Identification of Novel Classes for Improving Few-Shot Object Detection

Zeyu Shangguan  
University of Southern California  
zshanggu@alumni.usc.edu

Mohammad Rostami  
University of Southern California  
rostamim@usc.edu

Abstract

Conventional training of deep neural networks requires a large number of the annotated image which is a laborious and time-consuming task, particularly for rare objects. Few-shot object detection (FSOD) methods offer a remedy by realizing robust object detection using only a few training samples per class [16, 37, 36, 8, 48, 15]. A challenge for FSOD is that instances from unlabeled novel classes that do not belong to the fixed set of training classes appear in the background. These objects behave similarly to label noise, leading to FSOD performance degradation. We develop a semi-supervised algorithm to detect and then utilize these unlabeled novel objects as positive samples during training to improve FSOD performance. Specifically, we propose a hierarchical ternary classification region proposal network (HTRPN) to localize the potential unlabeled novel objects and assign them new objectness labels. Our improved hierarchical sampling strategy for the region proposal network (RPN) also boosts the perception ability of the object detection model for large objects. Our experimental results indicate that our method is effective and outperforms the existing state-of-the-art (SOTA) FSOD methods https://github.com/zshanggu/HTRPN

1. Introduction

The adoption of deep neural network architectures in object detection has led to a significant method in determining the location and the category of objects of interest in an image. In the presence of abundant training data, object detection models based on the region-based convolution neural networks (R-CNN) architecture reach high accuracy on most object detection tasks. However, preparing large-scale annotated training data can be a challenging task in some applications [35]. In the presence of insufficient training data, these models easily overfit and fail to generalize well. In contrast, humans are able a novel object class very fast based on a few samples [21, 38]. As a result, it is extremely desirable to develop models that can learn object classes using only a few samples, known as few-shot object detection (FSOD).

Current FSOD methods are based on pre-training a suitable model on a set of base classes with abundant training data and then fine-tuning the model on both the base classes and the novel classes for which only a few samples are accessible (see Figure 1). The primary approach in FSOD is to benefit from ideas in transfer learning or meta-learning to learn novel classes through the knowledge obtained during the pre-training stage while maintaining good performance in base classes. Despite recent advances in FSOD, current SOTA methods are still far from getting favorable results on novel classes similar to the base classes. Potential reasons for this performance gap include the confusion between visually similar categories, incorrect annotations (label noise), the existence of unseen novel objects during training, etc. Recent FSOD methods have focused on addressing these challenges for improved FSOD performance.

We study the phenomenon that unlabeled novel object classes that do not belong to either of the base or the labeled novel classes can appear in the training data. For example, we see in Figure 1 that among base-class training samples, there are a number of objects that remain unlabeled, such as the cow in the image. These unlabeled objects can potentially belong to unseen novel classes. Our experiments demonstrate that this phenomenon exists in PASCAL VOC [4] and COCO [25] datasets. This phenomenon leads to the objectness inconsistency for the model when...
recognizing the novel objects: for the novel class, objects are treated as background if their annotations are missing, but they are treated as foreground where they are labeled. Such nonconformity of foreground and background confuses the model when training the objectness and make the model hard to converge and degrades detection accuracy.

To tackle the above challenge, we develop a semi-supervised learning method to utilize the potential novel objects that appear during training to improve the ability of the model to recognize novel classes. We first demonstrate the possibility of detecting these unlabeled objects. Our experiment indicates that some unlabeled class objects are likely to be recognized if they are similar to the training base and novel classes. We collect the unlabeled novel objects from the background proposals by determining whether they are predicted as known classes, and then we give these proposals an extra objectness label in the region proposal network (RPN) so that the model could learn them. We also analyze the defect of the standard RPN in detecting objects of different sizes during training and propose a more balanced RPN sampling method so that objects are treated equally in all scales. We provide extensive experimental results to demonstrate the effectiveness of our method on the PASCAL VOC and COCO datasets. Our contributions include:

- We modify the anchor sampling strategy so that the anchors are evenly chosen from different layers of the feature pyramid in the R-CNN architecture.
- We design a ternary objectness classification in the RPN layer which enables the model to recognize potential novel class objects to improve consistency.
- We use contrastive learning in the RPN layer to distinguish between the positive and the negative anchors.

2. Related works

We assume that there are three classes: base classes, seen novel classes, and unseen novel classes. The base and the seen novel classes form the training dataset. For base classes, we have sufficient data but for seen novel classes, we have a few samples per class. Most works in FSOD only consider these classes. The unseen novel classes are not included in the training data but emerge as novel classes in the background.

**Few-shot object detection** Typical object detection networks are usually either two-stage or one-stage. For two-stage object detection networks, such as R-CNN [10], R-FCN [3], Fast-RCNN [9], and Faster-RCNN [33], the model first applies fixed anchors in the region proposal network (RPN) to determine if a proposal box contains an object. The selected proposals then are sent to the region of interest (RoI) pooling layer to get an instance-level classification and bounding boxes. One-stage object detection networks such as SSD [26], YOLO series [31, 32], and Overfeat [39], estimate the category and the location of an object directly from the backbone network without RPN. Two-stage object detection networks have higher detecting accuracy than one-stage schemes but lower inference speed [20]. Few-shot object detection (FSOD), is a case of object detection that only a few samples are available for training.

**Two-stage FSOD** For FSOD, the model is usually first pre-trained on the base classes for which we have data-sufficient. The model is then, fine-tuned on the seen novel classes [44], each with a few samples. As for the fine-tuning stage, meta-learning and transfer learning are two major end-to-end approaches. Methods based on meta-learning [5, 11, 12, 51] build an inquiry set and a support set that contains k categories with n samples in each, namely k-way n-shot setting. By creating the k-way n-shot episodes for training, meta-learning help to learn a metric to determine which support set category an inquiry image belongs to [5, 11, 12, 51]. In contrast, methods based on transfer learning start from the pre-training weights and fine-tune the model on the novel seen classes [21, 44, 40].

**Unseen novel objects** In an object detection problem, the set of the base and the seen novel classes are assumed to be a closed set. However, there may be potential novel unseen objects in the training data that do not belong to the initial set of classes, particularly when it comes to infrequent classes [30]. These objects naturally are classified as one of seen classes and hence, there has been an interest to mitigate the adverse effects of these objects. Semi-supervised object detection network is a potential solution for this problem which utilizes the challenging samples [34, 28, 50]. Kaul et al. [18] build a class-specific self-supervised label verification model to identify candidates of unlabeled (unseen) objects and give them pseudo-annotations. The model is then retrained with these pseudo-annotated samples to improve the object-detecting accuracy. However, this method requires two rounds of training and requires extra effort to adapt to other categories. Li et al. [24] propose a distractor utilization loss by giving the distractor proposals a pseudo-label during fine-tuning. This method is used only in the fine-tuning stage and hence, the objectness inconsistency from the pre-training stage is not addressed. Inspired by these shortcomings, we propose utilizing the unlabeled potential objects that belong to the unseen classes to reduce the negative effect of novel objects.

**Contrastive learning** can be used to enlarge the inter-class distances and narrow down the intra-class distances for classification tasks to enhance data representations. It has been applied to many classification tasks in topics such as visual recognition [29, 42, 14], semantic segmentation [43], super-resolution [45], and natural language processing [7, 2]. Self-supervised contrastive learning in few-shot object detection is introduced by FSCE [40] to better dis-
and with its ground truth are sent to the region of bbox feature map, 3 fixed anchors will be applied on are obj with ground truth objectness 0 = 0 iou are called negative anchors (and Prop). On the contrary, they would be iou A m iou × < Prop Prop = 1 0 (b) ∩ = C = 1 obj iou indicates a non-object and is the predicted ob- (d) represents the intersection over the union of a proposal box (i.e., Prop{bbox_c, iou^gt_p}). Anchors with iou^gt_a > 0.7 are called active anchors (A_n) and their corresponding proposals are called positive proposals (Prop_p); while anchors with iou^gt_a < 0.3 are called negative anchors (A_n) and the corresponding proposals are called negative proposals (Prop_n). Next, Prop_p and Prop_n are sent to the region of interest pooling layer (RoI pooling) to predict their instance level classification (i.e., cls^i, where i is the classification index) and the refined bounding box (i.e., bbox_r). The objects (Obj{cls^i, bbox_r}) are then finally detected.

The challenge that we want to address stems from the fact that an instance from the unlabeled and unseen novel object (C_N^m) can appear in the training dataset in the background (see Figure 2). The reason is that there are many potential classes that we have not included in either the base classes or the unseen novel classes. When detected, these objects would be treated as Prop_n and with its ground truth objectness obj^gt_pn = 0. On the contrary, they would be treated as Prop_p with ground truth objectness obj^gt_pn = 1 if it is labeled as such by the model. These instances can significantly confuse the model when adapting the model for learning the novel unseen classes. We argue that if the unlabeled potential novel object can be distinguished from the Prop_n, then its objectness could be modified as a foreground object. Consequently, the inconsistency of the objectness would be eliminated. In other words, we propose

3. Problem Description

We formulate the problem of FSOD following a standard setting in the literature [16, 44]. We use the Faster R-CNN network as the object detection model and follow the same evaluation paradigm defined by [44]. Accordingly, the base classes are those classes for which we have sufficient images and instances for each base class (C_B), while novel classes are those for which, we only have a few training samples in the dataset (C_N), where C_B ∩ C_N = ∅. An n-shot learning scenario means that we have access to n instances per seen novel categories. During the pre-training stage, the model is trained only on base class C_B, and also is only evaluated on the test set of C_B. We then proceed to learn the seen novel classes in the second stage. To overcome catastrophic forgetting about the learned knowledge about the base classes, the pre-trained model is then fine-tuned on both the seen novel classes and base classes C_N ∪ C_B and then is tested on both sets of classes.

Architectures based on R-CNN have been used consistently for object detection. In our work, we improve the R-CNN architecture to identify novel unseen classes as instances that do not belong to the seen classes. For an input image, R-CNN derives five scaled feature maps (p2 ∼ p6) using its feature pyramid network (FPN), and then size-fixed anchors in the region proposal network (RPN) are applied on these feature maps to predict the objectness (i.e., obj{obj^pre, obj^gt, iou^gt}), where obj^pre is the predicted objectness score in the range of 0 and 1. Here, the ground truth value obj^gt = 0 indicates a non-object and obj^gt = 1 represents a true object, and iou^gt represents the intersection over the union of an anchor with its ground truth box) and the coarse bounding box (i.e., bbox_c) of each anchor to get proposal boxes (i.e., Prop{bbox, iou^gt}). iou^gt represents the intersection over the union of a proposal box with its ground truth box). Anchors with iou^gt_a > 0.7 are called active anchors (A_n) and their corresponding proposals are called positive proposals (Prop_p); while anchors with iou^gt_a < 0.3 are called negative anchors (A_n) and the corresponding proposals are called negative proposals (Prop_n). Next, Prop_p and Prop_n are sent to the region of interest pooling layer (RoI pooling) to predict their instance level classification (i.e., cls^i, where i is the classification index) and the refined bounding box (i.e., bbox_r). The objects (Obj{cls^i, bbox_r}) are then finally detected.

The challenge that we want to address stems from the fact that an instance from the unlabeled and unseen novel object (C_N^m) can appear in the training dataset in the background (see Figure 2). The reason is that there are many potential classes that we have not included in either the base classes or the unseen novel classes. When detected, these objects would be treated as Prop_n and with its ground truth objectness obj^gt_pn = 0. On the contrary, they would be treated as Prop_p with ground truth objectness obj^gt_pn = 1 if it is labeled as such by the model. These instances can significantly confuse the model when adapting the model for learning the novel unseen classes. We argue that if the unlabeled potential novel object can be distinguished from the Prop_n, then its objectness could be modified as a foreground object. Consequently, the inconsistency of the objectness would be eliminated. In other words, we propose

Figure 2: The left images in each row indicate the predicted boxes with the class names and confidence, and the right images are the feature maps of the RPN layer.

Figure 3: Novel unseen classes versus novel seen classes: (a) a schematic diagram of how coarse anchors work. For an n × m feature map, 3 fixed anchors will be applied on each pixel of it. Therefore, each feature map would have 3m × n coarse anchors. (b) The mechanism of our proposed HSamp can equally choose anchors for each layer.
to reduce an effect similar to noisy labels as these objects would be objects with wrong labels, leading to confusion in the model and performance degradation.

4. Proposed solution

We outline our solution in this section. We first demonstrate that it is possible to encounter instances of novel unseen classes during the training stage. We then investigate the relationship between the number of anchors and the size of objects in each feature layer and provide a more effective sampling method for object detection. Finally, we describe our proposed pipeline to pick up objects from unseen classes with high confidence and then explain how we can modify their objectness loss to reduce their adverse effect.

4.1. Finding the Potential Proposals

According to the original Faster R-CNN network [33], the unlabeled area in an image would be wrongly treated as background objects in the RPN layer during training. Therefore, the potential unlabeled objects from unseen novel classes are suppressed and hard to be identified as true objects. However, to correct the objectness of these potential objects, the first step is how to find them. We have observed the fact that the network often has clear attention to the potential objects in the RPN layer. As an example, we have used Grad-CAM visualization of the feature map of the RPN layer on some representative training images, as shown in Figure. 2. We observe that although unlabelled novel objects appear in the base training images, the RPN layer could still have strong attention to them, and consequently predict some of them as a known class. In Figure. 2a and 2b, the feature map of p3 layer clearly shows the attention of the “chair” (base class) and “sofa” (novel class), but the sofa is predicted to be an instance of the base class “car”. Similarly, in Figure. 2c and 2d, the potential novel objects (“bus”) can also be seen in the feature map of p4 layer, and the “bus” is predicted as an instance of the base class “train”. This observation serves as an inspiration to identify potential proposals: some novel objects have high possibilities to be predicted as known base class.

Potential novel unseen class objects that appear in the base class training images usually have lower $iou_{gt}$ [24] and therefore must be contained by negative anchors ($A_n$). Theoretically, there are always exists anchors that can include the potential novel objects in an image. According to the architecture of the RPN layer, anchor boxes are used to determine whether an area contains objects, and each pixel of the feature map will have 3 fixed anchors in different sizes and aspect ratios, as shown in Figure. 3a. Consequently, the overall number of anchors decreases for higher feature maps. In the original RPN layer, different sizes of anchors are applied according to the size of the p2 to p6 feature maps. Large anchors are more suitable for detecting large objects in high feature layers due to having a larger receptive field, and vice versa. Based on this inch-by-inch sliding window like search, there should exist a sufficient number of candidates $A_n$ such that they contain potential unseen novel objects. However, to improve the training speed, not all of the anchors are used for determining proposal boxes. For an image, only 256 $A_a$ and $A_n$ anchors among all feature maps are randomly chosen to participate in the RoI pooling. Nonetheless, this random selection process dramatically reduces the chance to get desired negative anchors for large objects in higher feature maps in terms of probability, since the anchors of large size in p4 to p6 layers intrinsically have fewer cardinal numbers.

To identify instances of novel unseen classes, we randomly select $A_n$ in a hierarchically balanced way, namely hierarchically sampling (HSamp). That is, if we need to pick up $m$ negative anchors ($m<256$), we equally assign...
them to each feature layer so that each layer will have around $m/5$ anchors. This way, the anchors in each feature layer would share the same possibility for being selected, as shown in Figure. 3b. Therefore, the anchors that belong to the $p4$ to $p6$ layers are safely preserved. For example, in Figure. 3b, there are $[120,000], [30,000], [7,500], [1,875], [507]$ anchors for $p4$ to $p6$ layers in a training batch, and 218 negative anchors are needed. With the original RPN, these 218 $A_n$ anchors are randomly selected which means the number of $A_n$ for $p6$ feature map is only 3. In contrast, when HSamp is used, the number of $A_n$ is equal for each feature map. We have also visualized the effect of our method in Figure 4. There is hardly any $A_n$ that contains the motorbike (novel unseen object) when using the original RPN, as shown in Figure. 4a to 4f. However, when HSamp is used, the chance to have an $A_n$ that contains the motorbike is higher, as shown in Figure. 4g to 4l. We conclude that it is crucial to implement a balanced strategy in sampling negative anchors among all feature layers in order to find potential objects that belong to unseen novel classes. The approach will help us to isolate unseen novel class instances as instances that are not similar to the seen classes.

4.2. Hierarchical ternary classification region proposal network (HTRPN)

As mentioned in Section 4.1, faster R-CNN has the ability to recognize a number of potential novel objects that belong to unseen classes, despite the fact that they are unlabelled during training on base class images. We hypothesize that this ability is because of the feature similarity between the features of some novel unseen classes and the base classes. As a result, the model would predict a novel unseen class object as a base class based on its resemblance. In other words, the novel objects contained by the negative anchors could probably have a relatively high classification score towards a base class that resembles them the most. We mark the negative anchors that contain potential novel unseen class objects as potential anchors ($A_p$), while others...
During the pre-training stage since the model only learns to identify the base classes in this stage. However, in the fine-tuning stage, the $tobj^1$ and $tobj^2$ are both considered for ranking the proposals, because the objectness of some labeled objects might be predicted as $tobj^2$ due to knowledge transfer from the pre-training stage. This step is crucial to realize the objectness consistency because the combination of $tobj^1$ and $tobj^2$ could represent the highly confident proposal and especially improve the possibility of determining positive anchors while inferring.

As shown in Figure 6, if the top two proposals out of the five proposals are ranked only using $tobj^1$, then the proposal $\odot$ would be ignored. However, when the top two proposals are ranked by $tobj^1 \oplus tobj^2$, the proposal $\odot$ could be correctly included. Such a scheme significantly increases the possibility to dig up the true object as much as possible. As a result, the anchors that contain potential novel unseen class objects are well distinguished from the coarse negative anchors. The ternary RPN will let the model maintain its sensitivity to identify new objects from classes that have never been seen before. In practice, not all potential objects that exist in training datasets are going to be singled out during training and only a subset of them could be found. However, these identified novel unseen class objects still can alleviate the confusion of the model during few-shot learning due to their dissimilarity to the seen classes.

### 4.3. Contrastive learning on objectness

To further increase the inter-class distances between $A_n$, $A'_n$, and $A_p$ subsets in HTRPN, we also include an objectness contrastive learning head ($\text{ConsObj}$) in our architecture. Inspired by the existing literature [40, 19], the cropped features of proposals are sent into an encoder with their ground truth objectness logits to perform contrastive learning. The features of proposals are encoded as a default 128-dimension feature vector, and then the cosine similarity scores are measured between every two proposals. In this way, the HTRPN would give a higher objectness score.

### 4.4. Training Loss

The global total loss is composed of the classification loss ($\mathcal{L}_{\text{Cls}}$), the bounding box regression loss ($\mathcal{L}_{\text{Bbox}}$), our ternary objectness loss ($\mathcal{L}_{\text{Tobj}}$), and the RoI feature contrastive loss $\mathcal{L}_{\text{Contra}}$, as described in Equation 1. The $\mathcal{L}_{\text{Contra}}$ is computed using the contrastive head as described in FSCE [40]. We set $\alpha = 0.5$ to be the fixed weight for balancing the contrastive learning loss.

$$\mathcal{L} = \mathcal{L}_{\text{Cls}} + \mathcal{L}_{\text{Bbox}} + \mathcal{L}_{\text{Tobj}} + \alpha \mathcal{L}_{\text{Contra}}$$ (1)
Our proposed ternary objectness loss $L_{Tobj}$ in Equation. 2 is a sum of the cross entropy objectness loss ($L_{Obj}$) and ternary RPN feature contrastive learning loss ($L_{Tcon}$). Similar to $\alpha$ in Equation. 1, $\lambda$ is a balancing factor that is set to be equal 0.5 in our experiments.

$$L_{Tobj} = L_{Obj} + \lambda L_{Tcon}$$ (2)

The ternary RPN contrastive learning loss $L_{Tcon}$ is defined as an arithmetic mean of the weighted supervised contrastive learning loss $L_{zi}$ as the following:

$$L_{Tcon} = \frac{1}{N_{Prop}} \sum_{i=1}^{N_{Prop}} w(iou_{gt}) \cdot L_{zi},$$ (3)

where $N_{Prop}$ represents the number of RPN proposals. Weights $w(iou_{gt})$ are assigned by the function $g(*)$:

$$w(iou_{gt}) = I\{iou_{gt} \geq \phi\} \cdot g(iou_{gt}),$$ (4)

where $g(*) = 1$ is a good hard-clip [40] and $I\{\ast\}$ is a cut-off function that is 1 when $iou_{gt} \geq \phi$, otherwise is 0.

$L_{zi}$ in the RPN proposal contrastive learning loss is given as:

$$L_{zi} = \frac{-1}{N_{obj_{gt}} - 1} \sum_{j=1,j \neq i}^{N_{Prop}} I\{obj_{gt}^{i} = obj_{gt}^{j}\} \cdot \log \frac{e^{z_{i} \cdot z_{j} / \tau}}{\sum_{k=1}^{N_{Prop}} I_{k \neq i} \cdot e^{z_{i} \cdot z_{k} / \tau}},$$ (5)

where $z_i$ denotes the contrastive feature, $obj_{gt}^{i}$ denotes the ground truth ternary objectness label for the $i$-th proposal, $z_{i}$ denotes normalized features while measuring the cosine distances, and $N_{obj_{gt}}$ denotes the number of proposals with the same objectness label as $obj_{gt}^{i}$.

5. Experimental Results

We empirically demonstrate that our proposed architecture and training procedure improve the FSOD performance.

5.1. Experimental Setup

We use the Faster R-CNN as our object detection model and use ResNet-101 as the backbone along with the feature pyramid network (FPN). The evaluation scheme strictly follows the same paradigm as described in TFA [44]. The mAP50 evaluation results are separately calculated on the base classes (bAP50) and the novel seen class (nAP50). We report our results on the PASCAL VOC and COCO datasets. The contrastive learning head in the fine-tuning stage is computed similarly to FSCE [40]. We used four GPUs for training. The optimizer is fixed as SGD and the weight decay is 1e-4 with momentum as 0.9. We set our batch size equal to 16 for all experiments. The $Thr_{cls}$ is fixed as 0.75. These hyperparameters are not fine-tuned. In the pre-training stage, the top 1000 proposals used for RoI pooling, are ranked by the second objectness logit (is an object). While in the fine-tuning stage, the top 1000 proposals are ranked by the maximum of the second and the third objectness logits (potential object). There are many existing FSOD methods. We compare our performance against a subset of recently developed SOTA FSOD methods.

5.2. Results on PASCAL VOC

For the PASCAL VOC 2007 and 2012 dataset, 15 categories are chosen as the base classes for pre-training, and the remaining 5 categories served as the novel classes. We follow the 3 different categories splits defined in TFA [44]. To achieve a fairer comparison, TFA [44] defined three kinds of combinations of base classes and novel classes, namely split1, split2, and split3. In each split, we evaluate the average precision for novel classes (nAP) on 1,2,3,5,10 shots separately. The training iterations are 8000 for each training epoch. We set the initial learning rate to 0.02. Our results on the three categories splits are reported in Table. 1. We observe that our proposed method improves the performance in most cases. Especially, our method is more effective when the $n$-shot is smaller.

5.3. Results on COCO

For the COCO dataset, 60 categories are selected as base classes, and the remaining 20 categories are served as novel classes. The training iterations are set to 20000 during the training stage with an initial learning rate of 0.01. AP for novel classes is evaluated upon $n = 10$ and $n = 30$ shots separately. Our experiment results for COCO are shown in Table. 2. We again observe that our method outperforms the previous works in all cases and in some cases the margin of improvement is significant. These experiments demonstrate that our method is effective.

5.4. Ablation Study

Firstly, we discuss the effectiveness of our proposed modules separately, including the Hierarchical sampling of the RPN, the ternary objectness classification, and the contrastive head of the objectness. We implemented the ablation study experiment on PASCAL VOC 5-shot scenario. Each proposed module is added to the original network in an accumulated manner. The results are presented in Table. 3. We observe that all our proposed modules are necessary for optimal performance. By adding the HSamp, we can see that a balanced sampling in RPN is necessary, as it provides comprehensive improvement of bAP and nAP during the pre-training and the fine-tuning stages. We can also observe the results of adding the ternary objectness module indicate that our method will further improve the nAP and do not significant harm to the bAP. While the
Table 1: The novel classes of nAP50fp for the PASCAL VOC dataset are evaluated on three different category splits with 1 to 10-shot scenarios. The highest score of each few-shot setting is in red color, and the second highest score is in blue color.

<table>
<thead>
<tr>
<th>Method</th>
<th>Shot</th>
<th>Backbone</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>5</th>
<th>10</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>5</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>LSTD</td>
<td>AAAI 18 [1]</td>
<td>VGG-16</td>
<td>8.2</td>
<td>1.0</td>
<td>12.4</td>
<td>29.1</td>
<td>38.5</td>
<td>11.4</td>
<td>3.8</td>
<td>5.0</td>
<td>15.7</td>
<td>31.0</td>
</tr>
<tr>
<td>YOLOv2-ft</td>
<td>ICCV19 [46]</td>
<td></td>
<td>6.6</td>
<td>10.7</td>
<td>12.5</td>
<td>24.8</td>
<td>38.6</td>
<td>12.5</td>
<td>4.2</td>
<td>11.6</td>
<td>16.1</td>
<td>33.9</td>
</tr>
<tr>
<td>RepMet</td>
<td>CVPR 19 [17]</td>
<td>InceptionV3</td>
<td>26.1</td>
<td>32.9</td>
<td>34.4</td>
<td>38.6</td>
<td>41.3</td>
<td>17.2</td>
<td>22.1</td>
<td>23.4</td>
<td>28.3</td>
<td>35.8</td>
</tr>
<tr>
<td>FRCN-ft</td>
<td>ICCV19 [46]</td>
<td></td>
<td>13.8</td>
<td>19.6</td>
<td>32.8</td>
<td>41.5</td>
<td>45.6</td>
<td>7.9</td>
<td>15.3</td>
<td>26.2</td>
<td>31.6</td>
<td>39.1</td>
</tr>
<tr>
<td>FRCN+FPN-ft</td>
<td>ICML 20 [44]</td>
<td></td>
<td>8.2</td>
<td>20.3</td>
<td>29.0</td>
<td>40.1</td>
<td>45.5</td>
<td>13.4</td>
<td>20.6</td>
<td>28.6</td>
<td>32.4</td>
<td>38.8</td>
</tr>
<tr>
<td>TFA w/ fc</td>
<td>ICML 20 [44]</td>
<td></td>
<td>36.8</td>
<td>29.1</td>
<td>43.6</td>
<td>55.7</td>
<td>57.0</td>
<td>18.2</td>
<td>29.0</td>
<td>33.4</td>
<td>35.5</td>
<td>39.0</td>
</tr>
<tr>
<td>TFA w/ cos</td>
<td>ICML 20 [44]</td>
<td></td>
<td>39.8</td>
<td>36.1</td>
<td>44.7</td>
<td>55.7</td>
<td>56.0</td>
<td>23.5</td>
<td>26.9</td>
<td>34.1</td>
<td>35.1</td>
<td>39.1</td>
</tr>
<tr>
<td>MPSR</td>
<td>ECCV 20 [49]</td>
<td></td>
<td>41.7</td>
<td>-</td>
<td>51.4</td>
<td>55.2</td>
<td>61.8</td>
<td>24.4</td>
<td>-</td>
<td>39.2</td>
<td>39.9</td>
<td>47.8</td>
</tr>
<tr>
<td>FSCE</td>
<td>CVPR 21 [40]</td>
<td></td>
<td>42.4</td>
<td>45.8</td>
<td>45.9</td>
<td>53.7</td>
<td>56.1</td>
<td>21.7</td>
<td>27.8</td>
<td>35.2</td>
<td>37.0</td>
<td>40.3</td>
</tr>
<tr>
<td>TIP</td>
<td>CVPR 21 [22]</td>
<td></td>
<td>42.4</td>
<td>43.8</td>
<td>51.4</td>
<td>61.9</td>
<td>63.4</td>
<td>27.3</td>
<td>29.5</td>
<td>43.5</td>
<td>44.2</td>
<td>50.2</td>
</tr>
<tr>
<td>DC-Net</td>
<td>CVPR 21 [13]</td>
<td></td>
<td>33.9</td>
<td>37.4</td>
<td>43.7</td>
<td>51.1</td>
<td>59.6</td>
<td>22.7</td>
<td>30.1</td>
<td>33.8</td>
<td>40.9</td>
<td>46.9</td>
</tr>
<tr>
<td>FSOD-UP</td>
<td>ICCV 21 [47]</td>
<td></td>
<td>43.8</td>
<td>47.8</td>
<td>50.3</td>
<td>55.4</td>
<td>61.7</td>
<td>31.2</td>
<td>30.5</td>
<td>41.2</td>
<td>42.2</td>
<td>48.3</td>
</tr>
<tr>
<td>CME</td>
<td>CVPR 21 [23]</td>
<td></td>
<td>41.5</td>
<td>47.5</td>
<td>50.4</td>
<td>58.2</td>
<td>60.9</td>
<td>27.2</td>
<td>30.2</td>
<td>41.4</td>
<td>42.5</td>
<td>46.8</td>
</tr>
<tr>
<td>KFSOD</td>
<td>CVPR 22 [52]</td>
<td></td>
<td>44.6</td>
<td>-</td>
<td>54.4</td>
<td>60.9</td>
<td>65.8</td>
<td>37.8</td>
<td>-</td>
<td>43.1</td>
<td>48.1</td>
<td>50.4</td>
</tr>
<tr>
<td>Ours</td>
<td>FRCN-R101</td>
<td></td>
<td>47.0</td>
<td>44.8</td>
<td>53.4</td>
<td>62.9</td>
<td>65.2</td>
<td>29.8</td>
<td>32.6</td>
<td>46.3</td>
<td>47.7</td>
<td>53.0</td>
</tr>
</tbody>
</table>

Table 2: Evaluation on COCO dataset for novel classes for AP and AP75 settings. The highest score of each few-shot setting is in red, and the second highest score is in blue.

<table>
<thead>
<tr>
<th>Method</th>
<th>Shot</th>
<th>Novel AP</th>
<th>Novel AP75</th>
<th>bAP</th>
<th>bAP (fine-tuned)</th>
<th>nAP (fine-tuned)</th>
</tr>
</thead>
<tbody>
<tr>
<td>TFA w/ cos</td>
<td>ICML 20 [44]</td>
<td>10.0</td>
<td>13.7</td>
<td>9.3</td>
<td>13.4</td>
<td></td>
</tr>
<tr>
<td>FSCE</td>
<td>CVPR21 [40]</td>
<td>11.9</td>
<td>16.4</td>
<td>10.5</td>
<td>16.2</td>
<td></td>
</tr>
<tr>
<td>SRR-FSD</td>
<td>CVPR21 [53]</td>
<td>11.3</td>
<td>14.7</td>
<td>9.8</td>
<td>13.5</td>
<td></td>
</tr>
<tr>
<td>SVD</td>
<td>NeurIPS21 [48]</td>
<td>12.0</td>
<td>16.0</td>
<td>10.4</td>
<td>15.3</td>
<td></td>
</tr>
<tr>
<td>FORD+BL</td>
<td>IAMVIS22 [41]</td>
<td>11.2</td>
<td>14.8</td>
<td>10.2</td>
<td>13.9</td>
<td></td>
</tr>
<tr>
<td>N-PME</td>
<td>ICASSP22 [27]</td>
<td>10.6</td>
<td>14.1</td>
<td>9.4</td>
<td>13.6</td>
<td></td>
</tr>
<tr>
<td>Our</td>
<td></td>
<td>12.1</td>
<td>17.2</td>
<td>11.2</td>
<td>17.1</td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Ablation studies on different modules. The effect of incrementally adding each module to the Baseline network is demonstrated respectively. Sign * represents our reproductive results. We listed the base class mAP50 (bAP) during the pre-training and fine-tuning stage, as well as the novel class mAP50 (nAP) during the fine-tuning stage.

Table 4: Ablation studies on different hyperparameter settings. The effect of adjusting the Threcls is demonstrated. The highest nAP has been bolded.

<table>
<thead>
<tr>
<th>Modules</th>
<th>bAP (pre-trained)</th>
<th>bAP (fine-tuned)</th>
<th>nAP (fine-tuned)</th>
</tr>
</thead>
<tbody>
<tr>
<td>FSCE Baseline*</td>
<td>80.5</td>
<td>68.9</td>
<td>57.2</td>
</tr>
<tr>
<td>+ HSamp</td>
<td><strong>80.7</strong></td>
<td><strong>68.9</strong></td>
<td>57.6</td>
</tr>
<tr>
<td>+ Ternary Objectness</td>
<td>78.5</td>
<td>67.8</td>
<td>61.9</td>
</tr>
<tr>
<td>+ Contrastive Objectness</td>
<td>78.9</td>
<td>68.6</td>
<td><strong>62.9</strong></td>
</tr>
<tr>
<td>Threcls</td>
<td>0.05, 0.25, 0.5, 0.75, 0.95</td>
<td>nAP</td>
<td>60.5, 61.2, 62.1, <strong>62.9</strong>, 61.4</td>
</tr>
</tbody>
</table>

from 0.05 to 0.95 for training and record the nAP accordingly, as shown in Table 4. We observe that for lower Threcls, more candidate potential novel proposals can be distinguished. However, we have lower confidence and consequently lower quality. However, when a higher threshold Threcls is used, the number of candidates for potential novel proposals is smaller, which is insufficient to optimize objectness in our framework. As the result indicates, Threcls = 0.75 is a reasonable value for filtering the candidate proposals relatively well.

6. Conclusions

We improved the quality FSOD using R-CNN-based architecture via studying the phenomenon of objectness inconsistency due to the potential unlabeled novel objects that belong to unseen classes. By a balance anchor sampling strategy, we enhance the possibility of identifying anchors that may contain objects from unseen classes. In addition, we proposed HTRPN which leads to the recognition ability of potential novel objects and further enhances the objectness consistency. Our method mitigates model confusion about the novel classes and achieves SOTA performance on standard datasets. Future works include extensions to identify novel unseen classes in a zero-shot learning setting.
References


