, respectively. “Img/R” and “Ins/R” indicate the image-level and instance-level recall, respectively. “Img/P” and “Ins/P” indicate the image-level and instance-level precision, respectively.

Fig. 4: Some poorly located or missed training samples. The yellow rectangles are the generated labeled boxes, and the discs denote the ground-truth objects. In image 2, the green and purple circles indicate people and sofa, respectively. We observe that the sofa is missed due to occlusions and different people are not well separated.

very challenging instances. For example, as suggested in Figure 4, an image could be very complex and it may be challenging to locate the correct bounding box. Therefore, we empirically design a method to estimate the number of boxes for each class in an image. Specifically, we apply a non-maximum suppression (NMS) on the output of $F^*$ for each class, and then use a confidence threshold of 0.2. Later, we employ NMS to filter out the nested boxes, which usually occurs when there are multiple overlapping objects. If there are too many boxes ($\geq 4$) for one specific class or too many classes ($\geq 4$) in the image, this image will be removed. To generate relatively robust pseudo instance-level labels (Eq. (7)), a class-specific threshold is applied on the remaining boxes to select the instance-level instances with high confidence. Additionally, images in which no reliable pseudo objects are found are filtered out.

**Discussion of model convergence.** Algorithm 1 adopts the AOS to solve MSPLD. It alternatively updates the parameters of the object detectors and the parameters of the regularization terms. When updating the parameters for the regularization terms, we can achieve the optimal solution via Eq. (10). When updating the parameters for the object detectors (CNN models), the model should converge to a local minimum by loss back propagation. This alternative updating procedure converges when all the unlabeled samples have been traversed and when the objective function in Eq. (4) cannot be further minimized. Therefore, the algorithm will finally converge.

**Model complexity.** Suppose the time complexity of training a detector is $O(\text{Flops})$, where Flops represents the floating-point operations of the network forward procedure. The overall time complexity of MSPLD then relies on the number of iterations in the alternative optimization algorithm and the number of detectors. Based on Algorithm 1, the time complexity of MSPLD is $O(\text{iter}_{\max} \times m \times \text{Flops})$, where the $m$ is the number of detectors and $\text{iter}_{\max}$ is the maximum iteration number. On PASCAL VOC’07, MSPLD can converge in no more than six iterations, and the standard setting of MSPLD may take about 50 hours using one GTX 1080 Ti GPU on PASCAL VOC. To learn new concept, we need to change the structure of the last classification layer and bounding box regression layer of the detectors. So we need to re-train the model based on the new data.

**4 Experimental Evaluation**

In this section, we compare MSPLD with some baselines on several large object detection benchmark datasets at
TABLE 2: Method comparisons in average precision (AP) on the PASCAL VOC 2007 test set. * indicates the usage of full image-level labels for training. Our approach (the last four rows) requires only approximately four strong annotated images per class. [69] leverages the SVM classifier to train the object detector via SPL. “SPL+Fast R-CNN” is our approach using only one model, i.e., Fast R-CNN with VGG16, and “SPL+R-FCN” denotes R-FCN with ResNet50[hem]. “SPL+Ensemble” ensembles the three models: Fast R-CNN with VGG16, R-FCN with ResNet50[hem] and R-FCN with ResNet101.

<table>
<thead>
<tr>
<th>Methods</th>
<th>bike</th>
<th>bird</th>
<th>boat</th>
<th>bus</th>
<th>car</th>
<th>cat</th>
<th>chair</th>
<th>cow</th>
<th>table</th>
<th>dog</th>
<th>hors</th>
<th>mbik</th>
<th>pers</th>
<th>plnt</th>
<th>shp</th>
<th>sofa</th>
<th>train</th>
<th>tv</th>
<th>mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zhang et al. [69]</td>
<td>47.4</td>
<td>23.3</td>
<td>25.3</td>
<td>23.0</td>
<td>54.7</td>
<td>48.0</td>
<td>41.8</td>
<td>1.8</td>
<td>28.9</td>
<td>27.8</td>
<td>37.7</td>
<td>41.6</td>
<td>43.8</td>
<td>20.0</td>
<td>12.0</td>
<td>27.8</td>
<td>22.9</td>
<td>48.9</td>
<td>31.6</td>
</tr>
<tr>
<td>Wang et al. [31]</td>
<td>48.9</td>
<td>42.3</td>
<td>26.1</td>
<td>11.3</td>
<td>41.9</td>
<td>40.9</td>
<td>34.7</td>
<td>10.8</td>
<td>34.7</td>
<td>18.8</td>
<td>34.4</td>
<td>35.4</td>
<td>52.7</td>
<td>19.1</td>
<td>17.4</td>
<td>35.9</td>
<td>33.3</td>
<td>48.4</td>
<td>45.5</td>
</tr>
<tr>
<td>Teh et al. [70]</td>
<td>48.8</td>
<td>45.9</td>
<td>37.4</td>
<td>26.9</td>
<td>9.2</td>
<td>50.7</td>
<td>43.4</td>
<td>43.6</td>
<td>10.6</td>
<td>35.9</td>
<td>27.0</td>
<td>38.6</td>
<td>48.5</td>
<td>43.8</td>
<td>24.7</td>
<td>12.1</td>
<td>29.0</td>
<td>23.2</td>
<td>48.8</td>
</tr>
<tr>
<td>Kantorov et al. [71]</td>
<td>57.1</td>
<td>52.0</td>
<td>31.5</td>
<td>7.6</td>
<td>11.5</td>
<td>55.0</td>
<td>53.1</td>
<td>1.7</td>
<td>33.1</td>
<td>49.2</td>
<td>42.0</td>
<td>47.3</td>
<td>56.6</td>
<td>15.3</td>
<td>12.8</td>
<td>35.9</td>
<td>33.3</td>
<td>47.9</td>
<td>40.7</td>
</tr>
<tr>
<td>Bilen et al. [8]</td>
<td>46.4</td>
<td>58.3</td>
<td>35.5</td>
<td>25.9</td>
<td>14.0</td>
<td>66.7</td>
<td>53.0</td>
<td>0.9</td>
<td>8.9</td>
<td>21.5</td>
<td>26.6</td>
<td>38.4</td>
<td>44.7</td>
<td>59.0</td>
<td>10.8</td>
<td>17.3</td>
<td>40.7</td>
<td>56.9</td>
<td>50.8</td>
</tr>
<tr>
<td>Li et al. [9]</td>
<td>54.5</td>
<td>47.4</td>
<td>41.3</td>
<td>20.8</td>
<td>17.7</td>
<td>51.9</td>
<td>63.5</td>
<td>46.1</td>
<td>21.8</td>
<td>57.1</td>
<td>22.1</td>
<td>34.4</td>
<td>50.5</td>
<td>61.8</td>
<td>16.2</td>
<td>29.9</td>
<td>40.7</td>
<td>55.3</td>
<td>40.2</td>
</tr>
<tr>
<td>Diba et al. [32]</td>
<td>49.5</td>
<td>60.8</td>
<td>38.6</td>
<td>29.2</td>
<td>17.3</td>
<td>61.0</td>
<td>79.2</td>
<td>56.9</td>
<td>10.9</td>
<td>44.4</td>
<td>29.1</td>
<td>42.7</td>
<td>64.1</td>
<td>61.2</td>
<td>12.2</td>
<td>24.9</td>
<td>45.9</td>
<td>54.1</td>
<td>50.6</td>
</tr>
<tr>
<td>Dong et al. [63]</td>
<td>62.5</td>
<td>54.6</td>
<td>44.3</td>
<td>12.9</td>
<td>12.7</td>
<td>63.8</td>
<td>60.6</td>
<td>25.0</td>
<td>5.4</td>
<td>48.0</td>
<td>49.3</td>
<td>58.7</td>
<td>66.6</td>
<td>63.5</td>
<td>8.5</td>
<td>17.3</td>
<td>40.7</td>
<td>59.4</td>
<td>53.9</td>
</tr>
<tr>
<td>Tang et al. [33]</td>
<td>65.5</td>
<td>67.2</td>
<td>47.2</td>
<td>21.6</td>
<td>22.1</td>
<td>68.0</td>
<td>68.5</td>
<td>35.9</td>
<td>5.7</td>
<td>63.1</td>
<td>49.5</td>
<td>30.3</td>
<td>64.7</td>
<td>66.1</td>
<td>1.3</td>
<td>20.5</td>
<td>25.0</td>
<td>50.7</td>
<td>61.2</td>
</tr>
<tr>
<td>Ge et al. [34]</td>
<td>64.3</td>
<td>68.0</td>
<td>56.2</td>
<td>36.4</td>
<td>23.1</td>
<td>68.5</td>
<td>67.2</td>
<td>64.9</td>
<td>7.1</td>
<td>54.1</td>
<td>47.0</td>
<td>57.0</td>
<td>69.3</td>
<td>65.4</td>
<td>20.8</td>
<td>23.2</td>
<td>50.7</td>
<td>56.9</td>
<td>65.2</td>
</tr>
<tr>
<td><strong>SPL+Fast R-CNN</strong></td>
<td>41.4</td>
<td>55.9</td>
<td>24.5</td>
<td>15.7</td>
<td>22.4</td>
<td>37.3</td>
<td>52.4</td>
<td>37.9</td>
<td>14.3</td>
<td>17.5</td>
<td>33.0</td>
<td>27.9</td>
<td>41.4</td>
<td>50.2</td>
<td>36.7</td>
<td>19.5</td>
<td>27.2</td>
<td>46.0</td>
<td>47.5</td>
</tr>
<tr>
<td><strong>SPL+R-FCN</strong></td>
<td>25.6</td>
<td>34.3</td>
<td>26.0</td>
<td>15.3</td>
<td>23.3</td>
<td>39.3</td>
<td>48.3</td>
<td>30.4</td>
<td>18.8</td>
<td>17.3</td>
<td>2.2</td>
<td>18.6</td>
<td>40.9</td>
<td>54.8</td>
<td>35.4</td>
<td>13.5</td>
<td>26.2</td>
<td>46.4</td>
<td>65.1</td>
</tr>
<tr>
<td><strong>SPL+Ensemble</strong></td>
<td>38.4</td>
<td>51.1</td>
<td>41.4</td>
<td>21.6</td>
<td>25.9</td>
<td>45.0</td>
<td>57.6</td>
<td>50.0</td>
<td>22.0</td>
<td>21.7</td>
<td>7.5</td>
<td>23.8</td>
<td>47.4</td>
<td>56.0</td>
<td>40.4</td>
<td>23.2</td>
<td>41.6</td>
<td>57.8</td>
<td>42.0</td>
</tr>
<tr>
<td><strong>MSPLD</strong></td>
<td>1.0</td>
<td>0.0</td>
<td>3.4</td>
<td>6.7</td>
<td>12.1</td>
<td>31.3</td>
<td>5.9</td>
<td>12.1</td>
<td>40.1</td>
<td>17.5</td>
<td>31.6</td>
<td>52.0</td>
<td>65.2</td>
<td>61.4</td>
<td>52.1</td>
<td>41.7</td>
<td>50.8</td>
<td>57.8</td>
<td>51.2</td>
</tr>
</tbody>
</table>

TABLE 4: Performance comparison on PASCAL VOC 2007 of different proposal generation methods.

<table>
<thead>
<tr>
<th>Selective Search</th>
<th>EdgeBox</th>
<th>Selective Search + EdgeBox</th>
<th>mAP</th>
</tr>
</thead>
<tbody>
<tr>
<td>CorLoc</td>
<td>65.3</td>
<td>65.2</td>
<td>65.6</td>
</tr>
</tbody>
</table>

4.1 Datasets

We evaluate our method on PASCAL VOC 2007 [52], PASCAL VOC 2012 [74], MS COCO 2014 [75], and ILSVRC 2013 detection dataset [76]. These four datasets are the most widely used benchmarks in the object detection task. PASCAL VOC 2007 contains 10022 images annotated with bounding boxes for 20 object categories. It is officially split into 2501 training, 2510 validation, and 5011 testing images. PASCAL VOC 2012 is similar to PASCAL VOC 2007, but contains more images: 5171 training, 5823 validation images and 10991 testing images. MS COCO 2014 contains 80k images for training and 40k images for validation, which are categorized into 80 classes. ILSVRC 2013 is a much larger dataset with 200 categories for the detection task, which contains more than 400k images. The standard training, validation and test splits for training and evaluation are used for these three datasets.

4.2 Implementation Details

We build R-FCN and Fast R-CNN on various base models as different detection models. Three base models are tested in our experiments, i.e., GoogleNet [77], VGG [78], and ResNet [79]. These models are pre-trained on ILSVRC 2012 [80]. A boosting method, i.e., online hard example mining (OHEM) [81], is also tested in our experiments.

3. We suggest the following two-fold standards to select models in our method. First, each selected single model should exhibit possibly good performance in object detection. Second, the selected models should be possibly different from each other in aspects such as model structure and training strategy. In this manner, these models will be largely complementary to each other to guide a good performance of the final performance.
### 4.3 Comparison with State-of-the-art Algorithms

We compare MSPLD with recent state-of-the-art WSOD algorithms [8], [9], [31], [32], [69], [70], [71]. Fair comparisons are claimed because many of these methods use multiple models as well. Bilen et al. [8] use ensembles to improve performance. Li et al. [9] use multiple steps. They first train a classification model and apply a MIL model to mine the candidate objects, and then fine-tune a detection model to detect the objects. Diba et al. [32] cascade three networks: a location network, a segmentation network and a MIL network, and apply multi-scale data argumentation. ‘SPL+Ensemble’ in Table 2/3 represents the late fusion of multiple models. This method simply averages the confidence scores and the refined bounding boxes (Eq. (3)), then follows the standard NMS and thresholding procedures. In our comparison, we present the best results from their articles. To evaluate the sensitivity of our method w.r.t different initializations, we use random seeds to generate different initial fully annotated images. For each experiment, we repeat four times, and mean performance and the standard deviation are reported. Even if we only use few strong annotations for each class, our fused detection model can reduce the sensitivity to the initial annotated images.

Table 2 summarizes the AP on the PASCAL VOC 2007 test set. The competing methods usually use full image-level labels. In contrast, we use the same set of images but with much fewer annotations: totally 60 annotated images and the others are free-labeled. Although the annotated images account for less than 1% of the total number of training images, MSPLD achieves 41.7% mAP, a competitive performance compared to state-of-the-art WSOD algorithms. Our results achieve the best performance on some specific classes, e.g., the AP of person, bottle and cat exceeds the second best by 16%, 10%, and 12%, respectively. We view [69] as a comparable baseline to our method, which leverages the same base model VGG16 as our “SPL+Fast R-CNN” baseline. In comparison, our baseline method, SPL+Fast R-CNN, uses fewer annotations, but outperforms [69] by 2.4% and 10.3% in mAP and CorLoc, respectively. The SPL+Fast R-CNN model is superior to SPL+R-FCN, because Fast R-CNN of 50% for CorLoc and leverage the official evaluation code provided by [52] to calculate AP.

**Initially labeled images.** For each class, we randomly label $k$ images, which contain the box for this class. We use $k = 3$ if not specified, which results in a total of 60 initial annotated images. All the object bounding boxes in these 60 images are annotated, so in effect there are an average of 4.2 images per class, since some images have multiple classes.
TABLE 7: Ablation studies. “#Models” represents the number detection models used. “R.” indicates the R-FCN detector, and “F.” indicates the Faster RCNN detector. “R50”, “VGG16”, “Gog”, and “R101” indicate the base models, ResNet-50, VGG-16, GoogleNet-v1, and ResNet-101, respectively. “ohem” indicates whether the OHEM module is embedded. “no prior” represents that the filtration strategy is not used. “no SPL” means that we directly train the model with all the data after filtration, rather than using SPL.

<table>
<thead>
<tr>
<th>#Models</th>
<th>Detection Model</th>
<th>mAP</th>
<th>CorLoc</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>R-R50 no prior</td>
<td>28.6</td>
<td>50.1</td>
</tr>
<tr>
<td></td>
<td>R-R50 no SPL</td>
<td>27.2</td>
<td>44.4</td>
</tr>
<tr>
<td></td>
<td>R-R50ohem</td>
<td>28.9</td>
<td>50.6</td>
</tr>
<tr>
<td></td>
<td>R-Gogohem</td>
<td>29.9</td>
<td>53.2</td>
</tr>
<tr>
<td></td>
<td>F-VGG16 no prior</td>
<td>32.8</td>
<td>60.1</td>
</tr>
<tr>
<td></td>
<td>F-VGG16</td>
<td>33.7</td>
<td>60.9</td>
</tr>
<tr>
<td>2</td>
<td>R-R50ohem + F-VGG16</td>
<td>38.3</td>
<td>63.4</td>
</tr>
<tr>
<td></td>
<td>R-R50ohem + R-Gogohem</td>
<td>32.1</td>
<td>57.3</td>
</tr>
<tr>
<td></td>
<td>R-Gogohem + F-VGG16</td>
<td>35.8</td>
<td>61.6</td>
</tr>
<tr>
<td>3</td>
<td>R-R50ohem + F-VGG16 + R-Gogohem</td>
<td>38.5</td>
<td>62.8</td>
</tr>
<tr>
<td></td>
<td>R-R50ohem + F-VGG16 + R-R101</td>
<td>41.7</td>
<td>65.5</td>
</tr>
<tr>
<td></td>
<td>R-R50ohem + F-VGG16 + R-R101ohem</td>
<td>38.9</td>
<td>63.4</td>
</tr>
<tr>
<td></td>
<td>R-R50ohem + F-VGG16 + R-R101ohem</td>
<td>37.5</td>
<td>61.4</td>
</tr>
<tr>
<td></td>
<td>R-R50ohem + F-VGG16ohem + R-R101</td>
<td>37.1</td>
<td>61.1</td>
</tr>
</tbody>
</table>

TABLE 8: Performance comparison of MSPLD on PASCAL VOC 2007 using different numbers of noisy images for the MSPLD model with $k = 3$ for initialization.

<table>
<thead>
<tr>
<th>#noisy images</th>
<th>0</th>
<th>1000</th>
<th>5000</th>
<th>10000</th>
</tr>
</thead>
<tbody>
<tr>
<td>noise scale</td>
<td>0%</td>
<td>20%</td>
<td>40%</td>
<td>100%</td>
</tr>
<tr>
<td>mAP</td>
<td>41.7</td>
<td>39.9</td>
<td>39.8</td>
<td>39.8</td>
</tr>
<tr>
<td>CorLoc</td>
<td>65.5</td>
<td>64.0</td>
<td>63.9</td>
<td>63.5</td>
</tr>
</tbody>
</table>

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2844853, IEEE Transactions on Pattern Analysis and Machine Intelligence.
The robustness regarding the noisy images. All previous experiments are based on well-annotated datasets. For example, we know that the images in PASCAL VOC 2007 contain at least one object of the 20 classes. Therefore, we have added images from YFCC100M [85] as noisy images to the PASCAL VOC 2007 dataset. This experiment can make our algorithm completely unsupervised and demonstrate its robustness against outliers. Specifically, we first randomly sampled 10,000 images from YFCC100M and used various numbers of images from these 10,000 images as noisy images. We then employed this augmented dataset for detector learning. Results are shown in Table 8. It can be observed that our approach still yields a competitive detection accuracy when more than half of the augmented dataset are noisy images. These results demonstrate the robustness of our method against outliers.

Analysis of the generalization ability. Since all the classes of the detection datasets are contained in the 1000 classes of the ImageNet dataset, the pre-trained models use some pre-knowledge of their detection classes. Such knowledge may benefit the quality of the detectors obtained by MSPLD. To demonstrate the generalization ability of MSPLD, we use pre-trained models that are not trained on the detection classes. To this end, we construct Non-overlapping ImageNet-VOC/COCO sets for pre-training. For PASCAL VOC 2007, we manually select 746 ImageNet classes, which do not overlap with the 20 detection classes of PASCAL VOC. Images from these selected 746 classes compose of the None-overlap ImageNet-VOC subset. For MS COCO 2014, we manually select 706 ImageNet classes, which do not overlap with the 80 detection classes of MS COCO. Image samples from the selected 706 classes form the None-overlap ImageNet-COCO subset. We use such constructed Non-overlapping ImageNet-VOC and Non-overlapping ImageNet-COCO sets to pre-train the VGG16, ResNet-50, and ResNet-101 models for experiments on PASCAL VOC and MS COCO 2014, respectively. We observe that mAP on PASCAL VOC 2007 drops from 41.7% to 38.2%; the localization prediction mAP on MS COCO 2014 drops from 56.6% to 53.3%. There might be two reasons that cause such performance drop. The first should be the lack of detection classes during pre-training, while another important reason should be the less number of pre-trained data. To evaluate which one causes the performance drop, we have randomly sampled 74.6% training images from ImageNet to form the Overlapping ImageNet-VOC set, which contains the same number of training data with the Non-overlapping ImageNet-VOC set, but is not enforced not to contain PASCAL VOC classes. We then use Overlapping ImageNet-VOC to pre-train the VGG16, ResNet-50, and ResNet-101 models for experiments on PASCAL VOC. We observe that mAP on PASCAL VOC 2007 drops from 41.7% to 38.9%. The performance of Overlapping ImageNet-VOC pre-training is almost similar to the performance with Non-Overlapping ImageNet-VOC. This verifies that pre-training without the detection classes does not substantially affect the performance of MSPLD.

4.5 Qualitative Analysis

Qualitative results over the training iterations. We show pseudo-labeled images by MSPLD over the training iterations.
iterations in Figure 6. Briefly, in the first iteration, the detector tends to choose images with relatively high classification confidence aggregated over the bounding boxes. After the detector is updated, it can gradually label objects in more complicated situations, e.g., the rotated TV monitor and several small bottles in Figure 6.

**Error analysis.** Some of the images that are newly generated by our method are shown in Figure 7. We observe that the generated pseudo boxes have good localization accuracies, but cannot detect every object in complex images. For example, the pseudo boxes correctly localize the true objects in the first five images. However, all these images contain multiple objects, and have occlusions, or overlaps between the objects. The generated boxes do not cover all objects well, which will compromise the performance of the final detectors. Prior knowledge could filter out some of the complex images, but this problem remains to be solved. We will focus on generating robust pseudo boxes for complex images in the future.

### 5 Conclusion and Future Work

In this paper, we propose an object detection framework (MSPLD) that uses only a few bounding box labels per category by consistently implementing iterations between detector amelioration and reliable sample selection. To enhance its detector learning capability with the scarcity of annotation, MSPLD embeds multiple detection models in its learning scheme. It can fully use the discriminative knowledge for different detection models, and possibly complement them to ameliorate the detector training quality. Under such extremely limited supervision information, MSPLD can achieve competitive performance compared to state-of-the-art WSOD approaches, which use more supervised knowledge of samples than our method.

MSPLD still requires about 1% of the images in the entire dataset to be annotated. In future, we will focus on further reducing the annotation information, i.e., only using one image per class, to obtain the similar performance. Except for the improvement of the base CNN feature and the object detector, the challenges are how to initialize the detector from limited annotation and, design a robust learning scheme to ameliorate the detector stably. Besides, we will investigate to improve our method into accommodating novel classes while simultaneously not destroy the accuracy of the training models on the previously trained ones.

**References**


This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TPAMI.2018.2844853, IEEE Transactions on Pattern Analysis and Machine Intelligence

SUBMITTED TO IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 2018.


Xuanyi Dong received the B.E. degrees in Computer Science and Technology from Beihang University, Beijing, China, in 2016. He is currently a first year Ph.D. student in the Center of Artificial Intelligence, University of Technology Sydney, Australia, under the supervision of Prof. Yi Yang.

Liang Zheng received the Ph.D degree in Electronic Engineering from Tsinghua University, China, in 2015, and the B.E. degree in Life Science from Tsinghua University, China, in 2010. He was a postdoc researcher in University of Texas at San Antonio, USA. He is now a postdoc researcher in the Center of Artificial Intelligence, University of Technology Sydney, Australia. His research interests are image retrieval, person re-identification and deep learning.

Fan Ma received the B.E. degree from Xian Jiaotong University, Xian, China, in 2014. He is currently a graduate student at Xian Jiaotong University. His research interests include machine learning and computer vision, especially on semi-supervised learning, self-paced learning and person re-identification.

Yi Yang received the Ph.D. degree in computer science from Zhejiang University, Hangzhou, China, in 2010. He was a Post-Doctoral Research with the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA. He is currently a Professor with University of Technology Sydney, Australia. His current research interest includes machine learning and its applications to multimedia content analysis and computer vision.

Deyu Meng received the B.Sc., M.Sc., and Ph.D. degrees from Xian Jiaotong University, Xian, China, in 2001, 2004, and 2008, respectively. He is currently an Associate Professor with the Institute for Information and System Sciences, School of Mathematics and Statistics, Xian Jiaotong University. From 2012 to 2014, he took his two-year sabbatical leave in Carnegie Mellon University. His current research interests include self-paced learning, noise modeling, and tensor sparsity.