CS282 Machine Learning Report

Junda Shen [*1] **Zian He** [*1]

Abstract

In this project, we try to conquer the few-shot object detection problem by fine-tuning the last layer of some existing convolutional object detectors, e.g. Faster R-CNN. The prior methods usually encounter high variance problems which made them unreliable. However, we used an elegant sampling method so that the results are stable and the overfitting problem is diminished. We evaluated our approach on COCO and LVIS datasets, which shows that our method outperforms prior methods by $2 \sim 15$ percent. Since this method does not propose a new architecture, we implement it based on the Detectron2 package, which already contains many pretrained CNN architectures and is best suited for a fine-tuning task.

1. Introduction

Nowadays, many tasks can be done by machine perception systems and learning from only a few samples has raised wide attention due to the scarcity of samples in some particular areas. Although some few-shot learning problems has almost been solved, it is still hard to achieve image detection with few training samples, the detection results are still unideal.

In the prior works, people have tried to achieve few-shot image detection by two approaches

1. Meta-learning. This approach tries to transfer knowledge learnt from data-abundent classes to data-scarce novel classes so that the model could adapt to the new task more quickly. This method have been used in few-shot image classification for a while but it has not been proved effective in object detection task, which is much more challenging than image classification.

2. Metric-learning. This approach tries to build some metrics for the learner so that it can estimate and check the similarities between images, e.g. cosine similarities. Based on the built metrics, the model can try to fit to the input and learn how to detect objects. However, this approach relis on the metric, which may be hard to find a better one.

[*]Equal contribution [1]SIST, Shanghaitech University. Correspondence to: Junda Shen <shenjd@shanghaitech.edu.cn>, Zian He <heza@shanghaitech.edu.cn>.

The object detection task is much harder than the image classification task, since it contains not only classification but also object localization. Researchers (Kang et al., 2019; Yan et al., 2019; Wang et al., 2019b) have attempted to achieve few-shot object detection task, where a few labeled data, far from abundent, is provided to the learner as novel training data. These methods achieved out-of-random performance but their methods are not stable, which means the reproduced results might be far from their reported results.

In this project, we try to improve the detection results by adopting fine-tuning based approaches. We focus on the schedule of the training procedure and use proper instance-level feature normalization in this project.

The training is composed of two stages, which is shown in Figure 1. The first stage trains the whole network, e.g. Faster R-CNN, on data-abundent base classes. Then we only train the last layer of the network in the second stage, on a small balanced training set that is composed of both base and novel classes by properly sample data, parameters in other layers of the detector is kept intact during the second stage. We also use instance-level feature normalization in the second stage, which helps the model to diminish some parameter issues introduced by Gidaris & Komodakis (2018); Qi et al. (2018); Chen et al. (2019).

In our evalution, we find that our method outperfroms all previous few-shot object detection methods, our method can gain $2 \sim 15$ percent more accuracy than prior methods, and it even doubled the accuracy in one-shot learning, which dictates the effiction of our method.

We fixed several issues in the existing training procedures

1. We try to diminish overfitting by invoking multiple runs on different random samples so that the variance is lower and the results much are more stable.

2. We try to keep the accuracy of our method consistent with the paper.

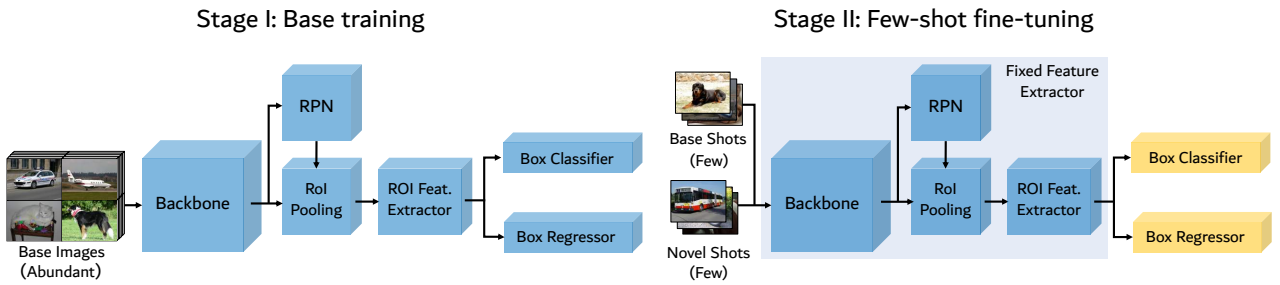

Figure 1. The two-stage fine-tuning approach (TFA). In the base training stage, the entire object detector are jointly trained on the base classes. In the second stage, the labeled feature extractor is kept intact and only the last layer, i.e. box predictor, is fine-tuned on a balanced set consisting sampled base and novel classes.

3. We report not only the accuracy for novel classes, but also accuracy for base classes and the overall accuracy for all classes, so that one can see that the accuracy (referred to as the generalized few-shot learning setting in the few-shot classification literature (Hariharan & Girshick, 2017; Wang et al., 2019a)) for other classes are not largely affected.

2. Algorithms for Few-Shot Object Detection

In this section, we first focus on the few-shot object detection setting. Then, we will show our two-stage fine-tuning approach in Section 2.2 and Section 2.3.

2.1. Task formalization

There are two kinds of classes, one kind is base classes C_b that have many instances and the other one is novel classes C_n that have only K (usually less than 10) instances per class. For an object detection dataset $\mathcal{D} = \{(x, y), x \in \mathcal{X}, y \in \mathcal{Y}\}$ (x is the input image, $y = \{(c_i, \mathbf{l}_i), i = 1, ..., N\}$ denotes the categories $c \in C_b \cup C_n$ and bounding box coordinates $\mathbf{l}$ of the N object instances in the image x). Now we will use the dataset $\mathcal{D}$ to learn categorie c and the corresponding box corrdinates l for each object and try to improve the detection accuracy.

2.2. Two-stage fine-tuning approach

Our method (TFA) including two stages, which is shown in Figure 1. The base detection model is formed by Faster R-CNN (Ren et al., 2015) and a two-stage object detector.

Since the features in the first few layers are class-agnostic, features learned from the base classes are likely to transfer to the novel classes with parameters fixed.

Base model training. In this stage, we train the network with large number of samples. The loss of the network consists of three parts,

$$\mathcal{L} = \mathcal{L}_{\text{rpn}} + \mathcal{L}_{\text{cls}} + \mathcal{L}_{\text{loc}}, \tag{1}$$

which are loss of the RPN network, cross-entropy loss for the box classifier and smoothed L_1 loss for the box regressor, respectively.

Few-shot fine-tuning. In this stage, we fine-tune the network based on rare samples. We keep the first few layers unchanged, assign random weights of the new class to the box predictior, and only fine-tune the last layer. We use the same loss function as the previous stage but decreases the learning rate by 20.

Cosine similarity for box classifier. The design of the classifier is based on the cosine similarity function, inspired by Gidaris & Komodakis (2018); Qi et al. (2018); Chen et al. (2019). The weight matrix of the box classifier is $W \in \mathbb{R}^{d \times c}$, where $w_c \in \mathbb{R}^d$ is the per-class weight vector. The output of $\mathcal{C}$ is scaled similarity scores S

$$s_{i,j} = \frac{\alpha \mathcal{F}(x)_i^\top w_j}{\|\mathcal{F}(x)_i\| \|w_j\|}, \tag{2}$$

where $s_{i,j}$ is the similarity calculated between the i-th proposed object and the weight vector of class j. α is a scaling factor. We use a fixed α of 20 and use instance-level feature normalization in our experiments to help diminish the variance.

2.3. Compared Meta-learning with Fine-tuning

Both the meta-learning method and our method are composed of two stages. However, since our fine-tuning method only fine-tunes the last layer of the network, our method is much more memory efficient than the meta-learning method.

3. Experiments

We evaluate our method on two datasets

1. COCO
2. LVIS

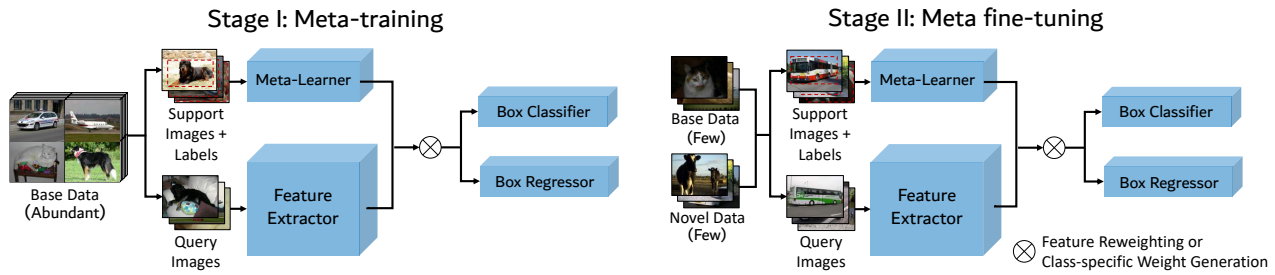

Figure 2. Abstraction of the meta-learning based few-shot object detectors. A meta-learner is introduced to acquire task-level meta information and help the model generalize to novel classes through feature re-weighting (*e.g.*, FSRW and Meta R-CNN) or weight generation (*e.g.*, MetaDet). A two-stage training approach (meta-training and meta fine-tuning) with episodic learning is commonly adopted.

We then provide some quantitative results and visualization results in this section.

Implementation details. We select Faster R-CNN (Ren et al., 2015) as our base detector and use Resnet-101 (He et al., 2016) with a Feature Pyramid Network (Lin et al., 2017) as the backbone of our method. All models are trained using SGD with a mini-batch size of 16, momentum of 0.9, and weight decay of 0.0001. A learning rate of 0.02 is used during base training and 0.001 during few-shot fine-tuning. For hyperparameters related to the model architecture, we use the default parameters provided by Detectron2.

3.1. Existing few-shot object detection benchmark

We evalute the method on COCO dataset with the same data splits and training samples, 60 classes are used as base classes while the other 20 classes are used as novel classes. We compare our method with some of the prior works, `FRCN` is Faster R-CNN for short.

We report the average AP and AP75 of the 20 novel classes on COCO in Table 1. We consistently outperform previous methods across all shots on both novel AP and novel AP75.

Table 1. Few-shot detection performance for the novel classes on the COCO dataset.

Model	novel AP		novel AP75	
	10	30	10	30
FSRW (Kang et al., 2019)	5.6	9.1	4.6	7.6
MetaDet (Wang et al., 2019b)	7.1	11.3	6.1	8.1
FRCN+ft+full (Yan et al., 2019)	6.5	11.1	5.9	10.3
Meta R-CNN (Yan et al., 2019)	8.7	12.4	6.6	10.8
FRCN+ft-full (Our Impl.)	9.2	12.5	9.2	12.0
TFA w/ fc (Ours)	**10.0**	13.4	9.2	13.2
TFA w/ cos (Ours)	**10.0**	**13.7**	**9.3**	**13.4**

3.2. Generalized few-shot object detection benchmark

As mentioned before, we evaluate not only performance on novel classes but also performance on base classes and all classes. That's because the fine-tuning might hurt performance on base classes. Besides, we try to diminish the variance. Therefore, we report AP on base classes (bAP) and the overall AP on the novel classes (nAP).

We evaluate our approach on LVIS dataset (Gupta et al., 2019). We let the common classes be base classes, and the rare classes be novel classes. The results are shown in Table 2. Our method is able to outperform prior methods in not only novel accuracy but also overall accuracy.

3.3. Visualization

We visualize some of the detection results in Figure 3.

4. Conclusion

In this project, we use a two-stage fine-tuning method to conquer the object detection problem, which significantly outperforms prior methods and is shown to be statisticly stable with low variance. In addition, we compared the performance after fine-tuning with the performance before fine-tuning on base classes, and it turns out that the performance on base classes are hardly hurt. Therefore, our method become the state of the art object detection method so far.

ACKNOWLEDGMENTS

This work was supported by Berkeley AI Research, RISE Lab, Berkeley DeepDrive and DARPA.

References

Chen, W.-Y., Liu, Y.-C., Kira, Z., Wang, Y.-C. F., and Huang, J.-B. A closer look at few-shot classification. *arXiv preprint arXiv:1904.04232*, 2019.

Gidaris, S. and Komodakis, N. Dynamic few-shot visual learning without forgetting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4367–4375, 2018.

Gupta, A., Dollar, P., and Girshick, R. Lvis: A dataset for

Table 2. Generalized object detection benchmarks on LVIS dataset.

Method	Backbone	Repeated sampling	AP	AP50	AP75	APs	APm	APl	APr	APc	APf
Joint training (Gupta et al., 2019)	FRCN w/ R-50		19.8	33.6	20.4	17.1	25.9	33.2	2.1	18.5	**28.5**
TFA w/ fc (Ours)			22.6	37.5	22.6	17.5	27.2	36.3	14.2	21.0	26.8
TFA w/ cos (Ours)			22.9	37.5	23.6	19.0	27.5	37.0	15.5	20.8	27.9
Joint training (Gupta et al., 2019)	FRCN w/ R-50	✓	23.1	38.4	24.3	18.1	28.3	36.0	13.0	22.0	28.4
TFA w/ fc (Ours)			24.0	40.0	26.0	19.3	28.9	36.5	15.0	24.1	28.1
TFA w/ cos (Ours)			**24.5**	**40.2**	**26.4**	**20.1**	**29.5**	**38.5**	**17.1**	**24.5**	27.8
Joint training (Gupta et al., 2019)	FRCN w/ R-101		21.9	35.8	23.0	18.8	28.0	36.2	3.0	20.8	**30.8**
TFA w/ fc (Ours)			24.0	39.2	25.1	19.2	29.8	38.8	15.8	22.7	29.1
TFA w/ cos (Ours)			24.5	39.6	26.0	20.3	30.6	39.8	**18.3**	21.5	30.1
Joint training (Gupta et al., 2019)	FRCN w/ R-101	✓	24.7	40.5	26.0	19.0	30.3	38.0	13.4	24.0	30.1
TFA w/ fc (Ours)			25.6	**41.8**	26.8	20.0	31.0	39.3	15.7	26.1	28.3
TFA w/ cos (Ours)			**26.5**	**41.9**	**27.6**	**20.1**	**32.2**	**40.0**	17.2	**26.3**	29.8

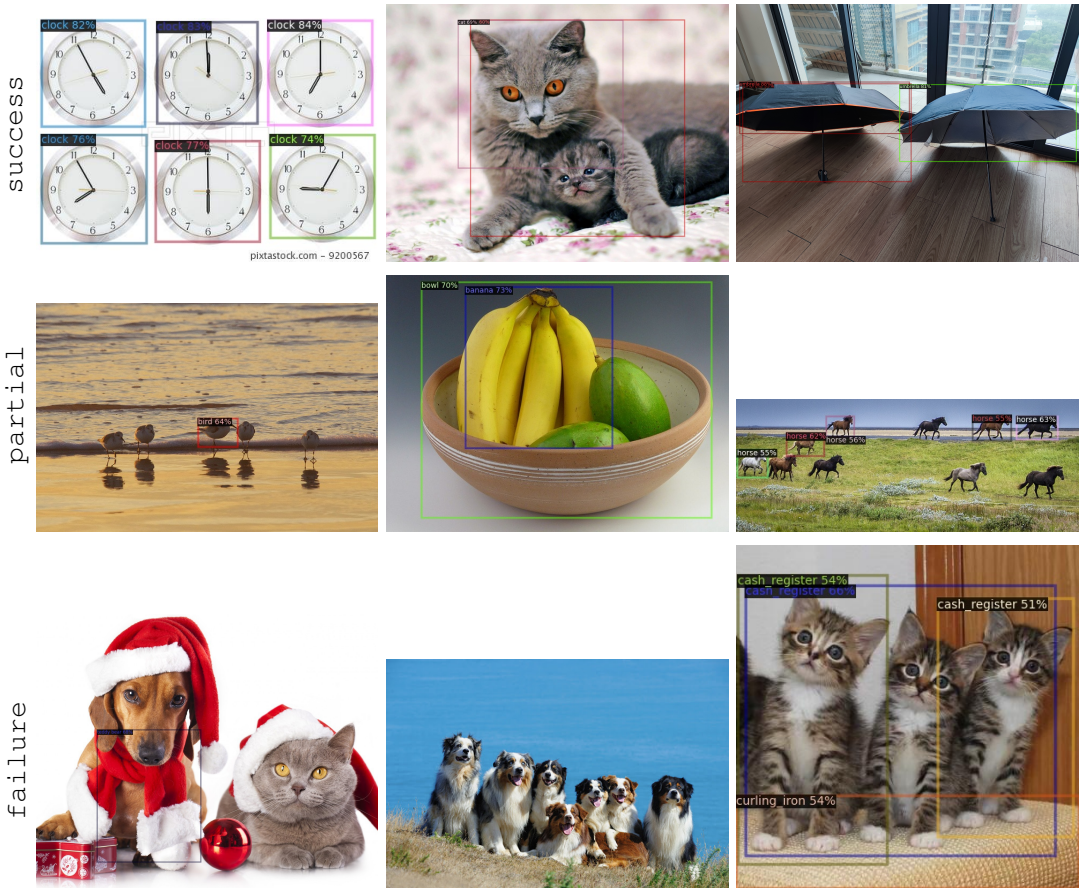

Figure 3. Visualizations of our method. The `success` row means all rare objects are successfully detected, the `partial` row means part of rare objects are successfully detected, and the `failure` row means no rare object is successfully detected.

large vocabulary instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5356–5364, 2019.

Hariharan, B. and Girshick, R. Low-shot visual recognition by shrinking and hallucinating features. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3018–3027, 2017.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Kang, B., Liu, Z., Wang, X., Yu, F., Feng, J., and Darrell, T. Few-shot object detection via feature reweighting. In *ICCV*, 2019.

Lin, T.-Y., Dollar, P., Girshick, R., He, K., Hariharan, B., and Belongie, S. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2117–2125, 2017.

Qi, H., Brown, M., and Lowe, D. G. Low-shot learning with imprinted weights. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5822–5830, 2018.

Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pp. 91–99, 2015.

Simonyan, K. and Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Wang, X., Yu, F., Wang, R., Darrell, T., and Gonzalez, J. E. Tafe-net: Task-aware feature embeddings for low shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1831–1840, 2019a.

Wang, Y.-X., Ramanan, D., and Hebert, M. Meta-learning to detect rare objects. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 9925–9934, 2019b.

Yan, X., Chen, Z., Xu, A., Wang, X., Liang, X., and Lin, L. Meta r-cnn: Towards general solver for instance-level low-shot learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 9577–9586, 2019.