

# Few-Example Object Detection with Model Communication

Xuanyi Dong<sup>1</sup>, Liang Zheng<sup>1</sup>, Fan Ma<sup>1</sup>, Yi Yang<sup>1</sup>, and Deyu Meng<sup>1</sup>

**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 [15], [16], [17], [18], [19], [20]. 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 percent 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], [21]. 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 [15], [16], [17], [18], [19], 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 methods in this category 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 four-fold. 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 percent mAP on the class of person, but we achieve

- X. Dong, L. Zheng, and F. Ma are with the Centre for Artificial Intelligence, University of Technology Sydney, Ultimo, NSW 2007, Australia. E-mail: {xuanyi.dong, fan.ma}@student.uts.edu.au, liangzheng06@gmail.com.
- Y. Yang is with the Centre for Artificial Intelligence, University of Technology Sydney, Ultimo, NSW 2007, Australia, and also with the State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100190, China. E-mail: yi.yang@uts.edu.au.
- D. Meng is with the School of Mathematics and Statistics, Ministry of Education Key Lab of Intelligent Networks and Network Security, Xi’an Jiaotong University, Xian Shi, Shaanxi 710048, P.R. China. E-mail: dymeng@mail.xjtu.edu.cn.

Manuscript received 17 Aug. 2017; revised 25 May 2018; accepted 1 June 2018. Date of publication 6 June 2018; date of current version 12 June 2019.

(Corresponding author: Deyu Meng.)

Recommended for acceptance by S. Lazebnik.

For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.

Digital Object Identifier no. 10.1109/TPAMI.2018.2844853

TABLE 1

Comparison of Different Supervision Information Used in Weakly (Semi-) Supervised and Few-Example Object Detection Algorithms

Methods	Data with Strong Supervision	Data with Weak Supervision	Data without Supervision	Test Dataset
[1]	[I] Flickr; PASCAL VOC [I] ILSVRC2013-DET	[V] YouTube	-	PASCAL VOC
[2]	- [V] YouTube-Object	[I] PASCAL VOC	-	PASCAL VOC
[3]	[I] Flickr [V] Part of VIRAT and KITTI	- [V] Part of KITTI	[V] Part of VIRAT KITTI	VIRAT
[4], [5], [6]	[I] ILSVRC2014 [V] Part of YouTube-Object	- [V] Part of YouTube-Object	[I] PASCAL VOC 2007 YouTube-Object	PASCAL VOC
[7], [8], [9], [10]	-	[I] PASCAL VOC	-	PASCAL VOC
[8], [11], [12], [13]	-	[I] ILSVRC2013	-	ILSVRC2013
[14]	[I] 10-200 images per class on SUN; PASCAL VOC	[I] SUN	[I] SUN	SUN
<b>Ours</b>	[I] 3-4 images per class on PASCAL VOC	-	[I] PASCAL VOC	PASCAL VOC

[I] and [V] denote the image and video dataset, respectively. Strong supervision provides the fully annotated images or videos; weak supervision only provides image-level or video-level labels. Data without supervision does not provide any annotation. Our method consumes negligible annotation efforts compared to others.

40.1 percent 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, during the training iterations, the selected training images go from “easy” (with relatively high confidence) to “hard”, and the object detector is gradually 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 incorporated 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 improves the precision and recall of generated training sample 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:

- 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.
- 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.
- 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 [16], [17], [18], [19], [22]. The roadmap of proposal-based methods starts from R-CNN [22] and is improved by SPP-Net [23] and Fast R-CNN [16] in terms of accuracy and speed. Later, Faster R-CNN [19] uses the region proposal network to quickly generate object regions, achieving a high recall compared to previous methods [24], [25]. Many methods directly predict bounding boxes without generating region proposals [17], [18]. For example, YOLO [18] uses the whole feature map from the last convolution layer. SSD [17] 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 [12], [27], [28]. Others design new CNN architectures to obtain object information from the classification loss and leverage this classification model to derive object detectors [7], [8], [9], [13]. For example, Bilen et al. [7] propose a weakly supervised detection network using selective search (SS) [24] to generate proposals and train image-level classification based on regional features. Li et al. [8] train an image-level classifier to adapt detection results through a mask-out strategy and MIL. Tang et al. [10] 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. [29] 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. [14] 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], [14], [30], 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 [31], [32], [33], [34] aims to learn a model based on a few training examples without unlabeled data. In contrast, learning from few samples [14], [30] 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 percent 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. [35] 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. [36] 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 [37] 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 [38], [39] propose an unsupervised algorithm to discover the common objects from large image collections via the Internet search. They usually assume the clean labels, but for most object classes, this assumption is unrealistic in real-world settings.

### 2.6 Zero-Shot Object Detection

Zero-shot object detection (ZSD) [40], [41] aims to locate object instances belonging to novel categories without any training examples. Rahman et al. [40] propose a deep network to model the interplay between visual and semantic domain information jointly. Bansal et al. [41] adapt visual-semantic embeddings for ZSD, and provide novel splits and baseline experiments on MSCOCO and Visual Genome [42]. ZSD 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 ZSD, 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. [43] ensemble multiple part detectors to form sub-structure detectors, which further constitute the final object detector. The algorithm of [44] is based on the linear SVM classifier, which is limited to using the off-the-shelf features. Yang et al. [45] use a low rank model to ensemble knowledge learned from different tasks. Zheng et al. [46] fuse the verification and classification models. Bilen et al. [7] averagely fuse three detection models with different architectures. Ma et al. [47] suggest assigning different weights of negative examples could improve the detection performance. Many previous detection methods [7], [43], [44] 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 [48] and self-paced learning [49]. Bengio et al. [48] first propose a learning paradigm in which organizing the examples in a meaningful order significantly improves the performance. Kumar et al. [49] propose to determine the training sample order by how easy they are. Many other researchers [50], [51], [52], [53], [54], [55] propose more theoretically analysis on this progressive paradigm. Some researches also apply the similar idea of the progressive paradigm [56], [57], [58]. For example, Wei et al. [59] propose a simple to complex framework that learns to segment with image-level labels. Liang et al. [60] leverage a iterative framework to learn segmentation from YouTube videos. Our algorithm extends this progressive strategy into multiple model ensemble. Consequently, we obtain a significantly improvement in object detection from few examples.

## 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 Section 3.1, and demonstrate the detailed formulation of our MSPLD in Section 3.2. Then, we describe the optimization method in Section 3.3. Lastly, we show the whole algorithm description in Section 3.4.

### 3.1 Preliminaries

We choose Fast R-CNN [16] and R-FCN [15] 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 Region-of-Interest (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 because they do not require human annotations and are applicable to handling the situation of few annotations in our setting, such as SS [24] and edge box [25]. 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

$$\text{rectangle} = (\text{up}, \text{left}, \text{bottom}, \text{right}), \quad (1)$$

$$B(I) = \{\text{rectangle}_i | 1 \leq i \leq n\}, \quad (2)$$

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

$$F(I, B(I)) = \{(\text{rectangle}, \text{score})_{(i,j)} | 1 \leq i \leq n, 1 \leq j \leq C\}, \quad (3)$$

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

### 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  $y_i \in [\mathbb{R}^4, C], i = 1, \dots, l$ . We also have  $u$  unlabeled images  $y_i^u \in [\mathbb{R}^4, C], i = 1, \dots, u$ . 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 Eqs. (4), (5), (6) and (7)

$$E(w^j, v_{i,c}^j, y_i^{u,j}, \lambda, \Psi) = \sum_{j=1}^m \sum_{i=1}^l L_s^j(y_i, I_i, B(I_i), w^j) + \sum_{j=1}^m \sum_{i=1}^u \sum_{c=1}^C v_{i,c}^j L_c^j(y_i^{u,j}, I_i, B(I_i), w^j) \quad (4)$$

$$- \sum_{j=1}^m \sum_{i=1}^u \sum_{c=1}^C \lambda_c^j v_{i,c}^j - \sum_{j_1=1}^m \sum_{j_2=j_1+1}^m \gamma^{j_1, j_2} (V^{j_1})^T V^{j_2}$$

$$\text{s.t.} \quad \sum_{c=1}^C v_{i,c}^j \leq 1 \quad \text{for} \quad 1 \leq j \leq m \quad \& \quad 1 \leq i \leq u, \quad (5)$$

$$v_{i,c}^j \in \{0, 1\} \quad \& \quad v \in \Psi_v, \quad (6)$$

$$y_i^{u,j} \in F^*(I_i, B(I_i), w) \quad \text{and} \quad y_i^{u,j} \in \Psi_y \quad \text{for} \quad 1 \leq i \leq u, \quad (7)$$

where  $w^j$  denotes the parameters of the  $j$ th basic detector.  $v_{i,c}^j$  encodes whether the bounding boxes in the  $i$ th image are determined as the  $c$ th class to train the  $j$ th model. Thus,  $v_{i,c}^j$  can only be 0 or 1.  $y_i^{u,j}$  is the generated pseudo bounding boxes for the unlabeled images from the  $j$ 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  $v_{i,c}^j$  for the  $j$ 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  $(V^i)^T V^j$  has been imposed on each pair of selection weights  $V^i$  and  $V^j$ . 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 [16], [19], [22]. The loss function

for the unlabeled images  $L_c$  is defined as

$$L_c = \begin{cases} L_s & \text{if the } c^{\text{th}} \text{ class appears in } y_i \\ \infty & \text{if otherwise.} \end{cases} \quad (8)$$

Given the constraints in Eqs. (5) and (6), it is guaranteed that  $L_s = \sum_{c=1}^C v_{i,c}^j L_c$  if the  $i$ 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  $y^u \in \Psi_y$ . 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

We adopt the alternative optimization strategy (AOS) to solve Eq. (4). The parameters are iteratively updated by the sequence  $y^{u1}, v^1, w^1, \dots, v^j, y^{uj}, w^j, y^{u1}, v^1, w^1, \dots$  until there are no more available unlabeled data or the maximum iteration number is reached. In this section, we show how to solve each parameter as follows.

*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  $v_c^j$  as

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

Then the closed-form solution is

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

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  $v_{i,c}^k$  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 [15], [16].

*Update  $y^{uj}$ .* Fixing  $v$  and  $w$ ,  $y^{uj}$  should be solved by the following minimization problem:

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

*s.t.*  $y_i^{uj} \subset F^*(I_i, B(I_i), w)$  for  $1 \leq i \leq u$ .

It's almost impossible to directly optimize  $y_i^{uj}$ , because  $y_i^{uj} \subset [\mathbb{R}^4, C]$  is a set of bounding boxes. Hence, we leverage prior knowledges to empirically calculate pseudo boxes  $y_i^{uj}$ . We fuse the results from all detection models and obtain the outputs of  $F^*$ . Then the post-processes of NMS and thresholding are applied on  $F^*$  to generate  $y_i^{uj}$ .

---

### Algorithm 1. AOS for Solving MSPLD

---

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

- 1:  $m$  basic detectors with parameters  $W$
- 2:  $\lambda, \gamma, \Psi_v, \Psi_y$  and max iteration
- 3: initialize  $W$  trained by  $\mathbb{L}$
- 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  $L_c^j$  by  $j$ th detector [15], [16]
- 10:     Update  $V_j$  according to Eq. (10)
- 11:     Update  $y^{uj}$  and  $V_j$  via the prior knowledge
- 12:     Retrain  $w_j$  via training pool  $\{(x_i^u, y_i^{uj})\} \cup \mathbb{L}$
- 13:   **end for**
- 14:   Update  $\lambda, \gamma$  to select more images in the next round
- 15: **end for**

**Output:** detectors' parameters  $W = \{w_j | 1 \leq j \leq m\}$

---

### 3.4 Algorithm Description

We summarize MSPLD in Algorithm 1. The 7th/11th steps are prior constrains to filter unreliable images, corresponding to Eq. (7). The 8th and 12th steps are the solution for updating  $y_i^u$  and  $W$ , respectively (see the second and third paragraphs in Section 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 Figs. 1 and 2.

Fig. 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 Fig. 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 Fig. 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

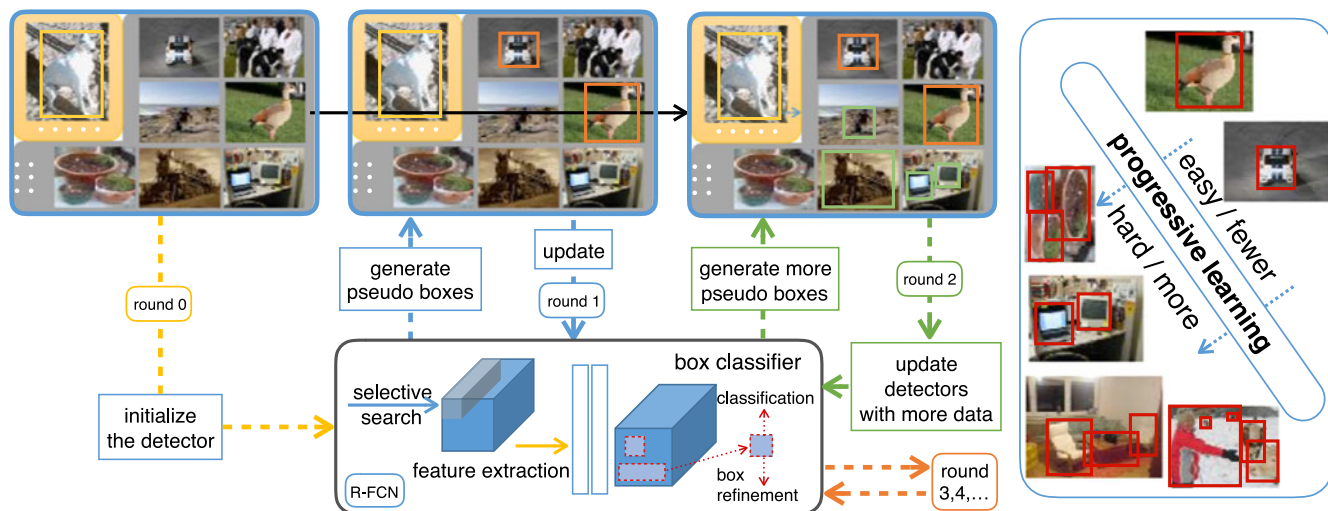


Fig. 1. A simplified version of MSPLD without multi-modal learning. The blue boxes in the top row contain the training images where the few labeled and the many unlabeled images are in the gray and yellow areas, respectively. The gray solid box represents our detector, R-FCN. We train the detector using the few annotated images. The detector generates reliable pseudo instance-level labels and then gets improved with these pseudo-labeled bounding boxes, as shown round 1. In the following rounds (iterations), the improved detector can generate larger numbers of reliable pseudo-labels that further update the detector. When the label generation and detector updating steps work iteratively, more pseudo boxes are obtained from “easy” to “hard”, and the detector becomes more robust.

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 Fig. 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<sup>1</sup> (denoted as “Img/P”) and instances<sup>2</sup> (denoted as “Ins/P”) is improved by about 6 and 13 percent using multi-modal; the recall of generated objects and selected images is improved by more than 5 percent. 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 Fig. 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.

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.

*Injecting Prior Knowledge.* In Eqs. (6) and (7), prior knowledge  $\Psi_v$  and  $\Psi_y$  are leveraged to filter out some very challenging instances. For example, as suggested in Fig. 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

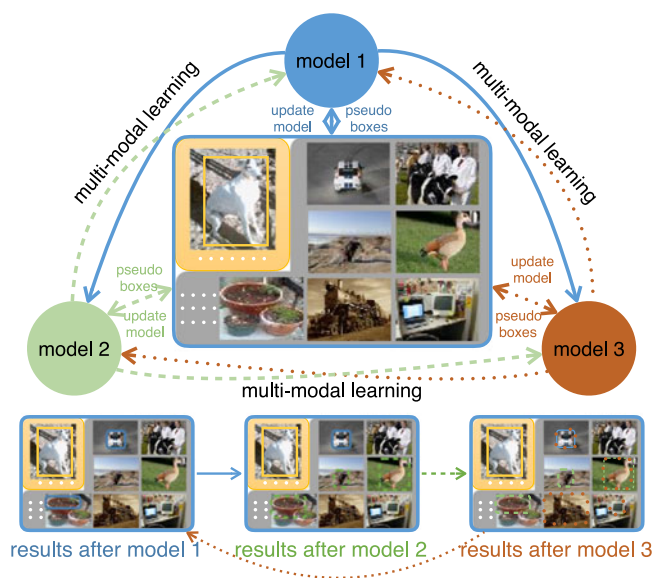


Fig. 2. The working flow of MSPLD when multi-modal learning is integrated with Fig. 1. An example with three models is shown. The three discs with different colors indicate the basic detectors. The images in the middle are the training data. The three detectors complement each other in validating the selected training samples. For example, as shown in the bottom row, the 1st model only detects two objects and misalignment exists with the detected plant. The 2nd model detects three other objects. When considering the detections of the 1st model, the misaligned plant is corrected, and the car with the blue box is also used to train the 2nd model. So more training data with reliable labels are used to improve the performance of model 2. Similarly, the 3rd model obtains more pseudo boxes and gets updated in turn. The whole procedure iterates until convergence.



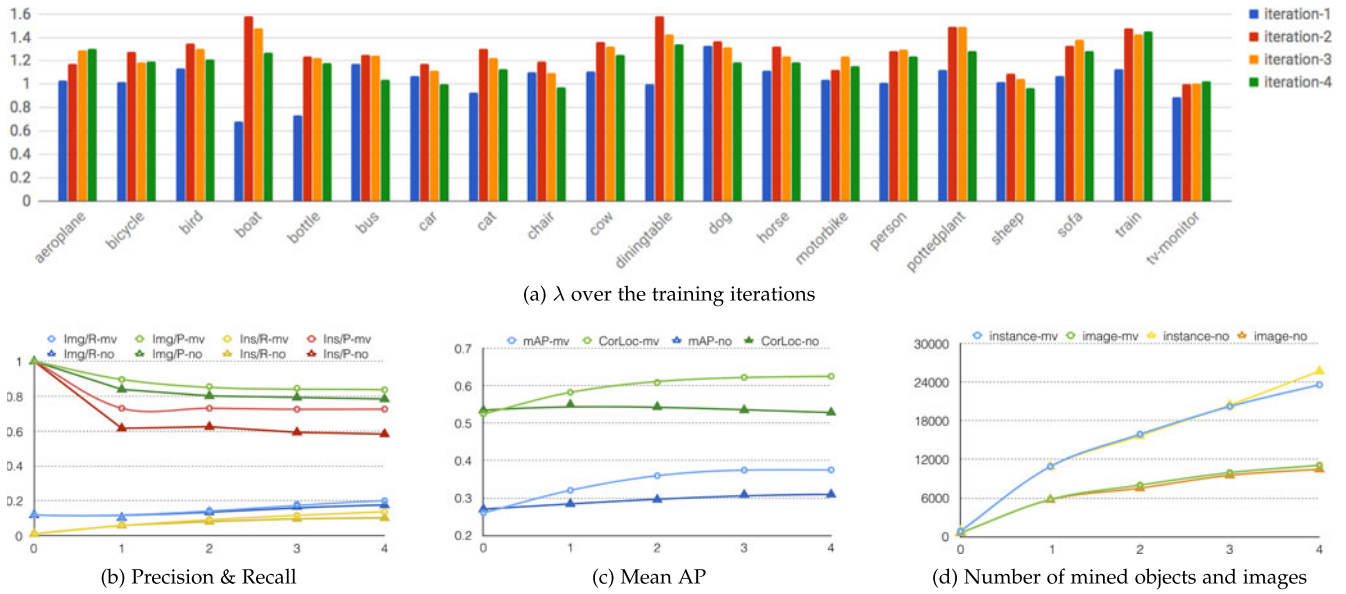


Fig. 3. The change of  $\lambda$ , precision, recall and mAP for the first four training iterations of MSPLD. “mv” and “no” denote using and not using multi-modal learning, respectively. “Img/R” and “Ins/R” indicate the image-level and instance-level recall, respectively. “Img/P” and “Ins/P” indicate the image-level and instance-level precision, respectively.

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  $O(Flops)$ , 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 strategy and the number of detectors. Based on Algorithm 1, the time complexity of MSPLD is  $O(iter_{max} \times m \times Flops)$ , where the  $m$  is the number of detectors and  $iter_{max}$  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. Therefore, 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 first. Second, we analysis the effect of different aspects of MSPLD to demonstrate the performance contribution of each composition in MSPLD. Third, we show the impact of supervision level in our algorithm by using different annotation information. Lastly, with the visualized error analysis, we show how to further improve MSPLD in the future.

### 4.1 Datasets

We evaluate our method on PASCAL VOC 2007 [66], PASCAL VOC 2012 [67], MS COCO 2014 [68], and ILSVRC 2013 detection dataset [69]. These four datasets are the most widely used benchmarks in the object detection task. PASCAL VOC 2007 contains 10,022 images annotated with bounding boxes for 20 object categories. It is officially split into 2,501 training,

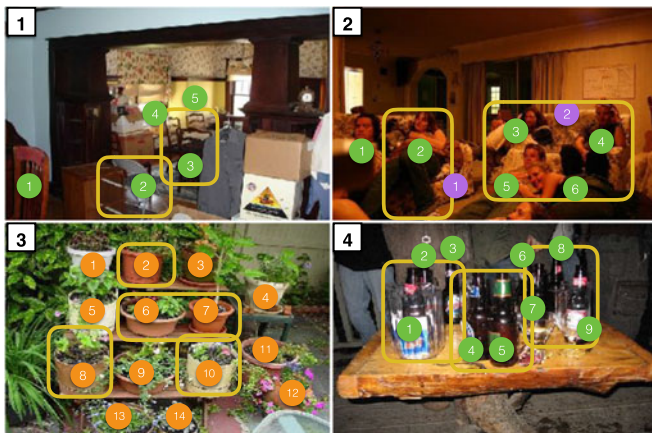


Fig. 4. Some poorly located or missed training samples. The yellow rectangles are the generated labeled boxes, and the discs denote the ground-truth objects. In image 2, the green and purple circles indicate people and sofa, respectively. We observe that the sofa is missed due to occlusions and different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

TABLE 2  
Method Comparisons in Average Precision (AP) on the PASCAL VOC 2007 Test Set

Methods	aero	bike	bird	boat	botl	bus	car	cat	chair	cow	table	dog	hors	mbik	pers	plnt	shp	sofa	train	tv	mean
Zhang et al. [61] <sup>*</sup>	47.4	22.3	35.3	23.2	13.0	50.4	48.0	41.8	1.8	28.9	27.8	37.7	41.6	43.8	20.0	12.0	27.8	22.9	48.9	31.6	31.3
Teh et al. [62] <sup>*</sup>	48.8	45.9	37.4	26.9	9.2	50.7	43.4	43.6	10.6	35.9	27.0	38.6	48.5	43.8	24.7	12.1	29.0	23.2	48.8	41.9	34.5
Kantorov et al. [63] <sup>*</sup>	57.1	52.0	31.5	7.6	11.5	55.0	53.1	34.1	1.7	33.1	49.2	42.0	47.3	56.6	15.3	12.8	24.8	48.9	44.4	47.8	36.3
Bilen et al. [7] <sup>*</sup>	46.4	58.3	35.5	25.9	14.0	66.7	53.0	39.2	8.9	41.8	26.6	38.6	44.7	59.0	10.8	17.3	40.7	49.6	56.9	50.8	39.3
Li et al. [8] <sup>*</sup>	54.5	47.4	41.3	20.8	17.7	51.9	63.5	46.1	21.8	57.1	22.1	34.4	50.5	61.8	16.2	29.9	40.7	15.9	55.3	40.2	39.5
Diba et al. [13] <sup>*</sup>	49.5	60.6	38.6	29.2	16.2	70.8	56.9	42.5	10.9	44.1	29.9	42.2	47.9	64.1	13.8	23.5	45.9	54.1	60.8	54.5	42.8
Dong et al. [57] <sup>*</sup>	62.5	54.6	44.3	12.9	12.7	63.8	60.6	25.0	5.4	48.0	49.3	58.7	66.6	63.5	8.5	17.3	40.7	59.4	53.9	51.4	43.0
Tang et al. [10] <sup>*</sup>	65.5	67.2	47.2	21.6	22.1	68.0	68.5	35.9	5.7	63.1	49.5	30.3	64.7	66.1	13.0	25.6	50.0	57.1	60.2	59.0	47.0
Ge et al. [29] <sup>*</sup>	64.3	68.0	56.2	36.4	23.1	68.5	67.2	64.9	7.1	54.1	47.0	57.0	69.3	65.4	20.8	23.2	50.7	59.6	65.2	57.0	51.2
SPL+Fast R-CNN	41.4	55.9	24.5	15.7	22.4	37.3	52.4	37.9	14.3	17.5	33.0	27.9	41.4	50.2	36.7	19.5	27.2	46.0	47.5	26.0	33.7 ± 0.5
SPL+R-FCN	25.6	34.3	26.0	15.3	22.3	39.3	48.8	30.4	18.8	17.3	2.2	18.6	40.9	54.8	35.4	13.5	26.6	36.1	52.1	35.8	29.9 ± 1.1
SPL+Ensemble	38.4	51.1	41.4	21.6	25.9	45.0	57.6	50.0	22.0	21.7	7.5	23.8	47.4	56.0	43.4	22.1	31.3	46.1	57.8	42.0	37.6 ± 0.8
MSPLD	46.6	55.6	37.9	26.1	27.9	46.6	57.9	58.1	24.1	37.6	12.8	33.1	51.4	59.7	40.1	17.5	36.1	52.0	61.4	52.1	41.7 ± 0.3

<sup>\*</sup>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. [61] 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<sup>ohem</sup>. "SPL+Ensemble" ensembles the three models: Fast R-CNN with VGG16, R-FCN with ResNet50<sup>ohem</sup> and R-FCN with ResNet101.

2,510 validation, and 5,011 testing images. PASCAL VOC 2012 is similar to PASCAL VOC 2007, but contains more images: 5,717 training, 5,823 validation images and 10,991 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 large 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

*The Details of Detection Models.* We build R-FCN and Fast R-CNN on various base models as different detection models. Three base models are tested in our experiments,<sup>3</sup> i.e., GoogleNet [70], VGG [71], and ResNet [72]. These models are pre-trained on ILSVRC 2012 [73]. A boosting method, i.e., online hard example mining (OHEM) [74], is also tested in our experiments to study the complementarity between different models. Region proposals are extracted by SS [24] using the fast version or EB [25], following the standard practice used in [7], [15], [16]. We extract about 2000 proposals using SS and EB, respectively. Proposals are extracted by SS in most experiments by default. When we use both SS and EE (denoted SS+EE) to extract proposals, the total generated proposals are about 4,000 for each image. We use ImageNet pre-trained models to make a fair comparison with other algorithms [12], [13], [61], [62], because they also utilize ImageNet pre-trained models to provide a good initialization.

*Hyper-Parameters.* We do not tune the parameter  $\gamma^{j_1, j_2}$  and always set it to  $0.2/(m-1)$  in all our experiments for simplicity. In our experiments, for the  $c$ th class,  $\lambda_c$  is decided by the number of selected images  $|V_c|$ , where  $V_c$  is the  $c$ th column of  $V$ . In fact, according to Eq. (10), a specific  $\lambda_c$  corresponds to a specific  $V_c$ . Moreover, given a specific

$V_c$ , there is one  $\lambda_c$  that can correspond to such  $V_c$ . Therefore, we can use  $|V_c|$  instead of  $\lambda_c$  to compute  $V_c$  in Eq. (10). In the implementation process of our experiments, supposing the number of selected images for the  $c$ th class is  $R_k = |V_c|$  at the  $k$ th iteration, then this number will increase to  $\frac{R_k(k+1)}{k}$  at the  $(k+1)$ th iteration. At the first iteration,  $R_k$  is the initial number of labeled images for each class. During basic detector training, we set the total training epochs to nine. We empirically use the learning rate 0.001 for the first eight epochs and reduce it to 0.0001 for the last epoch. In addition, the momentum and weight decay are set to 0.9 and 0.0005, respectively. The first two convolution layers of each network are fixed, following [15], [16]. We randomly flip the image for data augmentation in the training phase.

*Evaluation Metrics.* Average precision (AP) is used on the testing data to evaluate detection accuracy; correct localization (CorLoc) [64] is calculated for the training data to evaluate localization accuracy; the location prediction mAP is calculated for the validation data to evaluate location prediction accuracy, following [9]. We use an intersection-over-union (IoU) ratio of 50 percent for CorLoc and leverage the official evaluation code provided by [66] 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.

## 4.3 Comparison with State-of-the-Art Algorithms

We compare MSPLD with recent state-of-the-art WSOD algorithms [7], [8], [12], [13], [61], [62], [63]. Fair comparisons are claimed because many of these methods use multiple models as well. Bilen et al. [7] use ensembles to improve performance. Li et al. [8] use multiple steps. They first train a classification model and apply a MIL model to mine the confident objects, and then fine-tune a detection model to detect the objects. Diba et al. [13] cascade three networks: a location network, a segmentation network and a MIL network, and apply multi-scale data argumentation. 'SPL+Ensemble' in Tables 2 and 3 represents the late fusion of

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.



TABLE 3  
Method Comparisons in Correct Localization (CorLoc [64]) on the PASCAL VOC 2007 Trainval Set

Methods	aero	bike	bird	boat	botl	bus	car	cat	chair	cow	table	dog	hors	mbik	pers	plnt	shp	sofa	train	tv	mean
Zhang [61]*	75.7	37.9	68.3	53.2	11.9	57.1	59.6	63.7	16.4	63.9	17.5	62.3	71.6	71.5	45.6	14.7	53.1	41.1	75.5	24.4	49.3
Li et al. [8]*	78.2	67.1	61.8	38.1	36.1	61.8	78.8	55.2	28.5	68.8	18.5	49.2	64.1	73.5	21.4	47.4	64.6	22.3	60.9	52.3	52.4
Bilen et al. [7]*	73.1	68.7	52.4	34.3	26.6	66.1	76.7	51.6	15.1	66.7	17.5	45.4	71.8	82.4	32.6	42.9	71.9	53.3	60.9	65.2	53.8
Kantorov et al. [63]*	83.3	68.6	54.7	23.4	18.3	73.6	74.1	54.1	8.6	65.1	47.1	59.5	67.0	83.5	35.3	39.9	67.0	49.7	63.5	65.2	55.1
Diba et al. [13]*	83.9	72.8	64.5	44.1	40.1	65.7	82.5	58.9	33.7	72.5	25.6	53.7	67.4	77.4	26.8	49.1	68.1	27.9	64.5	55.7	56.7
Zhu et al. [65]*	85.3	64.2	67.0	42.0	16.4	71.0	64.7	88.7	20.7	63.8	58.0	84.1	84.7	80.0	60.0	29.4	56.3	68.1	77.4	30.5	60.6
Dong et al. [57]*	85.3	71.9	66.8	27.0	26.5	81.2	78.5	36.1	17.2	80.6	61.8	76.1	86.3	83.6	22.2	43.6	74.8	60.6	67.6	70.5	60.9
Tang et al. [10]*	85.8	82.7	62.8	45.2	43.5	84.8	87.0	46.8	15.7	82.2	51.0	45.6	83.7	91.2	22.2	59.7	75.3	65.1	76.8	78.1	64.3
Teh et al. [62]*	84.0	64.6	70.0	62.4	25.8	80.6	73.9	71.5	35.7	81.6	46.5	71.2	79.1	78.8	56.7	34.3	69.8	56.7	77.0	72.7	64.6
SPL+Fast R-CNN	63.3	72.3	49.6	43.8	42.4	54.4	78.7	58.1	35.4	72.8	43.0	63.1	78.1	82.3	59.1	37.8	68.8	56.6	64.5	51.7	58.8 ± 0.7
SPL+R-FCN	39.2	54.8	59.0	38.6	34.5	53.7	73.7	62.2	36.2	73.6	8.0	61.8	75.1	78.9	57.1	22.1	75.5	45.5	67.9	47.4	53.2 ± 1.2
SPL+Ensemble	54.6	65.0	71.2	50.8	52.1	62.4	81.9	67.7	41.4	74.5	21.0	69.6	78.4	86.5	66.5	46.1	76.0	57.6	74.7	56.3	62.7 ± 0.9
<b>MSPLD</b>	66.0	71.2	67.9	49.7	52.9	68.8	82.6	76.6	42.5	81.6	24.0	75.5	78.4	89.0	62.0	33.1	79.2	58.5	78.9	71.1	<b>65.5 ± 0.3</b>

\*indicates the usage of full image-level labels for training. The models that we use are the same as Table 2.

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 initialization, 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.

*Comparisons w.r.t. AP.* 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 percent of the total number of training images, MSPLD achieves 41.7 percent 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 percent, respectively. We view [61] 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 [61] by 2.4 and 10.3 percent in mAP and CorLoc, respectively. The *SPL+Fast R-CNN* model is superior to *SPL+R-FCN*, because Fast R-CNN may pay more attention to the pseudo boxes selection and thus benefits more from the SPL strategy. However, the two different architectures complement each other well, demonstrated by the improved performance of the *SPL+Ensemble*. Further, the proposed MSPLD is superior to the multi-model ensemble. From Tables 2 and 3, MSPLD outperforms the model ensemble method by about 4 percent in mAP and 3 percent in CorLoc. This observation further validates the effectiveness of our multi-model learning strategy.

*Comparisons w.r.t. CorLoc.* Table 3 shows the correct localization on the PASCAL VOC 2007 trainval set. MSPLD achieves an average CorLoc 65.5 percent, which sets a new state-of-the-art. Note that [62] has a similar CorLoc to our MSPLD, but we obtain a much higher mAP than [62] (41.7

versus 34.5 percent). From Tables 2 and 3, it can be seen that our method does not have large performance deviations under different initializations of fully annotated images. Moreover, it can be observed from the tables that when using multiple models, the performance of our method is less sensitive to different initializations than that of the baseline single model. In Tables 2 and 3, we note that recent works [10], [29] report very competitive accuracy with ours. Their methods work under the traditional weakly-supervised setting, while our method is implemented under semi-supervised learning setting with only very few examples provided. Specifically, Tang et al. [10] and Ge et al. [29] achieve higher mAP than MSPLD, and MSPLD achieves higher CorLoc than [10]. Two reasons may contribute to their higher mAP. First, they use superior architectures to generate region proposals rather than the selective search method in our work. Second, they employ multi-scale training strategies, but we use a single-scale training strategy. The advantage of our work over [10], [29] is that we are able to make better use multiple models to improve the performance of a single model.

*Results on Large-Scale Datasets.* Table 5 a presents the mAP and CorLoc of MSPLD on PASCAL VOC 2012, which also achieves the competitive performance compared with others. We also compared our algorithm on ILSVRC 13 only with [8], [11], [12], [13], since no other weakly supervised or few shot algorithms have been tried on this dataset. Results on Table 5b are similar to the previous one, we achieves the competitive performance with fewer annotation informations on ILSVRC 2013 validation set. Following [9], Table 5c uses the location prediction [9] mean average precision to compare our results with others on MS COCO 2014. As shown in Table 5, our algorithm achieves competitive or superior results on the large-scale detection datasets.

*Comparison of Different Variants.* We compare the impact of different proposal generations methods. SS, EB and their combination are tested. The results are presented in Table 4. We find that EB is inferior to SS due to its poorer initialization in the first iteration. Combining both of the two region proposals, we obtain a slight performance improvement.

*The Effect of Multi-Modal Learning.* Furthermore, we demonstrate the performance of the individual detection models with and without multi-modal learning in Table 6. The displayed models are used with MSPLD shown in Table 2. We

TABLE 4  
Performance Comparison on PASCAL VOC 2007  
of Different Proposal Generation Methods

	Selective Search	EdgeBox	Selective Search + EdgeBox
mAP	41.7	39.5	<b>41.9</b>
CorLoc	65.5	65.2	<b>65.6</b>

observe that the performance of individual detection models is much higher when using multi-modal learning, which proves the effectiveness of our method in enhancing each model.

#### 4.4 Ablation Studies

We examine the contribution of different components of MSPLD on PASCAL VOC 2007 and MS COCO 2014.

*The Impact of Different Models and the Curriculum Regime  $\Psi$ .* From Table 7, several conclusions can be made. (1) Since *R-R50* outperforms *R-R50 no SPL* and *R-R50 no prior*, we prove that the data selection strategy and prior knowledge are necessary. (2) Fast R-CNN with VGG16 achieves the best single model performance. (3) We observe that *R-R50* and *F-VGG16* are complementary and benefit from the multi-modal learning. The reason may be that R-FCN has the position-sensitive layer for box refinement, while Fast R-CNN with VGG-16 focuses more on the proposals' classification. (4) The use of *ohem* slightly improves mAP for *R-R50*, but harms the performance of *F-VGG16* and *R-R101*. (5) When adding *ohem* to *R-R101* or to *F-VGG16*, we observe inferior results. The probable reason for this observation is that VGG16 and ResNet-101 are larger than ResNet-50 and

TABLE 5  
Performance Comparison on the PASCAL VOC 2012, MS  
COCO 2014, and ILSVRC 2013 Datasets

(a) PASCAL VOC 2012		
Methods	mAP	CorLoc
Li et al. [8]	29.1	-
Kantorov et al. [63]	35.3	54.8
Diba et al. [13]	<b>37.9</b>	-
<b>MSPLD</b>	35.4	<b>64.6</b>
(b) ILSVRC 2013		
Methods	mAP	
Wang et al. [12]	6.0	
Felzenszwalb et al. [11]	8.8	
Li et al. [8]	10.8	
Diba et al. [13]	<b>16.3</b>	
<b>MSPLD</b>	13.9	
(c) MS COCO 2014		
Methods	mAP	
Oquab et al. [9]	41.2	
Sun et al. [75]	43.5	
Bency et al. [76]	47.9	
Zhu et al. [65]	55.3	
<b>MSPLD</b>	<b>56.6</b>	

On PASCAL VOC 2012, mAP is evaluated on the test set and CorLoc is evaluated on the trainval set. On ILSVRC 2013, we show the detection performance on the validation set. On MS COCO 2014, we use the location prediction mAP for evaluation, following the same setting in [9].

TABLE 6  
The Performance of Each Detector Employed in MSPLD

Models	Eval.	MV	w/o MV
Fast R-CNN (VGG16)	mAP	36.0	33.7
	CorLoc	60.9	58.8
R-FCN (Res50 <sup>ohem</sup> )	mAP	37.4	29.9
	CorLoc	62.7	53.2
R-FCN (Res101)	mAP	38.3	31.4
	CorLoc	62.0	54.1

"MV" indicates the use of multi-modal learning. "w/o MV" indicates we use the traditional self-paced method without multi-modal learning.

that the training set is relatively small (in our few-example setting). Therefore, the influence of *ohem* on VGG-16 and ResNet-101 is limited or even negative.

*The Impact of the Number of Initial Labels.* Using  $k = 2$  (totally 40 images in PASCAL VOC 2007) for initialization is not stable for training, and can result into severely reduced accuracy. We can observe that even one additional example per class could significantly improve the performance of our MSPLD. In Fig. 5, each category has a maximum of 250 images on average, which can reproduce a fully supervised object detector [15], [16]. In our method, when 100 images are randomly selected during initialization, we can obtain very close accuracy to the fully-supervised method. In this paper, we choose to use only 3-4 images which will suffice to ensure a decent accuracy at little manual cost.

*The Impact of Image-Level Labels.* Image-level supervision can be easily incorporated into our framework. We use the simplest approach to embed this supervision, i.e., only using the image label to filter out incorrect pseudo boxes. The results are shown in Fig. 5. The simplest method for appending image-level labels can greatly boost our framework.

TABLE 7  
Ablation Studies

#Models	Detection Model	mAP	CorLoc
1	R-R50 no prior	28.6	50.1
	R-R50 no SPL	27.2	44.7
	R-R50	28.9	50.6
	R-R50 <sup>ohem</sup>	29.9	53.2
	R-Gog <sup>ohem</sup>	24.9	50.6
	F-VGG16 no prior	32.8	60.1
2	F-VGG16	33.7	60.9
	R-R50 <sup>ohem</sup> + F-VGG16	38.3	63.4
	R-R50 <sup>ohem</sup> + R-Gog <sup>ohem</sup>	32.1	57.3
3	R-Gog <sup>ohem</sup> + F-VGG16	35.8	61.6
	R-R50 <sup>ohem</sup> + F-VGG16 + R-Gog <sup>ohem</sup>	38.5	62.8
	R-R50 <sup>ohem</sup> + F-VGG16 + R-R101	<b>41.7</b>	<b>65.5</b>
	R-R50 <sup>ohem</sup> + F-VGG16 + R-R101 <sup>ohem</sup>	38.9	63.4
	R-R50 <sup>ohem</sup> + F-VGG16 <sup>ohem</sup> + R-R101	37.5	61.4
R-R50 <sup>ohem</sup> + F-VGG16 <sup>ohem</sup> + R-R101 <sup>ohem</sup>	37.1	61.1	

"#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.

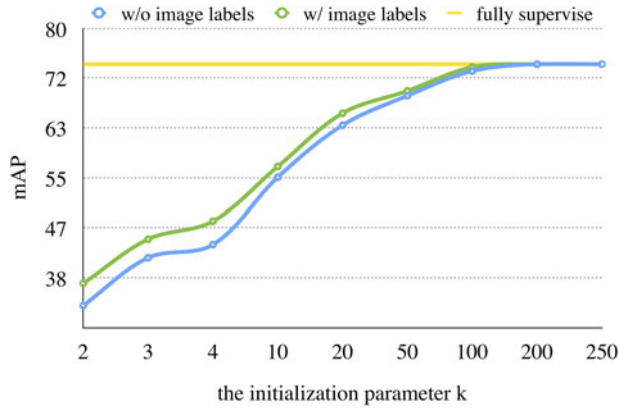


Fig. 5. Performance comparison of MSPLD on PASCAL VOC 2007 using different selection numbers for the initial labeled images. In “w/ image label”, we simply leverage the image label to filter the undesired pseudo boxes.

*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 [77] 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 1,000 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 constituted

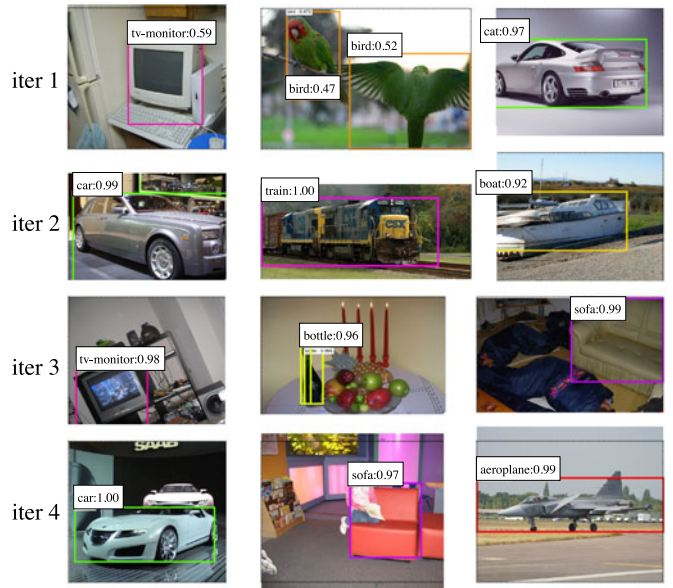


Fig. 6. Qualitative results of MSPLD over the training iterations. The boxes with different colors indicate the generated pseudo boxes by our method for different classes.

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 percent; the localization prediction mAP on MS COCO 2014 drops from 56.6 to 53.3 percent. 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 percent 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 percent. 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 in Fig. 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 situation, e.g., the rotated TV monitor and several small bottles in Fig. 6.

*Error Analysis.* Some of the images that are newly generated by our method are shown in Fig. 7. We observe that the generated pseudo boxes have the good localization accuracy, but cannot detect every object in complex images. For

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

#noisy images	0	1,000	2,000	5,000	10,000
noise scale	0%	20%	40%	100%	200%
mAP	41.7	39.9	39.8	39.8	39.3
CorLoc	65.5	64.3	64.0	63.9	63.5



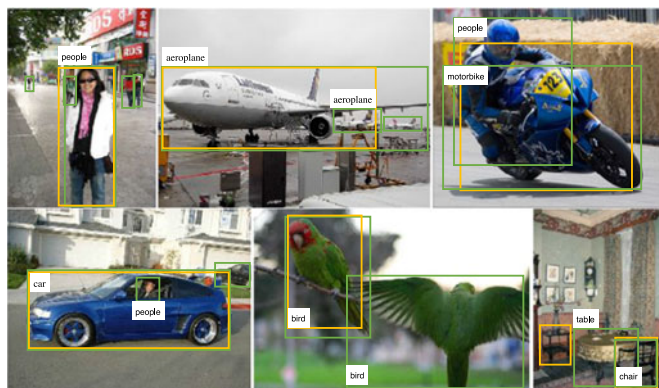


Fig. 7. Qualitative results of the inaccurate pseudo instance-level labels generated by MSPLD during the training procedure. The green boxes indicate the ground-truth object annotation. The yellow boxes indicate the generated pseudo boxes by MSPLD. The white blocks show the object classes.

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 percent 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.

## ACKNOWLEDGMENTS

This research was supported by the National Key R&D Program of China (2018YFB1004300), the Data to Decisions CRC (D2D CRC), the Cooperative Research Centres Programme and the China NSFC projects under contracts 61661166011, 11690011, 61603292, 61721002. Liang Zheng is partially supported by SIEF STEM+ Business Fellowship Program.

## REFERENCES

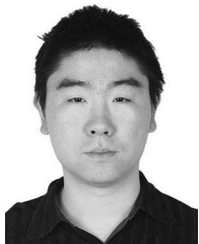
- [1] X. Liang, S. Liu, Y. Wei, L. Liu, L. Lin, and S. Yan, "Towards computational baby learning: A weakly-supervised approach for object detection," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 999–1007.
- [2] K. K. Singh, F. Xiao, and Y. J. Lee, "Track and transfer: Watching videos to simulate strong human supervision for weakly-supervised object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3548–3556.
- [3] I. Misra, A. Shrivastava, and M. Hebert, "Watch and learn: Semi-supervised learning for object detectors from video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3593–3602.
- [4] M. Rochan and Y. Wang, "Weakly supervised localization of novel objects using appearance transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 4315–4324.
- [5] Y. Tang, J. Wang, B. Gao, E. Dellandrea, R. Gaizauskas, and L. Chen, "Large scale semi-supervised object detection using visual and semantic knowledge transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2119–2128.
- [6] J. Hoffman, S. Guadarrama, E. S. Tzeng, R. Hu, J. Donahue, R. Girshick, T. Darrell, and K. Saenko, "LSDA: Large scale detection through adaptation," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 3536–3544.
- [7] H. Bilen and A. Vedaldi, "Weakly supervised deep detection networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2846–2854.
- [8] D. Li, J.-B. Huang, Y. Li, S. Wang, and M.-H. Yang, "Weakly supervised object localization with progressive domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3512–3520.
- [9] M. Oquab, L. Bottou, I. Laptev, and J. Sivic, "Is object localization for free?-weakly-supervised learning with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 685–694.
- [10] P. Tang, X. Wang, X. Bai, and W. Liu, "Multiple instance detection network with online instance classifier refinement," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2843–2851.
- [11] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, Sep. 2010.
- [12] C. Wang, W. Ren, K. Huang, and T. Tan, "Weakly supervised object localization with latent category learning," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 431–445.
- [13] A. Diba, V. Sharma, A. Pazandeh, H. Pirsiavash, and L. Van Gool, "Weakly supervised cascaded convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 914–922.
- [14] Y.-X. Wang and M. Hebert, "Model recommendation: Generating object detectors from few samples," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1619–1628.
- [15] J. Dai, Y. Li, K. He, and J. Sun, "R-FCN: Object detection via region-based fully convolutional networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 379–387.
- [16] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448.
- [17] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 21–37.
- [18] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 779–788.
- [19] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1137–1149, Jun. 2017.
- [20] C. Gao, D. Meng, Y. Yang, Y. Wang, X. Zhou, and A. G. Hauptmann, "Infrared patch-image model for small target detection in a single image," *IEEE Trans. Image Process.*, vol. 22, no. 12, pp. 4996–5009, Dec. 2013.
- [21] J. Wang, J. Yao, Y. Zhang, and R. Zhang, "Collaborative learning for weakly supervised object detection," in *Proc. Int. Joint Conf. Artif. Intell.*, 2018.
- [22] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 580–587.
- [23] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 346–361.

- [24] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, 2013.
- [25] C. L. Zitnick and P. Dollár, "Edge boxes: Locating object proposals from edges," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 391–405.
- [26] Y. Yang, G. Shu, and M. Shah, "Semi-supervised learning of feature hierarchies for object detection in a video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 1650–1657.
- [27] H. O. Song, R. Girshick, S. Jegelka, J. Mairal, Z. Harchaoui, and T. Darrell, "On learning to localize objects with minimal supervision," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1611–1619.
- [28] H. O. Song, Y. J. Lee, S. Jegelka, and T. Darrell, "Weakly-supervised discovery of visual pattern configurations," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 1637–1645.
- [29] W. Ge, S. Yang, and Y. Yu, "Multi-evidence filtering and fusion for multi-label classification, object detection and semantic segmentation based on weakly supervised learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1277–1286.
- [30] K. Levi and Y. Weiss, "Learning object detection from a small number of examples: The importance of good features," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2004, pp. II-53–II-60.
- [31] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al., "Matching networks for one shot learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2016, pp. 3630–3638.
- [32] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, and T. Lillicrap, "Meta-learning with memory-augmented neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1842–1850.
- [33] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 594–611, Apr. 2006.
- [34] Z. Xu, L. Zhu, and Y. Yang, "Few-shot object recognition from machine-labeled web images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1164–1172.
- [35] X. Chen and A. Gupta, "Webly supervised learning of convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1431–1439.
- [36] S. K. Divvala, A. Farhadi, and C. Guestrin, "Learning everything about anything: Webly-supervised visual concept learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 3270–3277.
- [37] K. Tang, A. Joulin, L.-J. Li, and L. Fei-Fei, "Co-localization in real-world images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2014, pp. 1464–1471.
- [38] M. Rubinstein, A. Joulin, J. Kopf, and C. Liu, "Unsupervised joint object discovery and segmentation in internet images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 1939–1946.
- [39] M. Cho, S. Kwak, C. Schmid, and J. Ponce, "Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3485–3492.
- [40] S. Rahman, S. Khan, and F. Porikli, "Zero-shot object detection: Learning to simultaneously recognize and localize novel concepts," arXiv preprint arXiv:1803.06049, 2018.
- [41] A. Bansal, K. Sikka, G. Sharma, R. Chellappa, and A. Divakaran, "Zero-shot object detection," arXiv preprint arXiv:1804.04340, 2018.
- [42] R. Krishna, Y. Zhu, O. Groth, J. Johnson, K. Hata, J. Kravitz, S. Chen, Y. Kalantidis, L.-J. Li, D. A. Shamma, et al., "Visual genome: Connecting language and vision using crowdsourced dense image annotations," *Int. J. Comput. Vis.*, vol. 123, no. 1, pp. 32–73, 2017.
- [43] S. Dai, M. Yang, Y. Wu, and A. Katsaggelos, "Detector ensemble," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2007, pp. 1–8.
- [44] T. Malisiewicz, A. Gupta, and A. A. Efros, "Ensemble of exemplar-SVMs for object detection and beyond," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 89–96.
- [45] Y. Yang, Z. Ma, A. G. Hauptmann, and N. Sebe, "Feature selection for multimedia analysis by sharing information among multiple tasks," *IEEE Trans. Multimedia*, vol. 15, no. 3, pp. 661–669, Apr. 2013.
- [46] Z. Zheng, L. Zheng, and Y. Yang, "A discriminatively learned CNN embedding for person reidentification," *ACM Trans. Multimedia Comput. Commun. Appl.*, vol. 14, no. 1, 2017, Art. no. 13.
- [47] Z. Ma, X. Chang, Y. Yang, N. Sebe, and A. G. Hauptmann, "The many shades of negativity," *IEEE Trans. Multimedia*, vol. 19, no. 7, pp. 1558–1568, Jul. 2017.
- [48] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proc. Int. Conf. Mach. Learn.*, 2009, pp. 41–48.
- [49] M. P. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2010, pp. 1189–1197.
- [50] L. Jiang, D. Meng, S.-I. Yu, Z. Lan, S. Shan, and A. Hauptmann, "Self-paced learning with diversity," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2078–2086.
- [51] L. Jiang, D. Meng, Q. Zhao, S. Shan, and A. G. Hauptmann, "Self-paced curriculum learning," in *Proc. AAAI Conf. Artif. Intell.*, 2015, pp. 2694–2700.
- [52] F. Ma, D. Meng, Q. Xie, Z. Li, and X. Dong, "Self-paced co-training," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 2275–2284.
- [53] L. Lin, K. Wang, D. Meng, W. Zuo, and L. Zhang, "Active self-paced learning for cost-effective and progressive face identification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 1, pp. 7–19, Jan. 2018.
- [54] D. Meng, Q. Zhao, and L. Jiang, "A theoretical understanding of self-paced learning," *Inf. Sci.*, vol. 414, pp. 319–328, 2017.
- [55] Z. Ma, S. Liu, D. Meng, Y. Zhang, S.-L. Lo, and Z. Han, "On convergence properties of implicit self-paced learning," *Inf. Sci.*, vol. 462, pp. 132–140, 2018.
- [56] Y. Yan, F. Nie, W. Li, C. Gao, Y. Yang, and D. Xu, "Image classification by cross-media active learning with privileged information," *IEEE Trans. Multimedia*, vol. 18, no. 12, pp. 2494–2502, Dec. 2016.
- [57] X. Dong, D. Meng, F. Ma, and Y. Yang, "A dual-network progressive approach to weakly supervised object detection," in *Proc. ACM Multimedia Conf.*, 2017, pp. 279–287.
- [58] L. Zheng, Y. Yang, and Q. Tian, "SIFT meets CNN: A decade survey of instance retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 5, pp. 1224–1244, May 2018.
- [59] Y. Wei, X. Liang, Y. Chen, X. Shen, M.-M. Cheng, J. Feng, Y. Zhao, and S. Yan, "STC: A simple to complex framework for weakly-supervised semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 11, pp. 2314–2320, Nov. 2017.
- [60] X. Liang, Y. Wei, L. Lin, Y. Chen, X. Shen, J. Yang, and S. Yan, "Learning to segment human by watching YouTube," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 7, pp. 1462–1468, Jul. 2017.
- [61] D. Zhang, D. Meng, L. Zhao, and J. Han, "Bridging saliency detection to weakly supervised object detection based on self-paced curriculum learning," in *Proc. Int. Joint Conf. Artif. Intell.*, 2016, pp. 3538–3544.
- [62] E. W. Teh, M. Roohan, and Y. Wang, "Attention networks for weakly supervised object localization," in *Proc. Brit. Mach. Vis. Conf.*, 2016, pp. 52.1–52.11.
- [63] V. Kantorov, M. Oquab, M. Cho, and I. Laptev, "ContextLocNet: Context-aware deep network models for weakly supervised localization," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 350–365.
- [64] T. Deselaers, B. Alexe, and V. Ferrari, "Weakly supervised localization and learning with generic knowledge," *Int. J. Comput. Vis.*, vol. 100, no. 3, pp. 275–293, 2012.
- [65] Y. Zhu, Y. Zhou, Q. Ye, Q. Qiu, and J. Jiao, "Soft proposal networks for weakly supervised object localization," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 1841–1850.
- [66] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.
- [67] M. Everingham, S. A. Eslami, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes challenge: A retrospective," *Int. J. Comput. Vis.*, vol. 111, no. 1, pp. 98–136, 2015.
- [68] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.
- [69] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.
- [70] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1–9.
- [71] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, 2015.
- [72] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
- [73] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

- [74] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 761–769.
- [75] C. Sun, M. Paluri, R. Collobert, R. Nevatia, and L. Bourdev, "ProNet: Learning to propose object-specific boxes for cascaded neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 3485–3493.
- [76] A. J. Bency, H. Kwon, H. Lee, S. Karthikeyan, and B. Manjunath, "Weakly supervised localization using deep feature maps," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 714–731.
- [77] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L.-J. Li, "YFCC100M: The new data in multimedia research," *Commun. ACM*, vol. 59, no. 2, pp. 64–73, 2016.



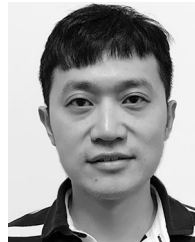
**Xuanyi Dong** received the BE degree in computer science and technology from Beihang University, Beijing, China, in 2016. He is currently working toward the PhD degree in the Center of Artificial Intelligence, University of Technology Sydney, Australia, under the supervision of Prof. Yi Yang.



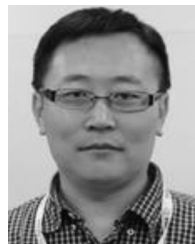
**Liang Zheng** received the BE degree in life science from Tsinghua University, China, in 2010, and the PhD degree in electronic engineering from Tsinghua University, China, in 2015. He was a postdoc researcher with the University of Texas at San Antonio, USA. He is now a postdoc researcher in the Center of Artificial Intelligence, University of Technology Sydney, Australia. His research interests include image retrieval, person re-identification, and deep learning.



**Fan Ma** received the BSc and MSc degrees from Xi'an Jiaotong University, Xi'an, China, in 2014 and 2017, respectively, under the supervision of Prof. Deyu Meng. He is currently working toward the PhD degree in the Center of Artificial Intelligence, University of Technology Sydney, Australia, under the supervision of Prof. Yi Yang. 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 PhD degree in computer science from Zhejiang University, Hangzhou, China, in 2010. He was a post-doctoral research in the School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA. He is currently a professor with the University of Technology Sydney, Australia. His current research interests include machine learning and its applications to multimedia content analysis and computer vision.



**Deyu Meng** received the BSc, MSc, and PhD degrees from Xi'an Jiaotong University, Xi'an, China, in 2001, 2004, and 2008, respectively. He is currently a professor in the Institute for Information and System Sciences, School of Mathematics and Statistics, Xi'an 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.

▷ For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/publications/dlib](http://www.computer.org/publications/dlib).