Strong Baseline for Continual Object Detection
– 2nd Place Solution for CVPR 2022 Workshop on Continual Learning Challenge Track 2&3–

Xingyi Yang[1] Songhua Liu[1] Jingwen Ye[1] Zhou Jun[2] Huang Jia[2] Tan Mingrui[2]
Pargi Mohan Kashyap[2] Tanvi Verma[2] Fei Gao[2] Xinchao Wang[1]
[1] National University of Singapore, Singapore
[2] A*STAR Institute of High Performance Computing, Singapore
xyang@u.nus.edu

Abstract

In this technical report, we present an improved method for sample incremental object detection. We apply the multi-head Faster-RCNN with heuristic data replying to accurately detect objects without forgetting. The proposed approach won 2nd place in CVPR 2022 Workshop on Continual Learning Challenge Detection Track 2&3.

1. Introduction

One fundamental limitation of deep neural networks is their inability to learn new tasks without forgetting previously acquired knowledge. Continual learning thus aims to incrementally refine the model on streaming data. Although it has been receiving tremendous attention to address the problem in a conventional classification setting, very few efforts have been made to study the continual object detection (COD). Until very recently, some works endeavors to learn detectors in the class incremental setup [5, 15]. In contrast to previous efforts, we explore the sample-wise continual learning problem under the scenario of object detection, where new samples are coming sequentially while only a small fraction of previous data are reserved.

There are several special properties that hiders the improvement for sample-wise COD. First, object detection is a hybrid task of classification and regression, where the object category and localization are predicted by one unified network. Typical approaches designed for classification are not well-suited for application on detection tasks. Second, object detection itself poses a class-imbalanced task. Such a phenomenon exacerbates the model performance when training the model on sequential data. Third, for sample-wise continual learning, previously acquired images are no longer available, which is march harder than the class incremental COD.

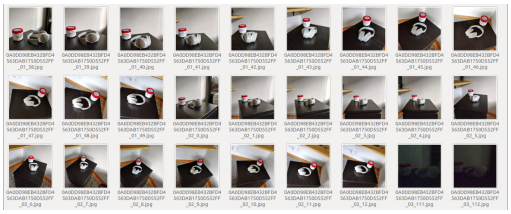
In this report, we present an improved method for sample incremental object detection. We apply the multi-head Faster-RCNN with heuristic data replying to accurately detect objects without forgetting. A large range of model modification and data augmentation strategies is applied to further improve the model performance. The proposed approach won 2nd place in CVPR 2022 Workshop on Continual Learning Challenge Detection Track 2&3.

2. Data Description

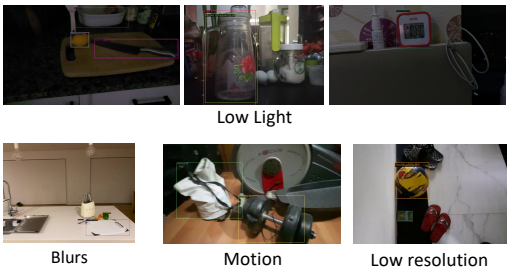
The challenge is supported by the **EgoObjects** dataset provided by Meta, a massive-scale egocentric dataset for objects. EgoObjects is a video dataset created to push the frontier of open-world object understanding from a Continual Learning perspective. We participated in Track 2 and Track 3 of the challenge

Track 2 Category Detection. The "Continual category-level object detection" track contains 70,905 images sampled from 6,076 video clips. There are a total of 277 categories are annotated. The track features a stream of 5 experiences. Each experience is fully annotated. To obtain the stream of training experiences, the ID of the main object is used to obtain a Class-Incremental scenario. However, even surrounding objects will be annotated (even if they do not belong to the same category as the main objects found in each experience). No task labels or other additional signals are provided at test time.

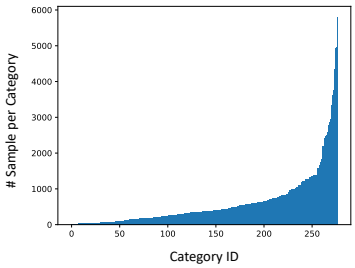
Track 3 Instance Detection. The "Continual instance-level object detection" track contains 70,905 training images sampled from 6,076 video clips. It contains instance-level annotations for 1109 objects. The track features a stream of 5 experiences. Each experience is fully annotated. the ID of the main object is used to obtain a Class-Incremental scenario. Only the main object will be annotated in each image. No task labels or other additional signals are provided at test time.



(a) Image sampled from repeated scene.



(b) Variety of data.



(c) Category imbalancing.

Figure 1. Properties of EgoObjects datasets.

For both Tracks, we are allowed to keep a replay buffer size up to 3500 samples. The detectors are evaluated after training experience, with the parameter size less than 70MB.

After careful inspection through EgoObjects benchmarks, we make the following observations

1. **Image sampled from the repeated scene.** As shown in Figure 1a, because all pictures come from Egocentric video recordings, images in each sequence are highly similar, with multiple views for the identical objects under the same background. The video id is known for the training sets.

2. **Variety of data complexity.** Videos frames are collected from a great variety of lighting conditions, scale, camera motion, and background complexity. Some examples are demonstrated in Figure1b.

3. **Category imbalancing.** We count the number of images from each category in EgoObjects and plot them in Figure 1c. We notice that the dataset poses a long tail pattern in category distribution. The number of objects for the majority class is 1000 times more than the minority class.

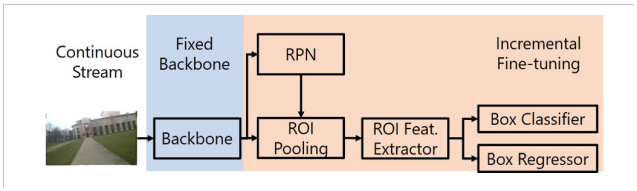


Figure 2. Faster RCNN with fixed and shared backbone and multiple detection head.

3. Solution: Strong Baseline for Continual Object Detection

From the data inspection, we need to design a sample incremental detector that (1) achieves strong performance without forgetting the previous knowledge, (2) performs robust to data quality and variation and (3) does not exceeding the required model size. For track 2, the model should be resistant to the long tail category distribution on the dataset. To cope with all above-mentioned problems, we design a multi-head Faster-RCNN with heuristic data reply strategies. Note that, except the replay strategy, we apply the same methods on both tracks.

3.1. Model Architecture

In general, we train a ResNet-50 [4] FPN [6] Faster RCNN [8] with fixed and shared backbone and multiple detection head [13]. As illustrated in Figure 2, we use a new detector head to deal with the new data at each iteration. Using distinct modules to store the knowledge of prior experiences, largely alleviates the cartographic forgetting problem. The unshared part includes the FPN neck, the RPN head, and the ROI head. Only the ResNet-50 backbone is shared across experiences and kept frozen during training.

According to the challenge, we have $T = 5$ experience therefore our model has 5 heads in total. At the training time for experience t, only the t-th head is updated, while other parameters are untouched. At the t-th evaluation, the predicted bounding boxes of the $1, \ldots, t$-th head are merged through Non-maximum Suppression (NMS). However, the original Faster RCNN with 5 heads reaches the model size of 100MB. We find that most of the parameter comes from the ROI pooling operation, where the $7 \times 7 \times C$ feature map is down-sampled to a C dimension vector with a full-connected layer. We replace the FC layer with an Average Pooling operation, thus shrinking the full model size to 65MB.

To further improve the model performance without adding extra parameters, we resort to two modification of the detector backbone and RPN. Specially, we substitutes

the deformable convolution [16] for vanilla convolution layers and Guided Anchoring RPN (GA-RPN) [12] for static RPN. Deformable convolutions [16] add 2D offsets to the regular grid sampling locations in the standard convolution, which enables an adaptive form deformation of the sampling grid and thus increase the receptive field of the network. Guided Anchoring [12] leverages the semantic features to adaptively predict non-uniform and arbitrary shaped anchors other than dense and Guided Anchoring makes use of semantic information to forecast irregularly shaped anchors as opposed to predetermined anchors on the dense grid.

3.2. Loss Function

For the classification task with long tail distribution in EgoObjects, we apply the Seesaw Loss [14] to dynamically re-balance gradients of positive and negative samples for each category. Given the ordinary d Cross-Entropy (CE) Loss formulation

$$L_{ce}(\mathbf{z}) = \sum_{i=1}^{C} y_i \log(\sigma_i), \quad \text{with } \sigma_i = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}} \tag{1}$$

where $\mathbf{z} = [z_1, z_2, \ldots, z_C]$ and $\sigma = [\sigma_1, \sigma_2, \ldots, \sigma_C]$ represents the predicted logits and probability of the classifier. And $y_i \in \{0, 1\}$ is the one hot class label. To mitigate the influence of the dominant categories, the Seesaw Loss applies a Mitigation Factor $\mathcal{M}_{ij}$ and a Compensation Factor $\mathcal{C}_{ij}$ as re-weighting factors

$$\hat{\sigma}_i = \frac{e^{z_i}}{\sum_{j \neq i}^{C} \mathcal{M}_{ij} \mathcal{C}_{ij} e^{z_j} + e^{z_i}} \tag{2}$$

Mitigation Factor $\mathcal{M}_{ij}$ accounts for the instance number N_i for each category

$$\mathcal{M}_{ij} = \begin{cases} 1, & \text{if } N_i \leq N_j \\ (\frac{N_j}{N_i})^p, & \text{otherwise} \end{cases} \tag{3}$$

Another compensation factor focuses on misclassified samples instead of adjusting the whole category.

$$\mathcal{C}_{ij} = \begin{cases} 1, & \text{if } \sigma_j \leq \sigma_i \\ (\frac{\sigma_j}{\sigma_i})^q, & \text{otherwise} \end{cases} \tag{4}$$

Both p, q are hyperparameters to control the re-weighting scale.

For the localization branch, we utilize the generalized IoU (GIoU) [9] loss to accurately regress the bounding boxes coordinates.

3.3. Data Augmentation and Sampling

As we noticed, the EgoObjects contains images with a large variety in lighting conditions, scale, camera motion, and background complexity. These data properties allow you to apply strong augmentation with multi-scale training to improve the model generalization ability at test time. The applied augmentation is listed in Listing 1.

```
train_transforms = [
    dict(
        type='ShiftScaleRotate',
        shift_limit=0.0625,
        scale_limit=0.0,
        rotate_limit=0,
        interpolation=1,
        p=0.5),
    dict(
        type='RandomBrightnessContrast',
        brightness_limit=[0.1, 0.3],
        contrast_limit=[0.1, 0.3],
        p=0.3),
    dict(
        type='OneOf',
        transforms=[
            dict(
                type='RGBShift',
                r_shift_limit=10,
                g_shift_limit=10,
                b_shift_limit=10,
                p=1.0),
            dict(
                type='HueSaturationValue',
                hue_shift_limit=20,
                sat_shift_limit=30,
                val_shift_limit=20,
                p=1.0)
        ],
        p=0.2),
    dict(type='JpegCompression', quality_lower=75, quality_upper=95, p=0.2),
    dict(type='ChannelShuffle', p=0.1),
    dict(
        type='OneOf',
        transforms=[
            dict(type='Blur', blur_limit=3, p=1.0),
            dict(type='MedianBlur', blur_limit=3, p=1.0),
            dict(type='MotionBlur', p=1.0),
            dict(type='ISONoise', p=1.0)
        ],
        p=0.25),
]
```

Listing 1. Training Augmentation.

Apart from the augmentation, we address the class imbalance issue with the Class-Aware Sampler [10]. A Class-aware sampling strategy equalizes the sample number in each mini-batch according to the category, thus effectively tackling the non-uniform class distribution.

3.4. Reply Strategy

We design two replay strategies according to our inspection of the dataset. We empirically find that both methods work well on the challenge.

Minority Replay. For the Category detection track, we use samples from the minority class as our replay buffer. At the beginning of each experience, we count the instance number for each category in the training sample and previous relay buffer. We sort the class ID by its instance counts, then pick the top 3500 images from the minority class into the pool for the next iteration.

Representative Replay. For the Instance Detection Track, we randomly sample one frame from each video sequence to make sure the replay samples are a good representative of the full data distribution. The replay buffer is initialized to an empty set. When samples from a new experience come, we re-arrange the images by their video ID. If the total number of video ID K is greater than or equal to 3500, $K \geq 3500$, we randomly select 3500 videos and pick one random frame from each clip. If the total number of video

Table 1. Performance On Category Detection Track Evaluation.

	EXP1	EXP2	EXP3	EXP4	EXP5	Mean
mAP	0.284	0.447	0.576	0.679	0.782	**0.554**
AP_{50}	0.359	0.556	0.705	0.822	0.940	**0.676**
AP_{75}	0.313	0.495	0.638	0.752	0.867	**0.613**

Table 2. Performance On Instance Detection Track Evaluation.

	EXP1	EXP2	EXP3	EXP4	EXP5	Mean
mAP	0.150	0.304	0.455	0.608	0.754	**0.454**
AP50	0.186	0.374	0.560	0.749	0.932	**0.560**
AP75	0.170	0.345	0.517	0.695	0.861	**0.518**

IDs K is less than 3500, $K < 3500$, we repeatedly sample from each video sequence until the buffer size reaches 3500.

4. Experiments and Results

Implementation Details. We optimize the detector using AdamW [7] optimizer, alongside an initial learning rate of 0.0001 and weight decay of 0.05. The batch size is set to 16. For each experience, we train the model for 12 epochs. The detectors are initialized with LVIS [3] pre-trained weights. We apply DropBlock [2] with $p = 0.1$ at the CONV3 of ResNet-50 LAYER3 and LAYER4 as a regularization technique to further improve the results. The exponential moving average (EMA) [11] of the parameters are used during evaluation. All models are implemented with MMDetection [1] and are trained on 8 RTX 3090 GPU.

Results on EgoObjects. The final evaluation results on the CodaLab [1] [2] for both tracks are shown in Table 1 and Table 2. We achieve superior detection performance and won second place on category detection tracks. It can be noticed that the mAP, AP_{50}, AP_{75} consistently increases as new experience comes. It illustrates that our multi-head strategy can learn to detect new objects without forgetting seen instances.

Ablation Study. As we include tons of advanced techniques to build our final detector, we would like to investigate whether they can improve the final performance collaboratively. In Figure 3, from the very baseline of a simple Faster RCNN, we sequentially add one module in the pipeline and evaluate it on the Category detection track. The x-axis is the method name and the y-axis is the mAP on the test set. We see that all introduced techniques can boost the detector mAP cooperatively. It supports that our proposed framework serves as a strong baseline for sample COD.

[1]https://codalab.lisn.upsaclay.fr/competitions/3568
[2]https://codalab.lisn.upsaclay.fr/competitions/3569

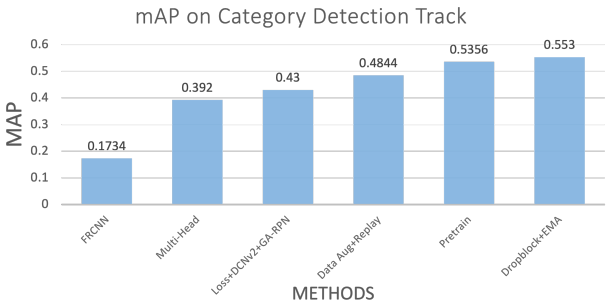


Figure 3. Ablation Study on Category Detection Track.

5. Conclusion

In this technical report, we present an improved method for sample incremental object detection. We apply the multi-head Faster-RCNN with heuristic data replying to accurately detect objects without forgetting. The proposed approach won 2nd place in CVPR 2022 Workshop on Continual Learning Challenge Detection Track 2&3.

References

[1] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019. 4

[2] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Dropblock: A regularization method for convolutional networks. *Advances in neural information processing systems*, 31, 2018. 4

[3] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 4

[4] 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. 2

[5] KJ Joseph, Salman Khan, Fahad Shahbaz Khan, and Vineeth N Balasubramanian. Towards open world object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5830–5840, 2021. 1

[6] 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. 2

[7] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 4

[8] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region

proposal networks. *Advances in neural information processing systems*, 28, 2015. 2

[9] Hamid Rezatofighi, Nathan Tsoi, JunYoung Gwak, Amir Sadeghian, Ian Reid, and Silvio Savarese. Generalized intersection over union: A metric and a loss for bounding box regression. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 658–666, 2019. 3

[10] Li Shen, Zhouchen Lin, and Qingming Huang. Relay backpropagation for effective learning of deep convolutional neural networks. In *European conference on computer vision*, pages 467–482. Springer, 2016. 3

[11] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017. 4

[12] Jiaqi Wang, Kai Chen, Shuo Yang, Chen Change Loy, and Dahua Lin. Region proposal by guided anchoring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2965–2974, 2019. 3

[13] Jianren Wang, Xin Wang, Yue Shang-Guan, and Abhinav Gupta. Wanderlust: Online continual object detection in the real world. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10829–10838, 2021. 2

[14] Jiaqi Wang, Wenwei Zhang, Yuhang Zang, Yuhang Cao, Jiangmiao Pang, Tao Gong, Kai Chen, Ziwei Liu, Chen Change Loy, and Dahua Lin. Seesaw loss for long-tailed instance segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9695–9704, 2021. 3

[15] Wang Zhou, Shiyu Chang, Norma Sosa, Hendrik Hamann, and David Cox. Lifelong object detection. *arXiv preprint arXiv:2009.01129*, 2020. 1

[16] Xizhou Zhu, Han Hu, Stephen Lin, and Jifeng Dai. Deformable convnets v2: More deformable, better results. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9308–9316, 2019. 3