3^{rd} Place Solution for the CLVISION Challenge 2022 - Continual Instance Detection (Track 3)

Angelo Menezes
University of São Paulo
angelomenezes@usp.br

Abstract

In this technical report, we present our approaches for the Track 3 (Continual Instance Object Detection) of the 3^{rd} CLVISION Challenge. Using a combination of knowledge distillation and balanced replay, our team secured the third position out of all the participants in the challenge.

1. Introduction

The task of training object detectors in an incremental manner is of importance for several real-world applications such as robotics and autonomous vehicles [10]. Despite its importance, most of the continual learning solutions for computer vision have only approached classification scenarios. For a thorough overview of the topic, explaining why continual object detection is a more challenging task, we point the readers to a current review on the subject [8].

The 3^{rd} CLVISION Challenge proposed the exploration of a massive-scale egocentric dataset for objects from a continual learning perspective in order to encourage research on the frontiers of egocentric perception.

2. Dataset and Challenge Constraints

The dataset used for this challenge is an adaptation of the Ego4D dataset [2], released by Meta, for the incremental setting called EgoObjects. The dataset features first-person videos of people handling objects in their daily lives. The videos were broken into frames and split into training and testing sets containing no data leakages. Through their API, the training set could also be easily split into training and validation. An example of the images present on the dataset is shown in Figure 1.

The 3^{rd} CLVISION Challenge presented 3 possible tracks involving continual learning for object recognition:

- **Track 1**: Continual instance-level object classification.
- **Track 2**: Continual category-level object detection.

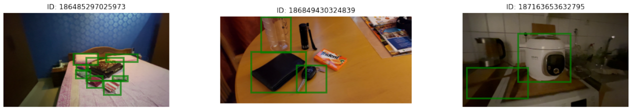

Figure 1. Samples from the EgoObjects dataset.

- **Track 3**: Continual instance-level object detection.

Specifically for Track 3, which is the focus of this report, the model needed to handle five learning experiences containing a total of 1111 different objects. The final solutions were constrained to use only pre-trained methods on the ImageNet-1K, COCO, or LVIS datasets. Other constraints related to the solution were:

1. **Max model size**: 70M parameters.
2. **Max replay buffer size**: 5000 samples.
3. The model needs to finish its **training and evaluation under 24 hours** on the reference server [1].
4. No test time training or augmentation.
5. The solution must not use information regarding the category of instances nor the video ID at test time.

3. Methodology

Considering the aforementioned data and constraints, in this section we describe the main components of our solution. The Avalanche framework [7] was used to load the challenge data and structure the continual learning plugins.

3.1. Architecture and training settings

For the neural network architecture, we chose the Fully Convolutional One-Stage Object Detector (FCOS) [13] using a ResNet50 with FPN as backbone from torchvision[2]. The model was pretrained on the COCO train2017

[1] The reference server was an AMD EPYC 7282, 128 GB RAM @ 2666 MHz with SSD and an Nvidia Quadro RTX 5000.

[2] https://pytorch.org/vision/0.12/models.html

dataset [4] and had only its head changed to be able to detect all the 1111 possible objects.

For the optimizer and learning rate (LR) across experiences, we applied SGD with a 0.05 LR and momentum of 0.9. The LR scheduler was set to be linear with a warmup of 1000 iterations, as in the general template given for the competition. The scheduler was applied only to the first epoch of the first experience and kept the LR stable across the whole training execution. Although this scheduler was chosen for our final solution, some preliminary tests using a StepLR starting at 0.01, decreasing by 50% every 10,000 steps, and restarting at each experience also showed a decent performance. Yet, we did not have time to explore diverse solutions based on this last setup.

Other general training settings were:

- **N° of Epochs:** 5.
- **Image size**: images were limited to 800 on the largest size and 600 on the shortest.
- **Training batch size**: 4.
- **Validation batch size**: 16.
- **Transforms**: Random Horizontal flip with 50% of probability.

3.2. Balanced Experience Replay

Considering the maximum buffer size and that the number of images for each object instance is highly unbalanced (e.g., some with more than 100 instances, others with 15), we proposed the use of a balanced replay buffer.

After the first experience, the replay buffer is initialized so that at least N samples from each class for the previous experience are present for the next task. In this case, N is the buffer size available for the task divided by the number of different object instances in it. The buffer size is defined by the maximum buffer size divided by the number of tasks seen so far.

Using such a strategy instead of an experience replay buffer based on reservoir sampling resulted in an increase from 34.6 to 40.8 on the final leaderboard metric (average AP).

3.3. Knowledge Distillation from Features and Outputs

Following the basic distillation procedure for incremental object detection introduced by Shmelkov *et al*. [11] and the following advances proposed by Chen *et al*. [1], we regularized the learning of each new experience by distilling knowledge from a saved version of the model trained on the previous experiences. The distilled knowledge comes from the L_2 loss, here named penalty, applied to the head (logits, bounding boxes and "center-ness" of objects) and intermediate features (layer2.3.relu, layer3.3.relu and layer4.2.relu) for both models as described by Equations 1 and 2.

$$L_{head} = \frac{1}{3}\sum_{j}\frac{1}{M}\sum_{i=1}^{M}||y_j^{teacher}(x_i) - y_j^{student}(x_i)||^2 \tag{1}$$

$$L_{feat} = \frac{1}{3}\sum_{k}\frac{1}{M}\sum_{i=1}^{M}||F_k^{teacher}(x_i) - F_k^{student}(x_i)||^2 \tag{2}$$

where y is the output of a head j, F are the feature activations from a layer k and x_i is a sample from a batch M for the current experience. The final penalty is calculated by adjusting the weight for the L_{feat}, as shown in Equation 3 since the losses were on different scales. We found that a λ of 10 was able to balance the contribution of the two terms.

$$L_{penalty} = \lambda\, L_{feat} + L_{head} \tag{3}$$

The final loss used to update the weights was the original L_{FCOS}, obtained when training the student detector, summed by the scaled penalty value as described in Equation 4.

$$Loss = L_{FCOS} + \alpha\, L_{penalty} \tag{4}$$

α is a parameter to calibrate the current model‘s stability-plasticity considering the previous one. Most distillation solutions in continual object detection use the value of 1 as reference [1,9], but we found that the value of 0.5 had better performance for the final validation metric. This distillation setting resulted in an increase from 40.8 to 41.7 average AP, as shown by our final performance on the leaderboard.

3.4. Failed attempts to improve the results

We evaluated several strategies to improve the incremental detector's final performance. Due to the considerable computational time required when training large object detectors, most of the experiments were executed for 5 epochs to briefly check their effectiveness. Some of the attempts are described below:

- **Adam optimizer**: The optimizer presented good performance but was less stable than SGD for higher LR and thus had worse validation metrics for 5 epochs.
- **Learning Rate Finder + One Cycle Policy [12]**: Strategy presented good learning performance but had lower validation metrics than plain SGD with LinearLR and warmup iterations.

- **Larger images**: Training with larger images (Max 1333 x Min 800) showed better validation results but impacted the total used memory and training time. Thus, we kept the smaller setup to comply with the evaluation server memory and time constraints.
- **More epochs**: Considering the increase in computational time brought by the distillation component, we limited the training to 5 epochs to comply with the constraints mentioned above. However, some initial tests pointed out that training for more than 5 epochs would not result in any considerable gains in performance. This might also be related to using a LinearLR that does not change across all the experiences.
- **Different Augmentations**: We applied some "stronger" augmentations such as random distortions and IOU crops, but the final validation metrics were affected negatively.

4. Final Considerations

In this short report, we described our strategies for the continual instance detection problem proposed by the 3^{rd} CLVISION Challenge. As shown by our ablation studies, the catastrophic forgetting effects when training incrementally with the EgoObjects dataset were mitigated. For that, the best setting happened when using an experience replay buffer balanced by the number of different categories in each experience along with a knowledge distillation component applied to the features and head outputs.

Considering the possible future directions for the ones who want to develop solutions to the same benchmarks, we believe that strategies using more advanced architectures, backbones, and losses such as the VarifocalNet [14] and Swin [6] can be promising mainly if some of the initial constraints can be relaxed (e.g., number of parameters and training time). Also, the exploration of self-labeling solutions [3] and other regularization methods, such as EWC adapted to object detection [5], should be considered.

References

[1] Li Chen, Chunyan Yu, and Lvcai Chen. A new knowledge distillation for incremental object detection. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2019. 2

[2] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, et al. Ego4d: Around the world in 3,000 hours of egocentric video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18995–19012, 2022. 1

[3] Linting Guan, Yan Wu, Junqiao Zhao, and Chen Ye. Learn to detect objects incrementally. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 403–408. IEEE, 2018. 3

[4] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 2

[5] Liyang Liu, Zhanghui Kuang, Yimin Chen, Jing-Hao Xue, Wenming Yang, and Wayne Zhang. Incdet: In defense of elastic weight consolidation for incremental object detection. *IEEE transactions on neural networks and learning systems*, 32(6):2306–2319, 2020. 3

[6] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 10012–10022, 2021. 3

[7] Vincenzo Lomonaco, Lorenzo Pellegrini, Andrea Cossu, Antonio Carta, Gabriele Graffieti, Tyler L Hayes, Matthias De Lange, Marc Masana, Jary Pomponi, Gido M Van de Ven, et al. Avalanche: an end-to-end library for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3600–3610, 2021. 1

[8] Angelo G Menezes, Gustavo de Moura, Cézanne Alves, and André CPLF de Carvalho. Continual object detection: A review of definitions, strategies, and challenges. *arXiv preprint arXiv:2205.15445*, 2022. 1

[9] Can Peng, Kun Zhao, and Brian C Lovell. Faster ilod: Incremental learning for object detectors based on faster rcnn. *Pattern recognition letters*, 140:109–115, 2020. 2

[10] Khadija Shaheen, Muhammad Abdullah Hanif, Osman Hasan, and Muhammad Shafique. Continual learning for real-world autonomous systems: Algorithms, challenges and frameworks. *Journal of Intelligent & Robotic Systems*, 105(1):1–32, 2022. 1

[11] Konstantin Shmelkov, Cordelia Schmid, and Karteek Alahari. Incremental learning of object detectors without catastrophic forgetting. In *Proceedings of the IEEE international conference on computer vision*, pages 3400–3409, 2017. 2

[12] Leslie N Smith. Cyclical learning rates for training neural networks. In *2017 IEEE winter conference on applications of computer vision (WACV)*, pages 464–472. IEEE, 2017. 2

[13] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9627–9636, 2019. 1

[14] Haoyang Zhang, Ying Wang, Feras Dayoub, and Niko Sunderhauf. Varifocalnet: An iou-aware dense object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8514–8523, 2021. 3