Received April 28, 2019, accepted June 2, 2019, date of publication June 12, 2019, date of current version June 27, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2922438*

# Meta-SSD: Towards Fast Adaptation for Few-Shot Object Detection With Meta-Learning

**KUN FU[1,2,3,4], TENGFEI ZHANG[1,2,4], (Student Member, IEEE), YUE ZHANG[1,2], (Member, IEEE), MENGLONG YAN[1,2], ZHONGHAN CHANG[1,2,4], ZHENGYUAN ZHANG[1,2,4], AND XIAN SUN[1,2]**

[1]Institute of Electronics, Chinese Academy of Sciences, Beijing 100190, China
[2]Key Laboratory of Network Information System Technology (NIST), Institute of Electronics, Chinese Academy of Sciences, Beijing 100190, China
[3]Institute of Electronics, Chinese Academy of Sciences, Suzhou 215000, China
[4]School of Electronic, Electrical and Communication Engineering, University of Chinese Academy of Sciences, Beijing 100190, China

Corresponding author: Tengfei Zhang (zhangtengfei16@mails.ucas.ac.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61725105 and Grant 41801349, and in part by the Gusu Innovation Talent Foundation of Suzhou under Grant ZXT2017002.

**ABSTRACT** The state-of-the-art object detection frameworks require the training on large-scale datasets, which is the crux of the present dilemma: overfitting or degrading performance with insufficient samples and time-consuming training process. On the basis of meta-learning, this paper proposes a generalized Few-Shot Detection (FSD) framework to overcome the above drawbacks of the current advances in object detection. The proposed framework consists of a meta-learner and an object detector. It can learn the general knowledge and proper fast adaptation strategies across many tasks. The meta-learner can teach the detector how to learn from few examples in just one updating step. Here, the object detector can be any supervised learning detection models in theory. Specifically, the proposed FSD framework employs Single-Shot MultiBox Detector (SSD) as the object detector in this paper, thus called Meta-SSD. Besides, a novel benchmark is constructed from Pascal VOC dataset for training and evaluation of meta-learning FSD. Experiments show that the Meta-SSD yields a promising result for FSD. Furthermore, the properties of Meta-SSD is analyzed. This paper can serve as a strong baseline and provide some inspiration for meta-learning FSD.

**INDEX TERMS** Meta-learning, few-shot, object detection, fast adaptation.

## I. INTRODUCTION

The current deep learning systems have achieved great success in image classification [1]–[4], object detection [5]–[13] and semantic segmentation [14]–[18]. Nevertheless, these state-of-the-art systems have to be trained for hundreds of thousands iterations on large-scale datasets, resulting in the characteristics of data-hungry and time-consuming. For object detection, in many situations, the insufficient examples will limit the performance of these supervised learning object detectors (Fig. 1 (a)). Moreover, collecting a large number of labelled examples is expensive and laborious. Hence, weakly supervised object detection [19]–[25] is proposed and has gained notable achievements to alleviate the heavy burdens for data annotation, which merely solves the dependencies on annotated examples, but still requires a large pool of training images.

In contrast, one can recognize a novel object with few samples or even one. The ability of few-shot learning in humans can be reference to the deep learning object detection methods and expand the application of the current advances. It has promoted related works in few-shot image classification [26]–[38], where the models can recognize new categories after updated a few steps or even once with few examples. While few researches focus on the few-shot object detection (FSD) problem.

As shown in Fig. 1 (b), transfer learning is a feasible method for FSD [39], where the model acquires priors by training on source domain, then, a finetuning process to transfer knowledge to target domain with few training examples. Nevertheless, the transfer learning models also need to be trained for hundreds or thousands of iterations on target domains, which may be unsuitable for dynamic environment and urgent tasks.

To further expand the flexibility and utility of the advanced object detection approaches, we address the few-shot detection (FSD) problem from a new perspective of

The associate editor coordinating the review of this manuscript and approving it for publication was Naveed Akhtar.

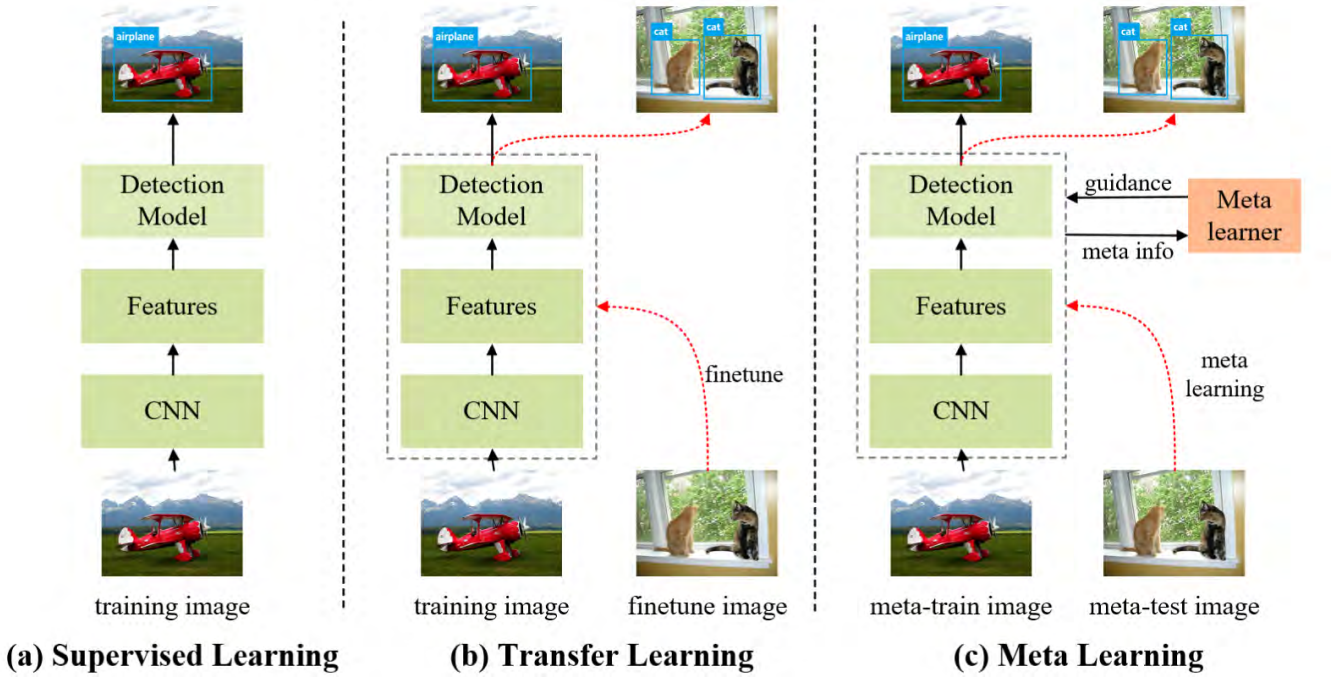


**FIGURE 1.** Comparison of three different object detection methods. (a) The supervised learning detectors work well on the objects of trained classes, e.g. aeroplane, but cannot detect the objects of unseen classes, e.g. cat. (b) Transfer learning approaches have found a solution by transfering the knowledge from source domain (aeroplane) to target domain (cat) of few examples per category. (c) The proposed meta-learning framework learns a meta-learner from a distribution of similar tasks of few examples, which can implement fast adaptation for new detection tasks with only few examples.

meta-learning, where models should achieve fast adaptation on new tasks with few examples. Here, fast adaptation means one step update on new tasks as in [38]. Considering that meta-learning is the basis of many excellent works with fast adaptation in few-shot image classification, which should be an alternative solution to solve the FSD problem. To the best of our knowledge, this work is the first study to propose a generalized meta-learning FSD methodology and analyze its feasibility.

The proposed meta-learning FSD framework, shown in Fig. 1 (c), containing a meta-learner and an object detector. Here, the meta-learning system learns a meta-learner from a series of FSD tasks, which can guide the detector how to update its network in a new task of few examples accurately and faster (usually updating only once). The learning process exists at two levels: rapid learning and gradual learning, shown in Fig. 2. **Rapid learning** occurs within each task for fast adaptation, where the detector updates its weights under the guidance of meta-learner to get an adapted detector which should be more suitable for this task. **Gradual learning** aims to acquire a general knowledge for all related tasks by updating the meta-learner with the meta-info from each batch of tasks. The meta-learning process is a life-long learning which can continue forever as long as provided similar tasks.

Specifically, we implement the generalized FSD framework by equipping the meta-learner with SSD, train and test Meta-SSD in the same process as meta-learning methods for few-shot image recognition. Besides, to measure the performance of the proposed meta-learning detection framework, we define a new benchmark NIST-FSD[1] on the Pascal VOC [40] dataset by splitting the classes into seen classes (for training) and unseeen classes (for test, not present in training phase).

Our contributions are as follows:

1) A generalized meta-learning FSD framework with a meta-learner and an object detector is proposed and implemented as Meta-SSD by transforming the supervised learning SSD to meta-learning FSD.
2) A benchmark NIST-FSD is built from Pascal VOC dataset for training and evaluation of the proposed meta-learning FSD framework and other FSD methods.
3) The feasibility of Meta-SSD is proved by experiments, furthermore, the properties of Meta-SSD are analyzed to promote the development of meta-learning FSD.

## II. RELATED WORKS

The goal of this work is to address the FSD problem by combining the meta-learning methodology with the supervised-learning object detection methods. Hence, we will expatiate related works in these areas.

### A. OBJECT DETECTION

The current advances [5]–[12] in object detection are based on Convolutional Neural Networks (CNN) and can mainly be classified into two categories: one-stage detectors and

[1]Code for building NIST-FSD is available at https://github.com/ztf-ucas/NIST-FSD

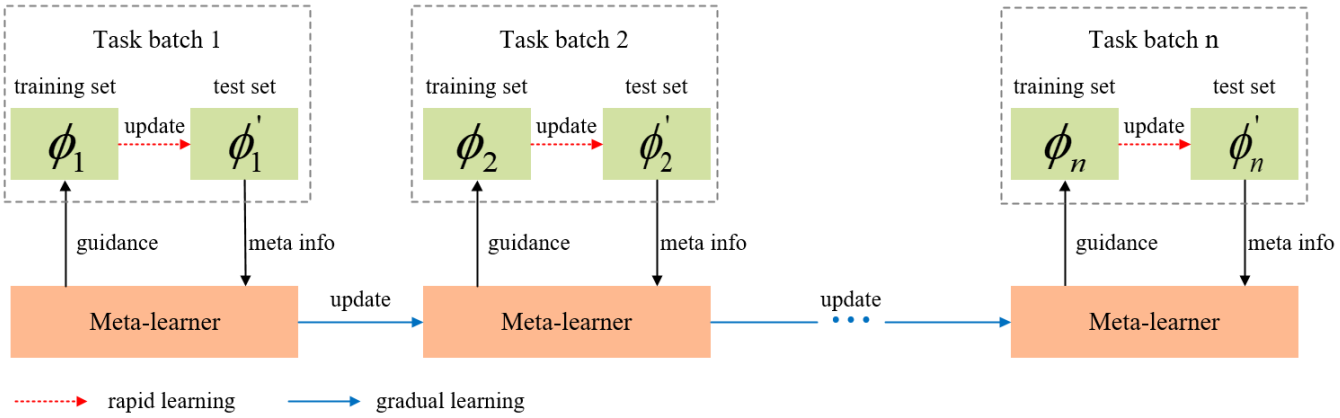


**FIGURE 2.** Meta-learning process of the generalized FSD framework. $\phi$ denotes the object detector. Meta-learner guides $\phi$ to adjust its parameters according to the feedback from the training set in each task, then is updated by leveraging the meta-info from the test-sets of a batch of tasks.

two-stage detectors. One-stage detectors are characterized with simpler structure and faster computation than two-stage detectors due to predicting object locations and classes in a straightforward one-step pipeline without region proposal process. In contrast, two-stage detectors are usually slower but more accurate with the region proposal methods to search a set of potential object locations, then classify the region proposals more precisely.

#### 1) ONE-STAGE DETECTORS

Overfeat [5] is the first integrated CNN based framework which can achieve object classification, localization and detection jointly. After that, many one-stage methods have been proposed, such as YOLO [7] and SSD [8]. YOLO makes predictions on a single scale feature map. To improve the accuracy, SSD adopts multi-scale feature maps and use convolutional layers instead of fully connected layers as in YOLO. Generally, one-stage detectors are faster than two-stage detectors but trailed in accuracy. [9] claims that performance degradation is due to the extreme foreground-background class imbalance during training, and it introduces focal loss to address this problem.

#### 2) TWO-STAGE DETECTORS

R-CNN [6] is the most representative approach among various two-stage detectors. In the first stage, it uses selective search to generate candidate regions. In the second stage, R-CNN performs forward calculations on the entire neural network for each proposal, leading to a heavy computational burden. Subsequent methods try to improve R-CNN in terms of speed by reducing redundant forward passes or the number of candidate regions. SPP-Net [11] and Fast R-CNN [10] extract feature maps for a test image only once. Faster R-CNN [12] achieves better performance by replacing selective search with region proposal network (RPN) and merging Fast R-CNN and RPN into a unified framework.

### B. META-LEARNING

Few-shot learning has made a splash in recent years [26]–[38], represented by a series of methods of metric-learning, transfer-learning and meta-learning. Meta-learning [41] is a general solution for few-shot learning, and have made a breakthrough progress in image recognition, regression and reinforcement learning. A standard meta-learning framework usually has two components: a meta-learner acting as a teacher and a learner viewed as a student. In the meta-learning regime, a meta-learner is trained on a distribution of similar tasks to teach the learner how to update its parameters.

Meta-learning is used to learn a generic transformation in [28], from models trained on few samples to those learnt from large-scale dataset to solve the problem of few-shot learning. While [27] learns a LSTM based meta-learning optimizer to train the learner on few-shot learning tasks. Besides, [26] learns a general initialization of the learner for rapid adaptation with few examples. Based on [26], Meta-SGD [38] is proposed to learn a set of good parameters as well as learning rates of each parameter. [29] employs a different approach with the memory-augmented model for rapid generalization on new tasks.

Most previous meta-learning works focus on image recognition, regression and reinforcement learning, few researches work toward a meta-learning system for FSD. Based on the fact that meta-learning approaches have led to advances in the above areas for few-shot learning, we introduce a generalized meta-learning framework for FSD.

### C. FEW-SHOT DETECTION

Although the supervised learning detectors [5]–[12] have made significant success, tend to struggle in the few-shot regime where models must adapt quickly on new tasks with scarce data. Most related works of few-shot learning focus on image classification and regression, forming a striking contrast with FSD.

[42] implements few-example detection by using a large pool of unlabeled images under semi-supervised learning setting. Most closely related to our work is LSTD [39] which solves the FSD problem by transfer learning. There are still some differences for the two methods. Firstly, Meta-SSD is

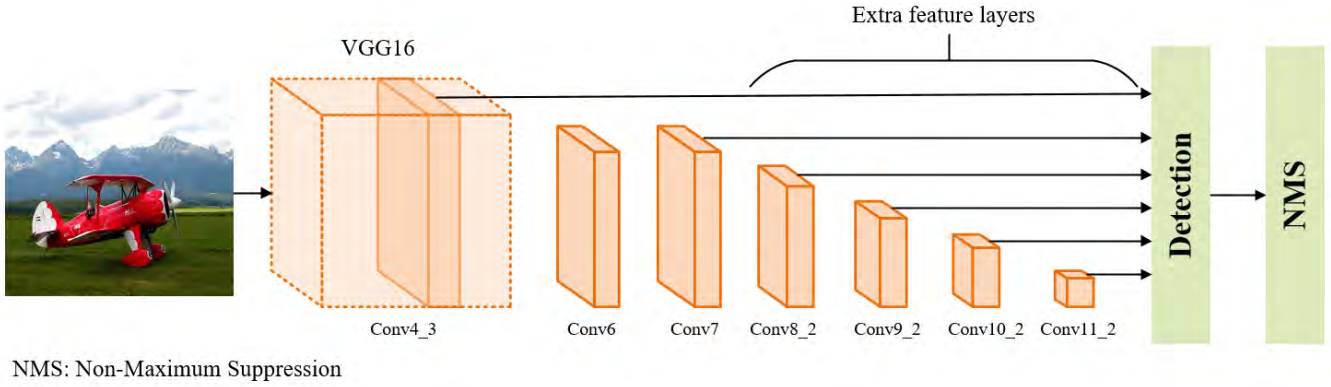


**FIGURE 3.** SSD framework. SSD employs a CNN as the base network for feature extraction, then adds the extra convolutional feature layers to the end of base network, which can make predictions at multi-scales by decreasing the feature sizes progressively.

trained and tested at the task level, LSTD is trained on a source domain and finetuned on a target domain. Secondly, Meta-SSD updates its parameters only once for each task, LSTD need to be finetuned for hundreds or thousands of iterations on the target domain. Thirdly, LSTD combines SSD with Faster R-CNN to implement a coarse-to-fine detection pipeline, Meta-SSD learns a pure SSD in the meta-learning manner without other techniques.

This work solves the FSD problem from a new perspective: proposing a generalized meta-learning detection framework to detect objects of newly interested classes, and opens a new door for the FSD problem.

## III. METHODOLOGY

This work aims to address the problem of FSD under the blessing of meta-learning. We first define the meta-learning FSD problem, spontaneously, propose a corresponding meta-leaning methodology which is implemented by integrating SSD into a meta-learning pipeline to deal with this problem.

### A. PROBLEM FORMULATION

The meta-learning detection system acquires strong prior knowledge on training tasks, then can implement a fast adaptation on new tasks with few examples. In the meta-learning FSD problem, the proposed generalized FSD framework works in a meta-learning process, composed of a meta-learner and an object detector. After trained on a series of tasks on seen classes, the meta-learner should teach the objector how to adjust its parameters rapidly (updating only once) in new tasks of unseen classes with few examples.

Different from the supervised-learning methods, the basic unit for training and evaluating meta-learning system is ''task'', shown in Fig. 2, which contains a training set and a test set. Let a meta-learning FSD detector $\phi$ can output predicted categories $y$ and locations $t$ from the given input $i$. In order to achieve fast adaptation on new tasks with few examples, this detector is trained on the distribution $p(\mathcal{T})$ of many similar tasks on the principle that the process of training and test should be consistent. We sample K examples for each task $\mathcal{T}_i$ in the K-shot setting. The meta-learning usually occurs on the batches of tasks, where $\phi$ is trained on the training set of $\mathcal{T}_i$ and feedback from the detection loss $\mathcal{L}_{\mathcal{T}_i}$, then tested on the test set of task $\mathcal{T}_i$. The meta-learner attempts to guide the detector to update its parameters $\theta$ to decrease the test error. We collect the loss $\mathcal{L}_{\mathcal{T}_i}$ on test sets from a batch of tasks as meta-info to update the meta-learner. The meta-learning process is repeated until the meta-learner can learn how to adjust the detector $\phi$ to get a satisfactory test performance. Once trained, this meta-learning FSD framework should achieve a good performance on test sets of new tasks sampled from $p(\mathcal{T})$ after learning from K examples.

### B. SSD REVISIT

#### 1) SSD

The proposed Meta-SSD utilizes the successful SSD for FSD, shown in Fig. 3. We will not enter into the details of SSD, and readers can refer to [8] for more details. SSD is a representative one-stage detection method, where a carefully designed multi-layer bounding box regression architecture can locate objects with various scales. SSD employs a CNN as the base network for feature extraction, then adds the extra convolutional feature layers to the end of base network, which can make predictions at multi-scales by decreasing the feature sizes progressively (extra feature layers in Fig. 3). These feature layers of the auxiliary structure can output the predictions of location and category by the followed convolutional layers. Besides, SSD has a set of default bounding boxes in each feature map cell for effective detection. In each feature map cell, the relative offsets to the default bounding boxes and classification scores can be predicted to implement object detection. Non-Maximum Suppression (NMS) is performed to reduce redundant detection boxes.

#### 2) WHY SSD

This work tries to address the FSD problem from the perspective of meta-learning and proposes a generalized meta-learning FSD framework. Then we need to evaluate the feasibility of the framework by equipping it with an object detection architecture. The detector should be relatively simple to reduce the difficulties of meat-learning FSD, causing

that the two-stage detectors [6], [10], [12] are excluded. For one-stage detectors, SSD is simple and effective by implementing multi-scale detection with an auxiliary convolutional structure. Others like [5], [7] have lower accuracy and [43]–[45] have more complex structures. The goal of this work is to verify the feasibility of meta-learning FSD and offer a new way for FSD, rather than pursue the accuracy of FSD. Therefore, we select SSD to implement meta-learning FSD, called Meta-SSD, and evaluate it by experiments. It should be noted that other detectors can also be combined with the proposed FSD framework, which should be explored in the future.

### C. META-LEARNING FSD

The effective methods commonly used in supervised learning (such as optimizers and learning rate strategies) with sufficient examples may not work well under the few-shot regime. The initialization of detector $\phi$ is crucial for FSD due to the limited information from few examples. Hence, acquiring a good set of initialized parameters $\theta$ as strong prior knowledge is reasonable. Furthermore, the way to update networks is also non-trivial for avoiding overfitting. This meta-learning FSD system is proposed by considering these key issues mentioned above. Inspired by the meta-learning methods [26], [38] in few-shot learning image classification, the proposed framework is designed to learn both good initializations and suitable learning rates, shown in Alg. 1.

In supervised learning object detection, models are updated by gradient descent:

$$\theta^t = \theta^{t-1} - \alpha \nabla \mathcal{L}_{\mathcal{T}}(\theta^{t-1}) \tag{1}$$

here, $\alpha$ denotes the learning rate and the loss $\mathcal{L}_{\mathcal{T}}$ usually contains cross-entropy loss $\ell_{cls}$ for classification and $smooth_{L1}$ loss $\ell_{reg}$ for bounding box regression, usually computed by:

$$\mathcal{L}_{\mathcal{T}} = \frac{1}{|\mathcal{T}|} \sum_{(i,y^*,t^*)} (\ell_{cls}(\phi_\theta(i), y^*) + \ell_{reg}(\phi_\theta(i), t^*)) \tag{2}$$

where $y^*$ and $t^*$ are the labels for classification and location regression respectively.

In the proposed generalized FSD framework, we employ a meta-learner to learn the learning rate $\alpha$, such that updating the parameters $\theta$ is just right, neither overfitting nor underfitting. Therefore, Equation. 1 can be rewritten as:

$$\theta^t = \theta^{t-1} - \alpha^* \nabla \mathcal{L}_{\mathcal{T}}(\theta^{t-1}) \tag{3}$$

The learning rate $\alpha^*$ is not predefined but can be learnt by a meta-learner from the distribution $p(\mathcal{T})$. We set a learnable learning rate $\alpha^*$ for each parameter of the detector.

With the above definition, the objective of meta-learning is:

$$\min_{\theta} \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(\phi_{\theta'}) = \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(\phi_{\theta - \alpha^* \nabla \mathcal{L}_{\mathcal{T}_i}(\phi_\theta)}) \tag{4}$$

Different from supervised-learning detection, the goal of meta-learning FSD is to adjust the detector's parameters $\theta$ to $\theta'$ for fast adaptation on new tasks. Hence, the objective should be computed on the updated parameters $\theta'$ as the above equation. The losses $\mathcal{L}_{\mathcal{T}}$ from the updated parameters $\theta'$ are collected as meta-info to update the meta-learner by stochastic gradient descent (SGD) as follows:

$$(\theta, \alpha^*) = (\theta, \alpha^*) - \beta \nabla \sum_{\mathcal{T}_i \sim p(\mathcal{T})} \mathcal{L}_{\mathcal{T}_i}(\phi_{\theta'}) \tag{5}$$

We set a meta-learning rate $\beta$ for the meta-learner so that the meta-learning process can be executed just like in supervised learning.

**Algorithm 1** Meta-Learning for FSD

**Input:** FSD task distribution $p(\mathcal{T})$, meta-learning rate $\beta$
**Ensure:** detector's parameters $\theta$, detector's learning rate $\alpha^*$

```
Initialize θ, α*
while not end do
    Sample n tasks from p(T)
    for all j = 1; j ≤ n do
        L_{T_j^train} = 1/|T_j^train| Σ_{i∈T_j^train} ℓ(φ_θ(i))
        θ' = θ − α* ∇ L_{T_{j_train}}
        L_{T_j^test} = 1/|T_j^test| Σ_{i∈T_j^test} ℓ(φ_θ'(i))
    end for;
    (θ, α*) = (θ, α*) − β ∇ Σ_{j∈(1,n)} L_{T_j^test}
end while
```

### D. IMPLEMENTATION

Meta-SSD is implemented with an end-to-end neural network. Although any supervised learning object detection models can be adopted in the proposed FSD framework theoretically, the goal of this work is trying to achieve FSD with meta-learning, not pursuing higher performance. We implement the generalized FSD framework by combing the relatively simple and effective SSD300 [8] with a meta-learning process [38].

VGG16 [2] is used as the base network. We set learnable learning rates for all parameters. Therefore, the meta-learning system can learn the good initialization as well as the fast adaptation strategy. The optimizer of meta-learning is SGD, such that we can train meta-learner in the supervised learning manner. In the training phase, SSD is adjusted with the learning rates from meta-learner, then tested on the test set. The losses from a batch of tasks are collected as the meta info to update the meta-learner. During test, we employ the same way as in training, except that the object classes are different from those in the training phase. Non-Maximum Suppression (NMS) is performed to reduce redundant detection boxes.

## IV. BENCHMARKS

### A. DATA ORGANIZATION

The whole NIST-FSD dataset can de divided into two sets: meta-train and meta-test sets (Fig. 4). Meta-train set only

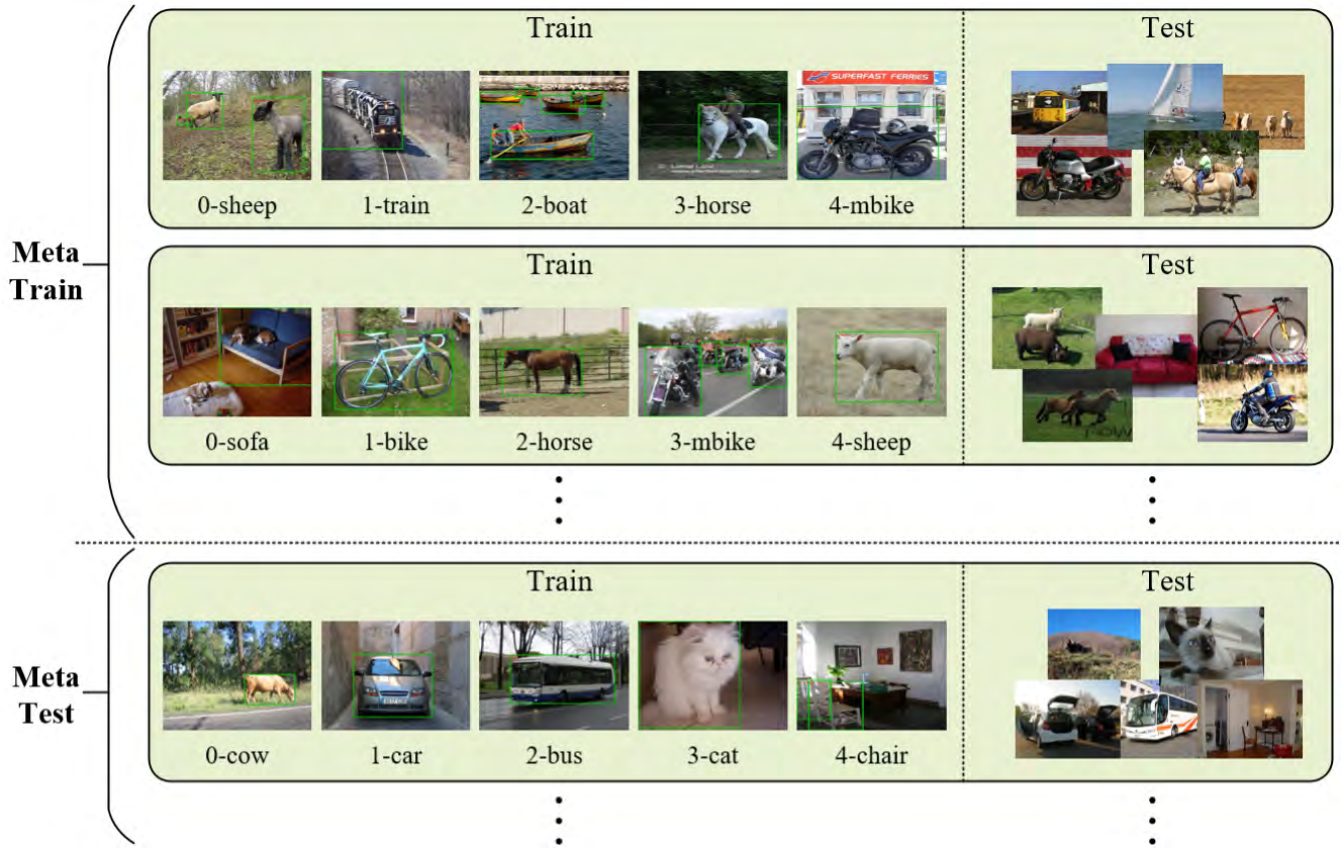


**FIGURE 4.** The 5-way 1-shot FSD setup. The whole dataset is divided into meta-train and meta-test sets. In each task (denoted by green box), there are training and test subsets sampled from meta-train or meta-test set, and they have same classes. One image per class is randomly selected for training, test subsets of meta-train and training subset of meta-test, while fifteen images per class are randomly selected for test subset of meta-test. Note that the classes of meta-train set are not present in meta-test set.

contains the objects of seen classes, while meta-test set only contains the objects of unseen classes. Furthermore, in each meta-train or meta-test set, there are two subsets: training and test subsets. The object detector is trained on the training subset, then updated under the guidance of meta-learner. The test subset is used for evaluating the updated detector and collecting the loss to update meta-learner. We employ 4 different partitions as [46] and more details about the data organization is shown in Table. 2. The images of meta-train set are from the VOC 2007 train&val and VOC 2012 train&val datasets, and the images of meta-test set are from the VOC 2007 test dataset.

### B. TASK DEFINITION

In the meta-learning setting, we define the task according to the number of detection categories and examples per class. A N-way K-shot meta-learning task is to detect N class objects with K examples per class.

In this work, we evaluate the proposed framework in 4 different settings: 3-way 1-shot, 3-way 5-shot, 5-way 1-shot, and 5-way 5-shot. For example, in a 3-way 1-shot meta-training task, we first sample 3 categories, then randomly select one example per class, resulting in 3 images for training set. Similarly, 3 images are selected from the remaining as test set. We execute the meta-test task in the same way as meta-training except that the number of test images is 15 per class. For details, refer to Table. 1.

**TABLE 1.** The settings of 4 different N-way K-shot FSD. There are N × K images for each subset except N × 15 images for test set of meta-test.

| setting | meta-train | | meta-test | |
|---|---|---|---|---|
| | train | test | train | test |
| 3-way 1-shot | 3×1=3 | 3×1=3 | 3×1=3 | 3×15=45 |
| 3-way 5-shot | 3×5=15 | 3×5=15 | 3×5=15 | 3×15=45 |
| 5-way 1-shot | 5×1=5 | 5×1=5 | 5×1=5 | 5×15=75 |
| 5-way 5-shot | 5×5=25 | 5×5=25 | 5×5=25 | 5×15=75 |

## V. EXPERIMENTS

### A. BASELINE

We compare Meta-SSD with two baselines. The first is to train SSD on seen classes with sufficient labelled examples, and finetune it on unseen classes with few examples. It is an appropriate baseline for a fair comparison due to SSD is the detector of Meta-SSD.

The second baseline is transfer-learning FSD method LSTD, which transfers knowledge from seen classes to unseen classes. But this is not a fair comparison, because that

**TABLE 2.** Detection results of different meta-learning settings. $S^1 - S^4$ denote different class partitions and K-N denotes the K-way N-shot setting. In each FSD setting, there are 500 randomly generated tasks for evaluation. Note that we just train 3-way 1-shot and 5-way 1-shot FSD due to the limitation of graphic memory. The 3-way 5-shot and 5-way 5-shot FSD are evaluated based on the trained 3-way 1-shot and 5-way 1-shot models respectively.

| | seen classes | | | | | | | | | | | | | | | | unseen classes | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $S^1$ | plant | tv | sofa | mbike | horse | boat | dog | bike | train | sheep | bottle | person | aero | table | bird | mAP(%) | cow | bus | cat | car | chair | mAP(%) |
| 3-1 | 6.3 | 29.3 | 26.5 | **40.3** | 46.6 | 17.8 | 36.2 | **30.8** | 34.9 | **34.3** | 3.4 | 12.6 | **36.3** | **29.5** | 23.9 | 27.2 | 32.3 | 29.1 | 37.9 | 25.0 | 6.2 | 26.1 |
| 3-5 | 7.1 | 25.3 | 32.2 | 40.2 | **51.9** | **23.9** | 37.1 | 30.0 | **40.6** | 30.1 | 4.5 | **15.4** | 31.9 | 29.1 | **25.3** | **28.3** | **36.2** | **31.2** | 40.4 | **29.0** | 7.3 | **28.8** |
| 5-1 | 5.9 | 31.2 | 38.5 | 35.0 | 40.2 | 15.9 | 38.7 | 27.8 | 35.5 | 33.1 | **6.7** | 12.0 | 32.8 | 25.4 | 21.6 | 26.7 | 30.5 | 27.0 | 38.8 | 23.6 | 6.5 | 25.3 |
| 5-5 | **8.1** | **32.7** | **39.4** | 35.9 | 43.7 | 18.1 | **40.1** | 27.2 | 39.6 | 33.3 | 6.6 | 13.7 | 33.1 | 26.2 | 21.1 | 27.9 | 34.7 | 27.5 | **43.4** | 25.8 | **7.4** | 27.8 |
| $S^2$ | plant | tv | sofa | bus | boat | bike | train | cow | cat | car | chair | sheep | bottle | aero | bird | mAP(%) | table | dog | horse | mbike | person | mAP(%) |
| 3-1 | 14.5 | 24.6 | **37.0** | **45.5** | **25.4** | 43.8 | **42.2** | 41.7 | 40.8 | 25.5 | **18.9** | 40.5 | 22.6 | **39.5** | 27.9 | 32.7 | 4.3 | 35.1 | 35.2 | 28.9 | 14.8 | 23.7 |
| 3-5 | **16.7** | **27.7** | 34.0 | 45.3 | 22.9 | **46.1** | 40.8 | **44.1** | 39.7 | 26.8 | 18.4 | **43.2** | **24.6** | 38.2 | 29.8 | **33.2** | 2.7 | **40.5** | **39.6** | 34.7 | **17.9** | **27.1** |
| 5-1 | 10.7 | 23.2 | 31.7 | 39.1 | 21.5 | 31.0 | 41.6 | 38.3 | 40.5 | 27.8 | 10.9 | 41.8 | 9.7 | 35.2 | 31.0 | 28.9 | 8.5 | 26.4 | 33.9 | 30.6 | 12.0 | 22.3 |
| 5-5 | 8.9 | 25.2 | 30.3 | 41.2 | 21.7 | 32.2 | 41.4 | 43.8 | **41.0** | **31.5** | 11.0 | 40.7 | 12.1 | 37.6 | **32.8** | 30.1 | **10.1** | 30.4 | 33.1 | **35.5** | 13.0 | 24.4 |
| $S^3$ | bus | mbike | horse | boat | dog | bike | cow | cat | car | chair | bottle | person | aero | table | bird | mAP(%) | plant | sheep | sofa | train | tv | mAP(%) |
| 3-1 | 34.2 | 33.4 | **46.6** | 14.3 | **39.5** | 27.6 | 29.5 | 41.7 | **29.6** | 6.7 | 2.7 | 10.4 | 27.9 | **21.9** | 20.3 | 25.8 | 4.1 | **31.8** | 19.7 | **36.6** | 6.6 | 19.8 |
| 3-5 | 37.6 | 31.3 | 46.0 | 14.2 | 39.3 | 25.4 | 34.3 | **50.0** | 28.0 | 6.8 | 3.0 | 9.2 | 28.7 | 20.1 | 19.8 | 26.2 | 4.2 | 31.0 | 21.3 | 33.2 | 7.7 | 19.5 |
| 5-1 | **39.0** | **36.5** | 40.8 | 20.1 | 33.4 | 30.3 | 31.8 | 39.3 | 23.7 | **9.3** | **7.6** | 11.1 | 29.7 | 16.3 | 19.9 | 25.9 | **6.4** | 29.6 | **24.8** | 30.1 | 11.5 | **20.5** |
| 5-5 | 38.4 | 36.1 | 43.3 | **22.7** | 36.6 | **34.1** | **34.7** | 38.3 | 23.8 | 9.2 | 7.2 | **14.9** | **30.6** | 20.2 | **20.8** | **27.4** | 5.3 | 31.1 | 23.0 | 29.7 | **11.9** | 20.2 |
| $S^4$ | bus | mbike | horse | dog | cow | cat | car | chair | person | table | plant | sheep | sofa | train | tv | mAP(%) | aero | bike | bird | boat | bottle | mAP(%) |
| 3-1 | **40.4** | 33.2 | 38.4 | 30.5 | 32.8 | 41.6 | 22.4 | 6.2 | 10.0 | 14.7 | **6.0** | 30.0 | 27.2 | 32.4 | 11.2 | 25.1 | 20.2 | **42.8** | 20.0 | 7.0 | 8.6 | 19.7 |
| 3-5 | 38.5 | 34.8 | 38.1 | 33.6 | **33.9** | **44.1** | **24.1** | 6.2 | 10.6 | **15.1** | 5.1 | 30.0 | 26.7 | 34.8 | 13.1 | 25.9 | 20.2 | 41.1 | 21.2 | 7.6 | **9.4** | 19.9 |
| 5-1 | 38.9 | 36.6 | **40.7** | 33.8 | 32.0 | 39.3 | 19.9 | 7.8 | 9.0 | 13.4 | 4.2 | 32.7 | **28.4** | 34.0 | 14.8 | 25.7 | 29.1 | 38.7 | **21.6** | 12.4 | 5.7 | 21.5 |
| 5-5 | 39.1 | **37.3** | 40.1 | **35.0** | 31.1 | 40.2 | 23.5 | **8.6** | **11.2** | 13.9 | 4.4 | **34.7** | 27.4 | **34.9** | **17.2** | **26.6** | **29.5** | 38.8 | 21.5 | **13.2** | 5.8 | **21.8** |

LSTD is a two-stage detector with higher performance by combining SSD with Faster R-CNN, while both Meta-SSD and SSD employ the one-stage detection pipeline.

### B. EXPERIMENTAL SETTING

For effectively evaluating the performance of Meta-SSD, we design 4 different class partitions with 4 few-shot settings, leading to 16 experiments shown in Table 2. Where $S^1 - S^4$ denote different class partitions and K-N denotes the K-way N-shot setting. It should be noted that we just train Meta-SSD for 3-way 1-shot and 5-way 1-shot FSD, because there is not enough graphic memory for training 3-way 5-shot and 5-way 5-shot FSD. However, we still evaluate 3-way 5-shot and 5-way 5-shot FSD based on the trained 3-way 1-shot and 5-way 1-shot models respectively. The 5-shot performance on seen and unseen classes will be better if we train Meta-SSD in the 5-shot setting.

In each experiment, we evaluate Meta-SSD on 500 meta-test sets (that is 500 random tasks) and compute the mean performance (average precision) of each category on the test set from each task. Besides, we update the parameters only once in each task for training and test.

Meta-SSD is optimized by SGD optimizer for 30000 episodes. There are 4 tasks in each episode. Learning rate $\alpha^*$ is $10^{-3}$ and $\beta$ is from $10^{-3}$ to $10^{-6}$. The threshold for NMS is 0.45 and the threshold for classification is 0.01.

### C. PERFORMANCE

The experiment results of different settings are shown in Table 2. Meta-SSD achieves a promising effect, indicating it is a feasible avenue for meta-learning FSD. During training and test, Meta-SSD can implement fast adaptation (updating only once) successfully with few examples. In each task, we randomly select new classes and shuffle them. This operation increases the difficulties of FSD, especially for object classification due to the parameters for classification on the previous task are useless for the current task. Thus, recognizing the novel classes rapidly is a challenge for Meta-SSD.

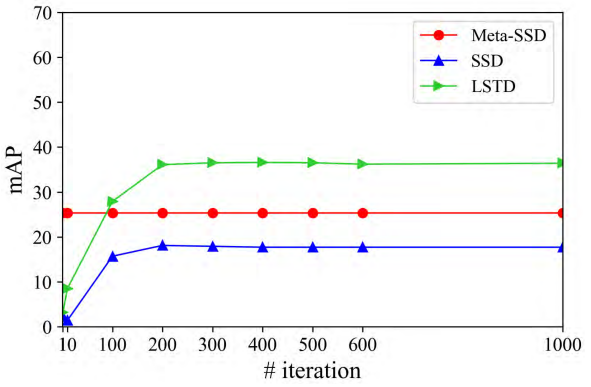


**FIGURE 5.** Comparison in different updating steps for 5-way 1-shot on unseen classes of $S^1$.

As shown in Table 2, Meta-SSD has good generalization ability and yields a similar performance on unseen classes to the seen classes. Note that the 5-shot experiments are evaluated based on the trained 1-shot models, resulting in the slight improvements except for unseen classes on $S^3$.

### D. COMPARISON WITH BASELINES

For a fair comparison, we train and evaluate Meta-SSD, SSD and LSTD in a same setting as much as possible. Based on $S^1$, we train Meta-SSD with 5-way, 1-shot and test it on unseen classes, and train LSTD, SSD on seen classes and finetune them on unseen classes. We update meta-SSD only once as [38] and finetune LSTD, SSD from 1, 10, 100, to 1000 steps on unseen classes, shown in Fig. 5.

Even though Meta-SSD updates only once, it performs better than SSD. It's difficult for a detector like SSD with a large number of parameters to converge well in one updating step,

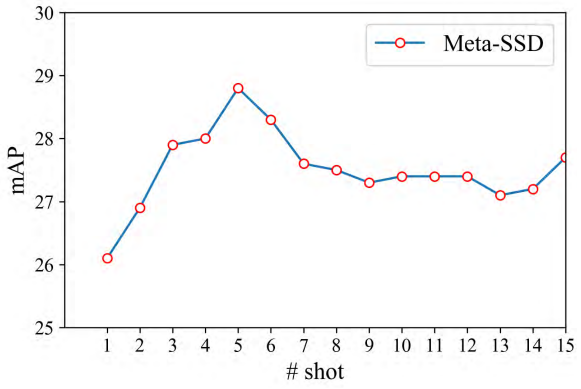


**FIGURE 6.** Performance of the trained 3-way 1-shot Meta-SSD with different number of training images per class on unseen classes of $S^1$.

but Meta-SSD can implement this fast adaptation process due to the carefully designed meta-learning pipeline.

Although LSTD employs the two-stage detection pipeline, Meta-SSD also outperforms LSTD with few iterations (less than 100). The experiment results indicate the effectiveness of Meta-SSD and meta-learning is a promising method for FSD.

### E. FEW-SHOT LEARNING

In Table 2, Meta-SSD is evaluated in the 5-shot setting. The trained 1-shot model can yield better performances after providing more examples (5-shot). We further explore the performance of Meta-SSD with different number of examples in few-shot setting (less than 15 examples per class), shown in Fig. 6. We train the 3-way 1-shot Meta-SSD on seen classes of $S^1$ and evaluate it in different K-shot settings on unseen classes. With the increase of examples, the performance can be further improved and tends to be stable. The results show that the meta-learning FSD framework is a rather challenging but promising method. Therefore, how to improve the upper bound of fast adaptation for meta-learning FSD methods should be researched in the future.

### F. LEARNING RATE ANALYSIS

We visualize the learning rates of each layer in Meta-SSD to qualitatively analyze the learnable learning rates and further understand it in principle, shown in Fig. 7. Here, we compute the arithmetic mean along the output channel and just select the learning rates from the first 64 input channels in each layer.

The learning process of the proposed Meta-SSD can be viewed as finding an appropriate sensitivity factor (learning rate) for each parameter of the detector. These parameters are the learned knowledge from the distribution $p(\mathcal{T})$ of FSD tasks. Some parameters represent the general knowledge across different tasks which are insensitive to the change of tasks, and they should be given smaller sensitivity factor (the light blue areas in Fig. 7). Whereas a few parameters are the specific knowledge corresponding to a task, which are sensitive to the change of tasks and enable the detector to adapt

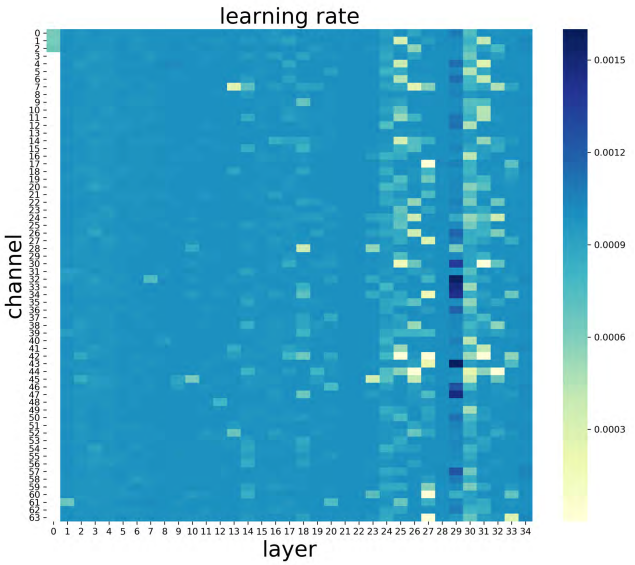


**FIGURE 7.** Learning rates of different layers. Here, we visualize the learning rates of the first 64 channels from each layer. Best viewed in color.

rapidly for a new task. These parameters should be assigned larger sensitivity factor (the dark blue areas in Fig. 7).

It is interesting to note that there are some yellowy areas which should be more insensitive to the change of tasks. It's reasonable that few parameters in the lower layers used to extract general image features should be more insensitive. However, some parameters in the higher layers are also more insensitive. These learning rates are hard to set well manually, indicating the effectiveness of the meta-learning strategy.

## VI. CONCLUSION

This work proposes a generalized FSD framework based on meta-learning and implements it by the carefully designed Meta-SSD, aiming to overcome the drawbacks of the current object detection advances when faced with the few-shot regime. Meta-SSD, consisting of a meta-learner and an object detector (SSD), can learn the general knowledge and proper fast adaptation strategies with the learnable learning rate set for each parameter. The proposed framework is trained on a distribution of similar FSD tasks, where the detector must achieve good performance for every task with few images. Then, the meta-learner can teach the detector how to adjust parameters rapidly from few examples. Besides, a benchmark NIST-FSD for FSD is built based on Pascal VOC dataset to evaluate meta-learning FSD methods. Experiments show that meta-learning is a promising approach for FSD. This work can give some suggestions to the future research and serve as a baseline for meta-learning FSD.

## REFERENCES

[1] 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,.

[2] A. Conneau, H. Schwenk, L. Barrault, and Y. Lecun, "Very deep convolutional networks for text classification," Jun. 2016, pp. 1107–1116, *arXiv:1606.01781*. [Online]. Available: https://arxiv.org/abs/1606.01781

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.
[4] 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.*, Jun. 2015, pp. 1–9.
[5] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "OverFeat: Integrated recognition, localization and detection using convolutional networks," Dec. 2013, *arXiv:1312.6229*. [Online]. Available: https://arxiv.org/abs/1312.6229
[6] 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.*, Jun. 2014, pp. 580–587.
[7] 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.*, Jun. 2016, pp. 779–788.
[8] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "SSD: Single shot multibox detector," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2016, pp. 21–37.
[9] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, to be published.
[10] R. B. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 1440–1448.
[11] P. Purkait, C. Zhao, and C. Zach, "SPP-Net: Deep absolute pose regression with synthetic views," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Dec. 2017, pp. 1–10.
[12] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 91–99.
[13] J. Yan, H. Wang, M. Yan, D. Wenhui, X. Sun, and H. Li, "IoU-adaptive deformable R-CNN: Make full use of IoU for multi-class object detection in remote sensing imagery," *Remote Sens.*, vol. 11, no. 3, p. 286, Feb. 2019.
[14] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2017.
[15] T. Pohlen, A. Hermans, M. Mathias, and B. Leibe, "Full-resolution residual networks for semantic segmentation in street scenes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 3309–3318.
[16] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Apr. 2015, pp. 3431–3440.
[17] X. Gao, X. Sun, Y. Zhang, M. Yan, G. Xu, H. Sun, J. Jiao, and K. Fu, "An end-to-end neural network for road extraction from remote sensing imagery by multiple feature pyramid network," *IEEE Access*, vol. 6, pp. 39401–39414, 2018.
[18] Z. Yan, M. Yan, H. Sun, K. Fu, J. Hong, J. Sun, Y. Zhang, and X. Sun, "Cloud and cloud shadow detection using multilevel feature fused segmentation network," *IEEE Geosci. Remote Sens. Lett.*, vol. 15, no. 10, pp. 1600–1604, Oct. 2018.
[19] W. Chong, W. Ren, K. Huang, and T. Tan, "Weakly supervised object localization with latent category learning," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2014, pp. 431–445.
[20] E. W. Teh, M. Rochan, and Y. Wang, "Attention networks for weakly supervised object localization," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, Sep. 2016, pp. 1–11.
[21] 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. (ECCV)*, Sep. 2016, pp. 350–365.
[22] H. Bilen and A. Vedaldi, "Weakly supervised deep detection networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2846–2854.
[23] L. Dong, 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.*, Jun. 2016, pp. 3512–3520.
[24] A. Diba, V. Sharma, A. Pazandeh, H. Pirsiavash, and L. V. Gool, "Weakly supervised cascaded convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5131–5139.
[25] R. G. Cinbis, J. Verbeek, and C. Schmid, "Weakly supervised object localization with multi-fold multiple instance learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 1, pp. 189–203, Jan. 2017.
[26] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, Aug. 2017, pp. 1126–1135.
[27] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2017.
[28] Y.-X. Wang and M. Hebert, "Learning to learn: Model regression networks for easy small sample learning," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2016, pp. 616–634.
[29] T. Munkhdalai and H. Yu, "Meta networks," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, Aug. 2017, pp. 2554–2563.
[30] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, Jun. 2016, pp. 3630–3638.
[31] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," ibn *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 1199–1208.
[32] M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J. B. Tenenbaum, H. Larochelle, and R. S. Zemel, "Meta-learning for semi-supervised few-shot classification," in *Proc. Int. Conf. Learn. Represent. (ICLR)*, 2018, pp. 1–5.
[33] A. A. Rusu, D. Rao, J. Sygnowski, O. Vinyals, R. Pascanu, S. Osindero, and R. Hadsell, "Meta-learning with latent embedding optimization," Jul. 2018, *arXiv:1807.05960*. [Online]. Available: https://arxiv.org/abs/1807.05960
[34] A. Nichol, J. Achiam, and J. Schulman, "On first-order meta-learning algorithms," 2018, *arXiv:1803.02999*. [Online]. Available: https://arxiv.org/abs/1803.02999
[35] T. Munkhdalai and A. Trischler, "Metalearning with hebbian fast weights," 2018, *arXiv:1807.05076*. [Online]. Available: https://arxiv.org/abs/1807.05076
[36] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2017, pp. 4077–4087.
[37] W. Yong, X.-M. Wu, Q. Li, J. Gu, W. Xiang, L. Zhang, and V. O. K. Li, "Large margin few-shot learning," 2018, *arXiv:1807.02872*. [Online]. Available: https://arxiv.org/abs/1807.02872
[38] Z. Li, F. Zhou, F. Chen, and H. Li, "Meta-SGD: Learning to learn quickly for few-shot learning," 2017, *arXiv: 1707.09835*. [Online]. Available: https://arxiv.org/abs/1707.09835
[39] H. Chen, Y. Wang, G. Wang, and Y. Qiao, "LSTD: A lowshot transfer detector for object detection," 2018, *arXiv:1803.01529*. [Online]. Available: https://arxiv.org/abs/1803.01529
[40] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, Sep. 2009.
[41] S. Thrun and L. Pratt, *Learning to Learn*. Springer, 2012.
[42] X. Dong, L. Zheng, F. Ma, Y. Yang, and D. Meng, "Few-example object detection with model communication," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 7, pp. 1641–1654, Jul. 2019.
[43] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 6517–6525.
[44] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," Apr. 2018, *arXiv:1804.02767*. [Online]. Available: https://arxiv.org/abs/1804.02767
[45] J. Jeong, H. Park, and N. Kwak, "Enhancement of SSD by concatenating feature maps for object detection," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, May 2017, pp. 1–12.
[46] A. Shaban, S. Bansal, Z. Liu, I. A. Essa, and B. Boots, "One-shot learning for semantic segmentation," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, Sep. 2017, pp. 1–13.

**KUN FU** received the B.Sc., M.Sc., and Ph.D. degrees from the National University of Defense Technology, Changsha, China, in 1995, 1999, and 2002, respectively. He is currently a Professor with the Institute of Electronics, Chinese Academy of Sciences, Beijing, China. His research interests include computer vision, remote sensing image understanding, geospatial data mining, and visualization.

**TENGFEI ZHANG** (S'19) received the B.Sc. degree from the Ocean University of China, Qing-Dao, China, in 2016. He is currently pursuing the Ph.D. degree with the Institute of Electronics, Chinese Academy of Sciences, Beijing, China. His research interests include computer vision, pattern recognition, and remote sensing image processing, especially on object detection

**YUE ZHANG** (M'18) received the B.E. degree in electronic engineering from Northwestern Polytechnical University, Xi'an, China, in 2012, and the Ph.D. degree from the University of Chinese Academy of Sciences, Beijing, China, in 2017, where he is currently an Assistant Professor with the Institute of Electronics. His research interest includes the analysis of optical and synthetic aperture radar remote sensing images.

**MENGLONG YAN** received the B.Sc. degree from Wuhan University, Wuhan, China, in 2007, and the M.Sc. and Ph.D. degrees from Peking University, Beijing, China, in 2012. He is currently an Associate Professor with the Institute of Electronics, Chinese Academy of Sciences, Beijing. His research interests include LiDAR data processing and high-resolution remote sensing image processing.

**ZHONGHAN CHANG** received the B.Sc. degree from Tianjin University, Tianjin, China, in 2016. He is currently pursuing the Ph.D. degree with the Institute of Electronics, Chinese Academy of Sciences, Beijing, China. His research interests include computer vision, pattern recognition, and remote sensing image processing, especially on object detection.

**ZHENGYUAN ZHANG** received the B.Sc. degree from Harbin Engineering University, Harbin, China, in 2016. He is currently pursuing the Ph.D. degree with the Institute of Electronics, Chinese Academy of Sciences, Beijing, China. His research interests include computer vision, pattern recognition, and remote sensing image processing, especially on image caption.

**XIAN SUN** received the B.Sc. degree from Beihang University, Beijing, China, in 2004, and the M.Sc. and Ph.D. degrees from the Institute of Electronics, Chinese Academy of Sciences, Beijing, in 2006 and 2009, respectively, where he is currently a Professor. His research interests include computer vision and remote-sensing image understanding.