

Few-Shot Object Detection by Attending to Per-Sample-Prototype

Hojun Lee^{1*} Myunggi Lee^{1,2} Nojun Kwak^{1†}

Seoul National University¹ NAVER WEBTOON²
{hojun815, nojunk}@snu.ac.kr myunggi@webtoonscorp.com

Abstract

Few-shot object detection aims to detect instances of specific categories in a query image with only a handful of support samples. Although this takes less effort than obtaining enough annotated images for supervised object detection, it results in a far inferior performance compared to the conventional object detection methods. In this paper, we propose a meta-learning-based approach that considers the unique characteristics of each support sample. Rather than simply averaging the information of the support samples to generate a single prototype per category, our method can better utilize the information of each support sample by treating each support sample as an individual prototype. Specifically, we introduce two types of attention mechanisms for aggregating the query and support feature maps. The first is to refine the information of few-shot samples by extracting shared information between the support samples through attention. Second, each support sample is used as a class code to leverage the information by comparing similarities between each support feature and query features. Our proposed method is complementary to the previous methods, making it easy to plug and play for further improvement. We have evaluated our method on PASCAL VOC and COCO benchmarks, and the results verify the effectiveness of our method. In particular, the advantages of our method are maximized when there is more diversity among support data.

1. Introduction

Multi-object detection is a classical computer vision task of recognizing and localizing the instances of specific objects categories from a given scene. In virtue of abundant images with bounding box annotations, object detection has experienced an enormous advancement with numerous deep learning-based approaches [28, 22, 25]. Notwithstanding its remarkable achievements, the methods still have diffi-

culty in learning novel object categories when the number of labeled data samples is small [32, 18]. Few-shot learning problems address such issues, which is common in real-world cases. However, learning few-shot samples by empirical risk minimization in a supervised manner easily overfits and may result in poor generalization [2, 40].

To alleviate this problem, several approaches have been studied, and *meta-learning* is one of the most successful ones in the few-shot classification scenario. In the few-shot setting, the model is given a small number of labeled *support* data for training, and at the time of inference, an input *query* image is classified as one of the support categories. Metric-based classification frameworks [37, 44, 31, 16, 34], one of the popular meta-learning methods, firstly calculate the centroid of each support class called *class prototype* from the support data, and then classify the query by measuring the similarity of the prototype with the query.

Since a significant progress has been made in the literature of few-shot classification problems, the problem of few-shot learning for object detection (FSOD) has also been studied. One of the successful approaches for FSOD is to extend typical meta-learning approaches to FSOD. One of the key issues in this line of research is how to aggregate the class prototype with the query image [42, 14, 9, 43, 41]. Although there have been performance improvements of FSOD through previous methods based on meta-learning, previous aggregation methods have a couple of main problems.

First, a handful of support data may be noisy and this can cause unexpected side effects. For example, instances of different categories may be close to one another or instances of the same class may differ in shape and perspective, which causes some samples to be far from intra-class samples in the feature space (Fig. 1 (a)). Therefore, if the information is not refined before averaging the support data, since it is a few-shot, the averaged prototype may be far from the centroid of the real distribution. Second, to our best knowledge, every method studied so far relies on class-wise *single representative* by averaging the information of

*This work is conducted during the author's research internship at NAVER WEBTOON Corp. †Corresponding author

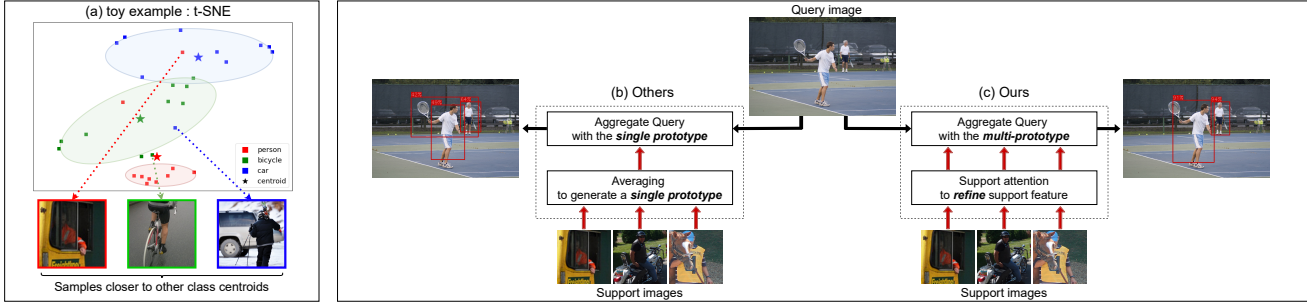


Figure 1. **Concept of our method:** LEFT: (a) Toy example when support data have large diversity and are misclustered. RIGHT: (b) Feature vectors of collected K -shot support images of a *person* contain diverse information and just averaging this information for aggregation with the query may deteriorate the detection performance. (c) Instead of using just a single averaged prototype per class, we use one prototype for each support image which has been refined by attending to other support images.

the support data, which are compared with the query data (Fig. 1 (b)). Instances in query images have large variations in size, perspective and even a possibility of occlusion. Furthermore, both the query and support images may have multiple instances with different categories close to each other. Therefore, rather than generating a single prototype per category that covers all the diversity and abolishing other information, it may be more advantageous to make better use of the information contained in the support data.

To resolve these problems, we propose a novel method (Fig. 1 (c)) consisting of two modules that aggregate the query and support data. First, we propose a method to refine the support information through an attention mechanism among support data before aggregating the query and support data. Second, rather than averaging the information of the support image, we use each support image as a prototype, which we call *per-sample prototype*. Through this method, we can better aggregate the diverse information of support data with queries.

We have applied the proposed method to two different architectures [42, 9] in different ways. Our method improves the average precision (AP) for new unseen classes on PASCAL VOC [7, 8] and COCO [21] benchmarks in both architectures. We also qualitatively verify that our method enhances the quality of the clusters available from support feature vectors of the same class via t-SNE [35].

Our contributions can be summarized as follows:

- We investigate that refinements of the support feature maps induce useful information for aggregation through an attention mechanism.
- We propose a method that aggregates query and support features without using one prototype per class, which allows fully leveraging information of support data.
- We show that our method can be applied to various types of architectures, and yields meaningful performance improvement on the PASCAL VOC and COCO benchmarks compared to our baselines [9, 42].

- Through t-SNE and clustering experiments, we demonstrate that intra-class support features are well clustered by our method, learning robust classifiable features.

2. Related Work

Object Detection It is the task of detecting instances of a specific category in an image. There have been many studies [28, 26, 22] on supervised learning with large annotated image datasets. Also, several variant tasks have been studied. For example, *weakly supervised object detection* [15, 6] is the task of learning to detect only with weak annotations (e.g., image-level category) without bounding box annotation. *Semi-supervised object detection* [13, 32] is a task using both labeled and unlabeled data, and *few-shot object detection*, which we deal with in this paper, aims to detect instances of novel categories with few samples.

Object detectors are largely divided into single-stage and two-stage detectors. The single-stage detectors [22, 20] predict the object’s class and bounding box directly from the features from the feature extractor. The two-stage detectors [28, 3] detect objects in two steps: first, they generate class-agnostic candidate boxes using RPN (Region Proposal Network). Then, the candidate boxes are classified and the corresponding bounding boxes are regressed. In FSOD, both single-stage [4, 14] and two-stage methods [43, 39, 9, 42] have been studied. Following the majority trend, we devised our method that can be applied to two-stage detectors such as Faster R-CNN [28].

Meta-Learning Briefly speaking, it is a research topic to learn how to learn. There are several approaches of meta-learning, such as 1) gradient-based methods [10, 17] that learn to well-transfer the knowledge learned by several tasks to a new task and 2) model-based methods [23, 24] that aim to design structures that can generalize well, or utilize an external meta-learner or memory. 3) metric-based methods [31, 37, 27, 44] that perform non-parametric learning by comparing query sample with support samples and predicting the category of test data by comparing with train-

ing support samples. A typical strategy of applying metric-based learning to few-shot learning is to generate a single class prototype for each category by averaging features of support data belonging to the category [31, 37, 34]. If networks learn sampled mini-batches called *episode* through meta-learning, an episode will consist of sub-classes at each iteration. This episodic training strategy has been shown to generalize better on novel few-shot data because it naturally mimics the few-shot task. In this paper, instead of generating one prototype for each class, we propose an alternative method, generating per-sample prototypes, that makes better use of the information of the support data by treating each sample as a prototype.

Few-Shot Object Detection One of the most successful approaches in FSOD are meta-learning-based methods. Given a support data S composed of K samples of a specific category and a query image Q , the goal of FSOD methods based on meta-learning is to recognize and localize the instances of Q with the help of S . MetaDet [41], MetaYOLO [14] and Meta R-CNN [43] proposed methods for meta-learner to generate a prototype per category from the support data and aggregate these prototypes with the query features by channel-wise multiplication. FsDetView [42] showed that when query features are aggregated with support features, it is more effective to concatenate three features together: the channel-wise multiplication feature map of query and support, the subtraction feature map of query and support, and the query feature map itself. FewX [9] proposed a method of aggregating query features and support features before the region proposal process, unlike the previous methods that aggregate after the region proposal process. All of these methods generate a single prototype per category when performing aggregation, but we propose a novel method of aggregating each sample through the per-sample prototype.

3. Approach

3.1. Problem Formulation

In the few-shot object detection scenario, we assume we have two sets of data sources, D_b and D_n . D_b is a *base* dataset with abundant annotated instances of base classes C_b , and D_n is a *novel* dataset with few labeled instances of novel classes C_n . Here, we assume there are no overlapping classes between C_b and C_n , i.e., $C_b \cap C_n = \emptyset$. Few-shot object detection aims to train a detector with limited data source D_n to recognize and localize novel instances of categories C_n with the help of knowledge learned from the base dataset D_b . In this paper, as with the previous researches [16, 14, 42, 39], it is assumed that the novel dataset, D_n , is composed of K annotated instances per category. Also, the number of novel categories is N , and we call this problem a K -shot, N -way few-shot object detection problem.

3.2. Overall Architecture

Our method aims to find the novel instances in the query image by leveraging meta-learning. In the meta-learning scenario, we concentrate on two aspects of the pipeline: how to extract informative representation from support set S and how to combine it with query features. To this end, we propose the *Intra-Support Attention Module* (ISAM) and the *Query-Support Attention Module* (QSAM), which can complement various frameworks in a plug-and-play manner and bring significant improvements in performance.

Figure 2 shows the overall architecture illustrating two baselines [9, 42] we used for FSOD. As shown in the figure, there are two candidate locations to apply the proposed aggregation modules. For different baselines [9] and [42], we designed similar aggregation methods composed of ISAM and QSAM and applied them at different locations as shown by [A] and [B] in Fig. 2.

Specifically, the overall architecture is based on the two-stage Faster R-CNN framework [28]. Backbone network receives a query image and K samples from the support set of the same class and outputs $K + 1$ feature maps. The *Region Proposal Network* (RPN) proposes candidate boxes from the query feature map. Here, the input for the RPN depends on the baseline. In FewX [9], the aggregated query feature map is fed into RPN ([A] in Fig. 2). On the other hand, in FsDetView [42], the query feature map is fed directly into the RPN without aggregation. In both baselines, N_{roi} query RoI features are aggregated with K support RoI features. Then, *RoI Head* outputs box offsets and class confidences.

Note that both baselines [9, 42] generate a single prototype per class by averaging the support feature maps before aggregation at [A] for FewX and at [B] for FsDetView. Unlike the two baselines, our method performs aggregation by treating the support features as an individual prototype, i.e., per-sample prototype. The detailed per-sample aggregation procedures of ISAM and QSAM are introduced in Section 3.3 and Figure 3.

3.3. Aggregation Module

We propose a novel approach to aggregate the query and support features by treating each support feature as an individual prototype rather than generating a single prototype per category. Figure 3 shows the aggregation methods according to the type of query feature map: (a) is for spatially boosting the prominence of the areas in the query feature map similar to the support RoI feature vectors, and (b) is to strengthen further the query RoI feature similar to support RoI feature vectors. Query Spatial Aggregation (a) is applied to Fig. 2 [A], and Query RoI Aggregation (b) is applied to Fig. 2 [B]. As shown in the figure, both of our aggregation methods are composed of two stages of *Intra-Support Attention Module* (ISAM) to refine the support features through the self-attention mechanism before aggrega-

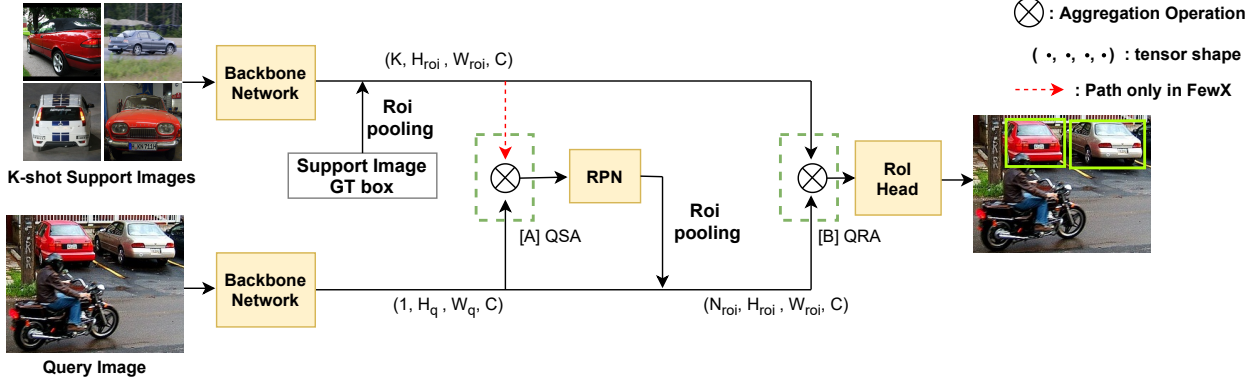


Figure 2. **Our overall Architecture** based on Faster R-CNN [28] to find instances of support category in the query image. [A] and [B] are aggregation procedure (Fig. 3) for the query feature and the support features. Baselines are FewX [9] and FsDetView [42], where FewX has both [A] and [B] operations, and in FsDetView, query features are directly fed into RPN without aggregation of [A] operation.

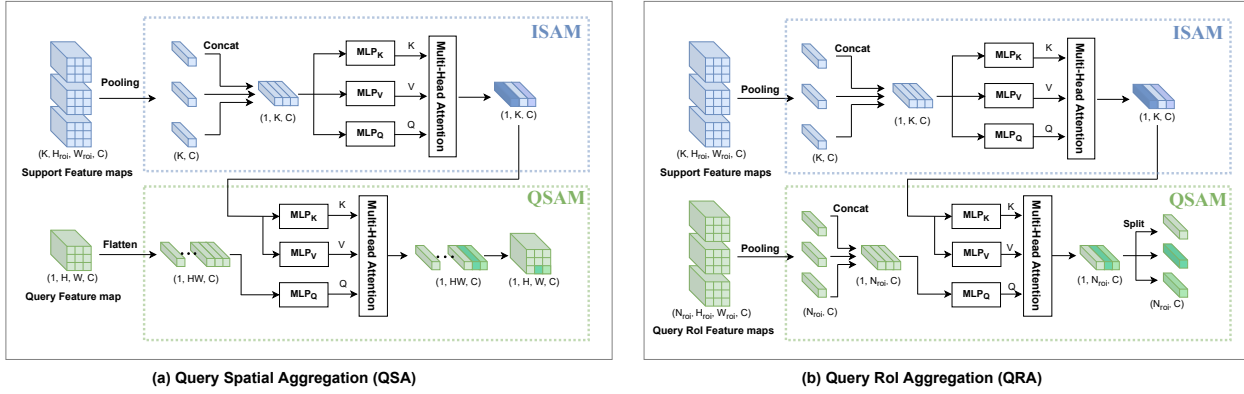


Figure 3. Two types of aggregation procedures (3-shot example) of the query feature maps and K support feature maps. (a) is for spatial attention to the query feature map and (b) is for aggregation of N_{roi} query feature maps and K support feature maps.

tion and *Query-Support Attention Module* (QSAM) that aggregates the queries with the supports.

Intra-Support Attention Module aims to refine each individual support feature vector with the help of other support feature vectors. Because some support feature vectors may be too far from other vectors due to the diverse nature of the support images, it can lead to performance degradation. For example, an upright instance of a person can represent the characteristics of the *person* class well, while an instance of a person doing a handstand cannot. Therefore, attention mechanism [36] shown in Eq. (1) is utilized to enhance the shared information of the support features. Especially, following [36], we use multi-head attention:

$$Q_{attn} = \text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d}}\right)V \quad (1)$$

$$Q_{attn}, Q \in \mathbb{R}^{q \times d}, \quad K, V \in \mathbb{R}^{v \times d}$$

where d is the length of a feature vector. q and v are the number of queries and keys respectively. Here, all of Q , K and V are assigned to the support feature vectors, i.e., the number of support samples $q = v = K$. Then, the support

feature vectors can pay attention to one another so that the inherent characteristics of the data can be refined.

For implementation, ISAM is designed as the encoder of shallow *Transformer* [36, 38] composed of *multi-head attention network* consisting of the process of Eq. (1) and *multi-layer perceptron* with *layer normalization* [1].

Query-Support Attention Module aggregates the query feature and support feature maps through attention mechanism shown in Eq.(1). Here, K and V are assigned to the support feature vectors, and Q is assigned to the query feature vectors. The query feature vectors are generated in one of two processes: by flattening the query feature map in (a), i.e., $q=HW$, or by concatenating query RoI feature vectors in (b), i.e., $q=N_{roi}$. In other words, aggregation the query features and the support features are performed by dot-producting each query feature vector with all of the support feature vectors.

For implementation, QSAM is designed as the decoder of shallow *Transformer* which is also composed of *multi-head attention network* containing *multi-layer perceptron* blocks with *layer normalization*.

3.4. Training and Inference

Training Our framework is trained by two phases. First, the network is trained with abundant labeled base data D_b with base class C_b . At this phase, the trained classes are C_b , i.e., $C_{train} = C_b$. Second, the network is finetuned with few-shot novel data D_n of novel classes C_n . At this phase, the training is done on a balanced dataset composed of K -shot instances per class for both base data and novel data, i.e., $C_{train} = C_b \cup C_n$. For both phases, the episodic training strategy is applied that each episode consists of N -way, K -shot support data and a query image. FewX [9] are trained with 2-way, K -shot. Specifically, an episode consists of the following triplet: (q_{c1}, s_{c1}, s_{c2}) where class $c1$ and $c2$ are different classes sampled from C_{train} , and q_{c1} indicates the query data containing instances of $c1$ -class. s_{c1} and s_{c2} indicate the 2-way, K -shot support data, i.e., $|s_{c1}| = |s_{c2}| = K$. And for FsDetView [42], an episode consists of a query image and all class of support data, i.e., $|C_{train}|$ -way, K -shot.

Objective function The loss function of RPN’s foreground proposal and RoI Head’s detection outputs are Eq. (2) where $\mathcal{L}_{\cdot,loc}$ is the bounding box regression loss calculated as the smooth \mathcal{L}_1 loss, and $\mathcal{L}_{\cdot,cls}$ is the classification loss calculated as the cross-entropy loss. Note that the output of FewX is binary classification whether the query RoI feature vectors match or not with the support RoI feature vectors, and FsDetView is multi-class classification with the softmax function. \mathcal{L}_{meta} is the cross-entropy loss used in FsDetView like Meta R-CNN [43] for class features to be diverse for different classes.

$$\mathcal{L} = \mathcal{L}_{rpn,loc} + \mathcal{L}_{rpn,cls} + \mathcal{L}_{det,loc} + \mathcal{L}_{det,cls} + \mathcal{L}_{meta} \quad (2)$$

Inference The few-shot samples used in finetuning are used as the support data at the inference time. Therefore, all the support data are passed into the backbone network and ISAM once, and the output support feature vectors of ISAM are stored for repeated use as multiple prototypes.

4. Experiments

4.1. Dataset

We evaluate our method on PASCAL VOC [7, 8] and MS COCO [21] benchmarks. We follow the experimental setup of previous works [43, 14, 39, 42]. For VOC experiments, our network is trained using PASCAL VOC 07+12 trainval dataset and tested on VOC 07 test dataset. The 20 classes are divided into 15 base classes and 5 novel classes and evaluated with three different splits. The number of shots is set to $K \in \{1, 2, 3, 5, 10\}$. For 10-shot and 30-shot experiments on MS COCO, of the total 80 classes of COCO, the 20 classes overlapping with those of VOC are set as novel classes. As in TFA [39], it is assumed that K -shot base data can be used when finetuning with K -shot

novel data. Because the performance can vary depending on the few-shot sample configuration, we distinguished between the experiments in fixed support samples and the experiments in which random sampling is performed multiple times.

4.2. Implementation detail

The query images are resized to short size 600 and the maximum long side is set to be 1000. The support images are resized to 320x320 for FewX and 224x224 for FsDetView. We use Resnet-101 [11] for the VOC experiment and Resnet-50 for the COCO experiment as the weight-shared backbone network except the first convolution layer of FsDetView. Because input support data for FsDetView have 4 channels that consists of 3 rgb channels and 1 channel for the binary mask of ground truth bounding box, FsDetView has a 3-channel convolution layer for the query image and a 4-channel convolution layer for the support data. The backbone networks are pretrained on Imagenet-1k [29]. The learnable parameters of batchnorm [12] trained on imagenet 1k are frozen at both base training and finetuning for both baselines. Learning schedulings are the same for both FewX and FsDetView including the optimizer, learning rate, batch size and training/finetuning iterations. Unless otherwise noted, the numbers of support samples K during base training of FewX, FewX+Ours, FsDetView and FsDetView+Ours are set to 10, 10, 1 and 3¹, respectively.

ISAM and QSAM are implemented by utilizing the encoder and the decoder of the *Transformer* [36], respectively. Both are set to have 2-heads, 2-layers with *layer normalization*, and *ReLU* is used as an activation function. The dropout [33] rate of Transformer is 0.1. And the hidden dimensions of the Transformer are set to 256.

4.3. Analysis for ISAM and QSAM

We applied our method to FsDetView [42] and analyzed how the performance changes on PASCAL VOC by two experiments: ablation study of ISAM and QSAM (Table 1), and the effect of the number of per-class samples K during base training (Table 2). All experiments in Table 1 and Table 2 were conducted 30 times to calculate the means and standard deviations.

Ablation study Table 1 shows the average precision at IoU=0.50 (Intersection over Union) of the novel classes with or without ISAM and QSAM on PASCAL VOC07 test dataset. It can be seen that it is effective to refine support features by paying attention to other support features through ISAM. In addition, it can be seen that aggregation with the support features as they are through QSAM is more effective than generating a single class prototype by averaging support features. The table shows that both modules

¹ The K during base training of FewX and FsDetView without ours followed the official code.

method	Average precision at IoU=0.5				
	K=1	K=2	K=3	K=5	K=10
FsDetView	23.8 ± 6.0	35.9 ± 6.1	42.1 ± 4.3	48.7 ± 3.4	56.9 ± 2.9
FsDetView+ISAM	24.0 ± 6.1	35.6 ± 5.2	44.0 ± 4.6	50.0 ± 4.0	57.9 ± 3.1
FsDetView+QSAM	23.9 ± 6.9	35.9 ± 5.5	43.9 ± 4.9	50.5 ± 3.9	58.1 ± 3.0
FsDetView+ISAM+QSAM	24.3 ± 6.2	36.5 ± 5.3	44.9 ± 4.3	52.0 ± 3.8	59.2 ± 2.6

Table 1. Ablation study on Novel set 1 of VOC07 test dataset. At base training, 3 support images per class are used.

method	base train K	Average precision at IoU=0.5				
		K=1	K=2	K=3	K=5	K=10
FsDetView	1	23.8 ± 6.5	35.3 ± 6.0	41.8 ± 4.8	48.3 ± 3.6	56.5 ± 2.9
FsDetView	3	23.8 ± 6.0	35.9 ± 6.1	42.1 ± 4.3	48.7 ± 3.4	56.9 ± 2.9
FsDetView	10	23.2 ± 5.7	34.6 ± 5.9	41.5 ± 4.5	48.8 ± 3.4	57.1 ± 2.9
FsDetView+ISAM+QSAM	1	24.7 ± 6.5	35.9 ± 5.4	42.2 ± 4.4	50.4 ± 4.0	57.2 ± 3.0
FsDetView+ISAM+QSAM	3	24.3 ± 6.2	36.5 ± 5.3	44.9 ± 4.3	52.0 ± 3.8	59.2 ± 2.6
FsDetView+ISAM+QSAM	10	24.1 ± 5.9	36.2 ± 5.1	44.1 ± 4.2	51.8 ± 3.7	59.8 ± 2.2

Table 2. Ablation study with changes of K during base training on Novel set 1 of VOC07 test dataset.

helped improve detection performance, and both processes work better as the number of shots K increased.

The number of shots K during base training Table 2 shows the AP50 results of the novel classes on PASCAL VOC07 test dataset according to the change of K during base training. Our performances are generally higher, but the more similar the K for base training and the K for finetuning, the higher the performance. When FsDetView is trained by base data with ours at $K = 1$, ISAM did not learn how to pay attention to support features, and QSAM did not learn how to aggregate multiple prototypes together. Even if ISAM and QSAM learn their roles when finetuning, the performances were lower than those of $K = 3$ or 10 when base training. In addition, the higher the number of shots during base training, the lower the standard deviations.

4.4. Clustering of support feature vectors

t-SNE Our hypothesis is that collected natural images can be far from class prototypes. Hence, it is better that support features are refined into shared information through ISAM by paying attention to other support features. Figure 4 is the t-SNE [35] results of the novel classes on COCO 30-shot to verify the hypothesis. As in (a) and (b) of the figure, some features exist close to others despite the categories are different. In these cases, the data are noisy; for example, there are instances of several categories together in RGB images (a), (b). However, as can be seen in (c), ISAM makes clustering better for support features by paying attention to other support features. Some points are misclustered in all of (a), (b), and (c) when there are ambiguities in RGB images, such as multiple categories, occlusions, or partial appearances.

Distance from class centroid We evaluated quantita-

tively whether each support feature vector, which we plotted on t-SNE, is actually closest to the corresponding class mean (single prototype) calculated by 30-shots on the novel support data of COCO 30-shot. The accuracy was evaluated by measuring L1-distance with class means. The accuracies of Baseline, before ISAM and after ISAM are 75.2%, 77.8% and 97.8%, respectively.

4.5. Comparison with state-of-the-art

We applied our method to two baselines [9, 42] and compared it with other methods on PASCAL VOC and COCO benchmarks. Note that the base models are trained by base data with $K=10$ for our method with FewX and $K=3$ for ours with FsDetView, as mentioned in implementation details (Sec. 4.2). Both models are finetuned from each base model and are evaluated on novel classes.

PASCAL VOC Table 3 shows the AP50 results of the novel classes on PASCAL VOC07 test dataset. We evaluated Ours with FewX [9] with the same few-shot samples as MetaYOLO [14] and TFA [39], and there are significant performance improvements compared to the baseline. However, as shown in Table 1, the variance of performance is large. Therefore, we evaluated ours with FsDetView [42] by averaging 30 times of random samplings of few-shot samples. Likewise, significant performance improvements are found in FsDetView [42].

MS COCO Table 4 summarizes the results for novel classes on MS COCO dataset, and we report the standard COCO metrics, average precision (AP) and average recall (AR). As shown in the table, our methods outperform the baselines in both cases of 10 shots and 30 shots. In addition, the box plot is visualized by repeating it 30 times in Fig. 5. As shown in the figure, if the networks learned the same

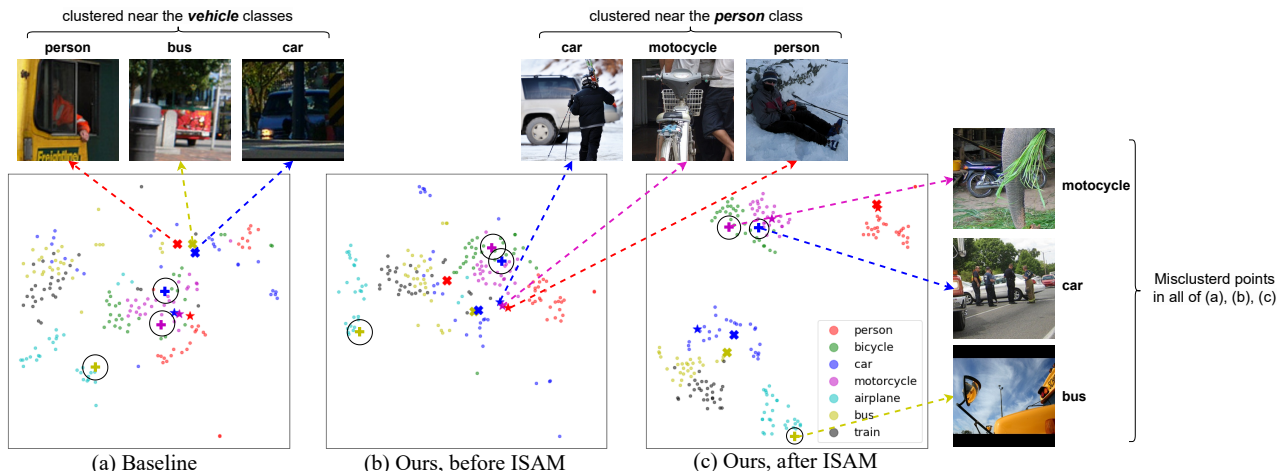


Figure 4. t-SNE visualization for 30-shot support feature vectors of MS COCO novel classes. (b) and (c) are support feature vectors before and after our ISAM, respectively. For visualization, some samples are marked with different markers (+, X, *), and only some classes are plotted. Some patches shown on the rightmost column, which are considered as misclustered in all of (a), (b), and (c), are circled.

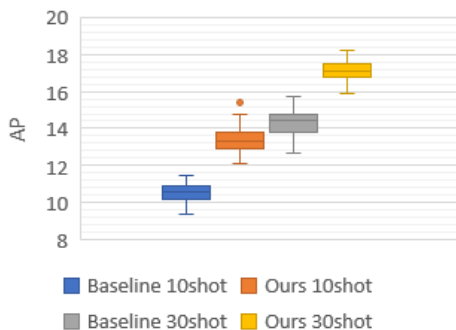


Figure 5. Box plot of Ours with FsDetView on MS COCO

shot, it is confirmed that the min value of ours is higher than the max value of the baseline.

5. Future work

Aggregation with spatial information of support data

As the support feature vectors were abstracted, spatial information of the vectors disappeared. Therefore, it is worth noting that the attention mechanism operates to reflect *spatial information of support images*. In other words, when ISAM refines the support data, it is helpful to pay attention to each other without the support feature maps being pooled by global average pooling. Similarly, when QSAM aggregates the query with the support, it is helpful to aggregate with spatial information of support feature maps rather than that of support feature vectors.

Toward class scalable detector We evaluate Ours with FewX trained only on the base data of COCO dataset without finetuning on the novel data. The novel 20-class AP is 7.1 (+0.9 point increase compared to baseline) even though the novel data were not finetuned. Because the current

framework based on metric-based meta-learning has the form of $P(box|query\ image, support\ images)$. If the networks learn how to match given support images with the query image well, detection can be performed without finetuning the novel classes. If this characteristic is well utilized, it may be more advantageous to detect many unknown classes in the same domain or to perform it incrementally.

6. Conclusion

There are studies on the Few-Shot Object Detection framework based on meta-learning that detects instances of support category in a query image. Based on this framework, we propose the *Intra-Support Attention Module* (ISAM) and *Query-Support Attention Module* (QSAM) applicable to various methods. ISAM performs an attention mechanism between support features of the same class to refine the information that may be noisy, and QSAM aggregates the query features and the support features by per-sample prototypes, not a single prototype per class for using unabridged information of support data. Better feature maps for detecting unseen novel classes in K -shot support data are generated through these two modules. We demonstrate the effectiveness of the proposed modules in that the support feature vectors are clustered when collected support samples are somewhat far from the prototype. And the performances are improved when the attention vectors were refined and aggregated as per-sample prototypes.

Acknowledgments

This work was supported by IITP grant (No. 2021-0-01343, Artificial Intelligence Graduate School Program) and NRF grant (2021R1A2C3006659), both funded by Korean Government.

method	backbone	Novel Set 1					Novel set 2					Novel set 3				
		1	2	3	5	10	1	2	3	5	10	1	2	3	5	10
LSTD [4]	Darknet19	8.2	11.0	12.4	29.1	38.5	11.4	3.8	5.0	15.7	31.0	12.6	8.5	15.0	27.3	36.3
YOLOv2-ft [14]		6.6	10.7	12.5	24.8	38.6	12.5	4.2	11.6	16.1	33.9	13.0	15.9	15.0	32.2	38.4
MetaYOLO [14]		14.8	15.5	26.7	33.9	47.2	15.7	15.3	22.7	30.1	40.5	21.3	25.6	28.4	42.8	45.9
MetaDet [41]		17.1	19.1	28.9	35.0	48.8	18.2	20.6	25.9	30.6	41.5	20.1	22.3	27.9	41.9	42.9
MetaDet [41]	VGG16 [30]	18.9	20.6	30.2	36.8	49.6	21.8	23.1	27.8	31.7	43.0	20.6	23.9	29.4	43.9	44.1
NP-RepMet [45]	R101DCN-FPN	37.8	40.3	41.7	47.3	49.4	41.6	43.0	43.4	47.4	49.1	33.3	38.0	39.8	41.5	44.8
TFA w/ fc * [39]	R101FPN	36.8	29.1	43.6	55.7	57.0	18.2	29.0	33.4	35.5	39.0	27.7	33.6	42.5	48.7	50.2
TFA w/ cos * [39]	R101FPN	39.8	36.1	44.7	55.7	56.0	23.5	26.9	34.1	35.1	39.1	30.8	34.8	42.8	49.5	49.8
Meta R-CNN [43]	R101	19.9	25.5	35.0	45.7	51.5	10.4	19.4	29.6	34.8	45.4	14.3	18.2	27.5	41.2	48.1
FewX †	R101	29.8	35.5	36.3	48.4	53.6	22.2	28.9	25.2	31.2	39.7	24.3	29.9	34.4	47.1	50.4
FewX+Ours	R101	31.1	36.1	39.2	50.7	59.4	22.9	29.4	32.1	35.4	42.7	24.3	28.6	35.0	50.0	53.6
TFA w/ cos * b [39]	R101FPN	25.3	36.4	42.1	47.9	52.8	18.3	27.5	30.9	34.1	39.5	17.9	27.2	34.3	40.8	45.6
FsDetView b [42]	R101	24.2	35.3	42.2	49.1	57.4	21.6	24.6	31.9	37.0	45.7	21.2	30.0	37.2	43.8	49.6
FsDetView † b	R101	23.8	35.3	41.8	48.3	56.5	19.4	26.4	30.3	36.6	44.6	21.7	31.3	34.2	40.6	47.9
FsDetView+Ours b	R101	24.3	36.5	44.9	52.0	59.2	20.5	27.5	33.1	40.9	47.1	22.4	33.0	37.8	43.9	51.5

Table 3. AP50 on VOC2007 test dataset. The first four rows are based on YOLOv2 [26], and the rest are based on Faster R-CNN [28] with/without FPN [19] or DCN [5]. Methods with * marks are based on finetuning and the others are based on meta-learning. † indicates the re-implemented version using the official code. b marks mean multiple-run results. Red/Blue texts indicate the first/second best on multiple-run results.

shots	method	backbone	image size	AP			AR		
				AP50:95	AP50	AP75	1	10	100
10	MetaYOLO [14]	Darknet19	416x416	5.6	12.3	4.6	10.1	14.3	14.4
	MetaDet [41]	VGG16	-	7.1	14.6	6.1	11.9	15.1	15.5
	TFA w/ fc * [39]	R101FPN	short800	10.0	19.2	9.2	-	-	-
	TFA w/ cos * [39]	R101FPN	short800	10.0	19.1	9.3	-	-	-
	Meta R-CNN [43]	R50	short600	8.7	19.1	6.6	12.6	17.8	17.9
	FewX † [9]	R50	short600	11.9	23.7	10.6	19.1	26.2	26.3
	FewX+Ours	R50	short600	13.0	24.7	12.1	19.3	27.7	27.8
	TFA w/ fc * b [39]	R101FPN	short800	9.1	17.3	8.8	-	-	-
	TFA w/ cos * b [39]	R101FPN	short800	9.1	17.1	8.8	-	-	-
	FsDetView b [42]	R50	short600	12.5	27.3	9.8	20.0	25.5	25.7
FsDetView † b	R50	short600	10.6	25.5	6.3	18.1	23.8	23.9	
FsDetView+Ours b	R50	short600	13.4	30.6	9.1	20.7	26.7	26.8	
30	MetaYOLO [14]	Darknet19	416x416	9.1	19.0	7.6	13.2	17.7	17.8
	MetaDet [41]	VGG16	-	11.3	21.7	8.1	14.5	18.9	19.2
	TFA w/ fc * [39]	R101FPN	short800	13.4	24.7	13.2	-	-	-
	TFA w/ cos * [39]	R101FPN	short800	13.7	24.9	13.4	-	-	-
	Meta R-CNN [43]	R50	short600	12.4	25.3	10.8	15.0	21.4	21.7
	FewX † [9]	R50	short600	13.8	25.8	13.5	20.8	30.8	31.0
	FewX+Ours	R50	short600	15.3	29.3	14.5	21.2	31.7	32.1
	TFA w/ fc * b [39]	R101FPN	short800	12.0	22.2	11.8	-	-	-
	TFA w/ cos * b [39]	R101FPN	short800	12.1	22.0	12.0	-	-	-
	FsDetView b [42]	R50	short600	14.7	30.6	12.2	22.0	28.2	28.4
FsDetView † b	R50	short600	14.3	31.5	10.6	21.9	28.7	28.8	
FsDetView+Ours b	R50	short600	17.1	35.2	14.7	24.8	31.2	32.0	

Table 4. AP and AR of novel classes on MS COCO minival. † is re-implemented using official code. * marks method based on finetuning. b marks multiple-run results. Red/Blue texts indicate the first/second best on multiple-run results.

References

- [1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.
- [2] Olivier Bousquet, Stéphane Boucheron, and Gábor Lugosi. Introduction to statistical learning theory. In *Summer School on Machine Learning*, pages 169–207. Springer, 2003.
- [3] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018.
- [4] Hao Chen, Yali Wang, Guoyou Wang, and Yu Qiao. Lstd: A low-shot transfer detector for object detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.
- [5] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017.
- [6] Thibaut Durand, Taylor Mordan, Nicolas Thome, and Matthieu Cord. Wildcat: Weakly supervised learning of deep convnets for image classification, pointwise localization and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 642–651, 2017.
- [7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. <http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html>.
- [8] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. <http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html>.
- [9] Qi Fan, Wei Zhuo, Chi-Keung Tang, and Yu-Wing Tai. Few-shot object detection with attention-rpn and multi-relation detector. In *CVPR*, 2020.
- [10] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.
- [11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [12] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR, 2015.
- [13] Jisoo Jeong, Seungeui Lee, Jeessoo Kim, and Nojun Kwak. Consistency-based semi-supervised learning for object detection. 2019.
- [14] Bingyi Kang, Zhuang Liu, Xin Wang, Fisher Yu, Jiashi Feng, and Trevor Darrell. Few-shot object detection via feature reweighting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8420–8429, 2019.
- [15] Daesik Kim, Gyujeong Lee, Jisoo Jeong, and Nojun Kwak. Tell me what they’re holding: Weakly-supervised object detection with transferable knowledge from human-object interaction. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11246–11253, 2020.
- [16] Aoxue Li, Tiange Luo, Tao Xiang, Weiran Huang, and Liwei Wang. Few-shot learning with global class representations. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9715–9724, 2019.
- [17] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [18] Yu Li, Tao Wang, Bingyi Kang, Sheng Tang, Chunfeng Wang, Jintao Li, and Jiashi Feng. Overcoming classifier imbalance for long-tail object detection with balanced group softmax. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10991–11000, 2020.
- [19] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.
- [20] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017.
- [21] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015.
- [22] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.
- [23] Tsendsuren Munkhdalai and Hong Yu. Meta networks. In *International Conference on Machine Learning*, pages 2554–2563. PMLR, 2017.
- [24] Sachin Ravi and Hugo Larochelle. Optimization as a model for few-shot learning. 2016.
- [25] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.
- [26] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2017.
- [27] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. *arXiv preprint arXiv:1803.00676*, 2018.
- [28] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *arXiv preprint arXiv:1506.01497*, 2015.
- [29] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and

- Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.
- [30] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [31] Jake Snell, Kevin Swersky, and Richard S Zemel. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175*, 2017.
- [32] Kihyuk Sohn, Zizhao Zhang, Chun-Liang Li, Han Zhang, Chen-Yu Lee, and Tomas Pfister. A simple semi-supervised learning framework for object detection. *arXiv preprint arXiv:2005.04757*, 2020.
- [33] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1):1929–1958, 2014.
- [34] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1199–1208, 2018.
- [35] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.
- [36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017.
- [37] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. *arXiv preprint arXiv:1606.04080*, 2016.
- [38] Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F Wong, and Lidia S Chao. Learning deep transformer models for machine translation. *arXiv preprint arXiv:1906.01787*, 2019.
- [39] Xin Wang, Thomas E Huang, Trevor Darrell, Joseph E Gonzalez, and Fisher Yu. Frustratingly simple few-shot object detection. *arXiv preprint arXiv:2003.06957*, 2020.
- [40] Yaqing Wang, Quanming Yao, James T Kwok, and Lionel M Ni. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (CSUR)*, 53(3):1–34, 2020.
- [41] Yu-Xiong Wang, Deva Ramanan, and Martial Hebert. Meta-learning to detect rare objects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9925–9934, 2019.
- [42] Yang Xiao and Renaud Marlet. Few-shot object detection and viewpoint estimation for objects in the wild. In *European Conference on Computer Vision*, pages 192–210. Springer, 2020.
- [43] Xiaopeng Yan, Ziliang Chen, Anni Xu, Xiaoxi Wang, Xi-aodan Liang, and Liang Lin. Meta r-cnn: Towards general solver for instance-level low-shot learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9577–9586, 2019.
- [44] Shuo Yang, Lu Liu, and Min Xu. Free lunch for few-shot learning: Distribution calibration. *arXiv preprint arXiv:2101.06395*, 2021.
- [45] Yukuan Yang, Fangyun Wei, Miaoqing Shi, and Guoqi Li. Restoring negative information in few-shot object detection. *arXiv preprint arXiv:2010.11714*, 2020.