Technical Report for Domain Incremental Object Detection

Jiangtian Zhai, Xialei Liu*
Nankai University

Abstract

Deep learning has achieved great performance in many computer vision applications recent years. However, when facing a new task with different distributions, such as domains with different weather conditions, periods and location scenes, these models usually suffer from catastrophic forgetting without access to previous data. One simple baseline is to train a model from scratch on whole dataset, i.e. *Joint Training, which can cause large computing complexity and storage problems. Continual learning is a research field to greatly reduce the time and space complexity in training deep learning models when facing new domains or new tasks. In this work, we use Experience Balanced Storage for replay on four different domains learning sequentially (a scenario from SSLAD Challenge [1]). We find out that random sampling obtains better results over baselines and we rank the third in the leader board.*

1. Introduction

With the rapid development of deep learning recent years, object detection has improved significantly in precision. However, most works assume a static world, when transferred to a new environment, it usually gives poor performance with the pre-trained network. This problem severely restricted the possibility for neural networks to be used widely in real world tasks, since it is infeasible to train a new model from scratch every time with all data. A simple method is to fine-tune the model directly with the new data, but this can cause catastrophic forgetting problem. Therefore, finding a graceful approach to reach a balance between these two hard problems becomes more and more important.

Traditional object detection task for street scene often pays attention to several key major classes. However, when the model adapts to different weather conditions or periods of a day, the system is required to learn continually from incoming data without forgetting previous knowledge. As for this Challenge, the final objective is to achieve good performance after training sequentially from four different domains with same categories:

- Task 1: Daytime, citystreet and clear weather
- Task 2: Daytime, highway and clear/overcast weather
- Task 3: Night
- Task 4: Daytime, rain

In this technical report, we apply Experience Balanced Storage for replay to address the continual object detection problem. According to the restrictions of this competition, we set the number of samples in memory as 250. When applying this memory storage method, it is an open problem to decide which samples to be stored for further replay. We experiment empirically several methods in this report.

2. Related Work

Object Detection In recent years, object detection task has been dominated by the rapid development of deep learning. There are two major categories of models: one-stage and two-stage. Some custom but powerful ones like Faster-RCNN [4] belongs to two-stage models, which composed of a region proposal network(RPN), a classification head and a regression head. YOLO [3] and SSD [2] are two representatives of one-stage models, generating predictions of bounding box and class probabilities directly from input sample image. As for this competition, we select Faster-RCNN as the base model, which is the provided baseline in the framework code. We use this plain model to explore some other aspects. To find a persuasive solution, we also use the default training hyperparameter to make a fair comparison.

Continual learning Deep learning models suffer from catastrophic forgetting when transferred to some new tasks without access to previous data, and continual learning methods [1] focus on addressing this problem. As for continual object detection learning, Shmelkov et al. [5] considers learning a object detector continually using $L2$ distillation losses to both logits and bounding boxes.

*Xialei Liu is the corresponding author (xialei@nankai.edu.cn).
[1] $https://competitions.codalab.org/competitions/33993$

3. Method

In this section, we will introduce the details of Experience Balanced Storage for replay in our experiments. As for this specific task, we also take a look at the actual data distribution in order to improve the model's continual learning ability.

3.1. Random sampling

There are four domain datasets in total. During the training, after the training of current task, part of its samples are randomly picked to save in a specific memory. According to the competition's restriction, the size is limited to 250 exemplars. We sample randomly to divide the capacity equally among these previous tasks. For instance, after learning on the second task, we save 125 samples for each previous task.

As for training batches, we follow the traditional method to stack current dataset's samples and samples from memory pool. We set the number of samples from two sources to be equal. e.g. If the batch size is 4, there are 2 samples from current datasets and 2 others from the memory pool.

3.2. Distribution-aware sampling

Compared with random sampling in previous datasets, we also observe some features of these datasets. There are 6 classes of object: pedestrian, cyclist, car , truck, tram, tricycle. Similar to most of datasets, these datasets also has the problem of Long-tailed distribution. From validation and test accuracy, we can observe that tricycle is the most rare class in these samples, which greatly damages the performance of the system. Therefore, instead of sampling the memory randomly, we choose to pick samples having the most number of these rare classes. In other words, we sort all samples from each of these previous datasets according to the number of tricycle class bounding boxes in images, and select a fixed memory of these samples.

4. Experiment

We use the avalanche framework [2] based on Pytorch to complete all experiments. To make it convenient and clear, we choose the official implementation with Faster-RCNN from torchvision, fixing the first three layers in ResNet-50 backbone.

As for datasets, we follow the rules in framework code, to keep a split rate of 9:1 between train and validation set. The mean average precision(mAP) over six object classes is used for each scenario, and for each training scenario the total mAP over four tasks is calculated for final assessment.

Training details We the default training implementation. The parameters for Faster-RCNN is set to 4 anchor ratios. The initial learning rate is 1e-3, and we decrease it by 10 times every 3 epochs. We use a batch size of 2 on a single GPU.

Comparison methods We compare Experience Balanced Storage for Replay with baselines. As for details of implementation, we use random sampling and distribution-aware sampling considering tricycle objects.

method	Performance(mAP)	
	val	test
Baseline	55.1	50.6
Distribution-aware sampling	57.1	51.7
Random sampling	59.5	53.1

Table 1. Comparison of Experience Balanced Storage for replay with baselines

From the results we can observe that random sampling strategy gives better final performance. Compared with the distribution-aware sampling, we think the reason is that we use too radical strategy to select samples for memory pool. And we hope to find a more proper way to reach better results in the future.

5. Conclusion

In this technical report, we study the effect of Experience Balanced Storage for replay on continual object detection task in four different scenarios. Compared with baselines, we find that random sampling plays a good start point in memory replay. We also find that our distribution-aware sampling method does not meet the expected performance at the time of competition deadline.

References

[1] Matthias Delange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Greg Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[2] 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.

[3] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pages 779–788, 2016.

[4] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017.

[2] https://github.com/ContinualAI/avalanche

[5] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. Incremental learning of object detectors without catastrophic forgetting. In *ICCV*, pages 3400–3409, 2017.