Continual Learning: 3rd CLVISION CVPR Workshop Task 1 Instance Classification

Anonymous CVPR submission

Abstract

Continual learning is a branch of deep learning that aims to mitigate catastrophic forgetting, but still allow the model to change and adapt to new data it is being trained on. For the first challenge track of the CVPR CLVISION workshop is focused on class incremental learning which is learning new classes in separate phases. In each phase only the samples of the new classes can be used, and the samples of previously used data is unavailable. For this challenge track there is a template provided with several commonly used strategies for continual learning. Thus, the purpose of this project is to test how effective each strategy is using the data set provided.

1. Introduction

The first challenge track for CLVISION is an instance classification track. In this track the focus on how to mitigate catastrophic forgetting while going through a series of learning different classifications. The challenge tracks also comes with a devkit that is provided that also contains commonly used continual learning strategies that are readily available as a plugin [1]. These plugins are from a library called Avalanche which is an End-to-End Continual Learning Library Based on PyTorch [1]. So, for my project the goal will be to find the best strategy or strategies available for the dataset that is provided. Finding strategies to mitigate catastrophic forgetting will allow models to be able to run and learn in more dynamic environments. Which allows more applications of deep learning models in the real world. This is challenging because the current models for deep learning usually do training and testing in two different static phases which is hard to transfer to a dynamic environment. Thus, improving the effectiveness of strategies in continual learning will be beneficial for many areas both research and real-world applications.

1.1. Instance Classification Challenge Track.

The Continual Instance Classification challenge track is of course a classification task. The challenge is structured as a Class Incremental model in which there are multiple phases of training and testing. There will be 15 of these experiences and a total of 1100 categories and 187 classes which is about 74 categories per experience and 12 classes per experience. The inputs must be loaded as 224x224 RBG images and the outputs are the class. This is a fully supervised task so there are labels provided at training time. To measure the accuracy of the outputs we will be using the average mean class accuracy which is for each pass on the test step the mean class accuracy is calculated and then the 15 results are averaged to get the final result.

1.2. Dataset

The dataset is from is called The EgoObjects Dataset which is a video dataset which is egocentric [2]. Then from there the videos are cropped to get an image of the object in the dataset. In some cases, there are multiple instances of an object within an image. In this case we only must identify they type of objects and not how many there are. Images come in varying sizes, but the most common image size by far is 1920x1080. The total number of images in the data set totals to 84 620 images. The dataset is divided into two categories the training category and the testing category. The training category is divided further into the actual images used to train and the validation dataset, but these are determined by the seed that comes with the devkit. The testing set does not have any labels and does not tell you which category the image belongs in. There are similar datasets that have short video sessions like COREe50[3] and OpenLORIS[4]

1.3. Sample Images

Figure 1. Scarf [2]

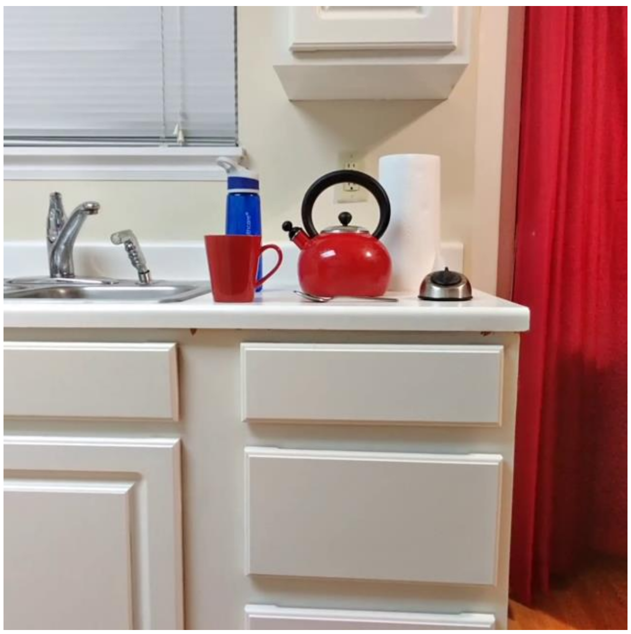

Figure 2. Teapot [2]

Figure 3. Can opener [2]

.

2. Visualize your labels

This is only for the training part of the dataset because the test set does not have any labels for any of the objects. For the labels I created a histogram with 50 bins that measures the number of classes that have a certain number of images. I also computed the mean, standard deviation and median for the label distribution.

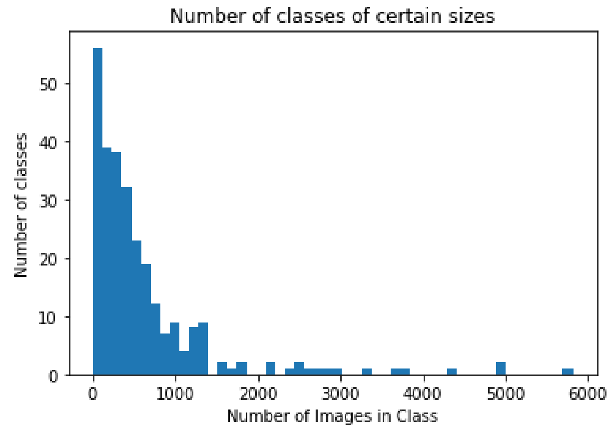

Figure 4: Histogram of the number of images in a class showing the distribution of the images

Table 1. Label distribution

Statistic	Value
Mean	618.41
Standard Deviation	833.71
Median	371

The are some class imbalances within the dataset. Most of them have less than 100

3. Visualize Images

For visualizing images I plotted the image size distribution on a bar graph and plotted the mean intensity of each image and the mean intensity of each image class

3.1 Image Size distribution

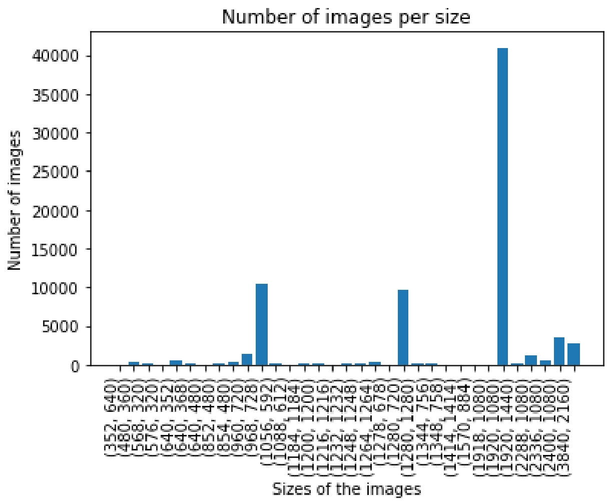

Figure 5: Bar graph showing the distribution of image sizes in the dataset

3.2. Mean intensity

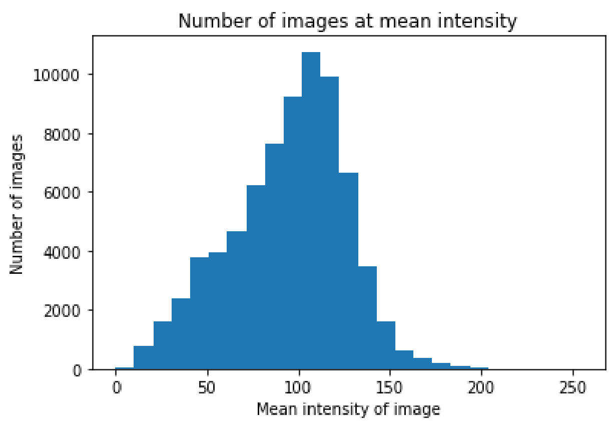

Figure 6: Histogram with 25 bins showing the distribution of the mean pixel intensity per image

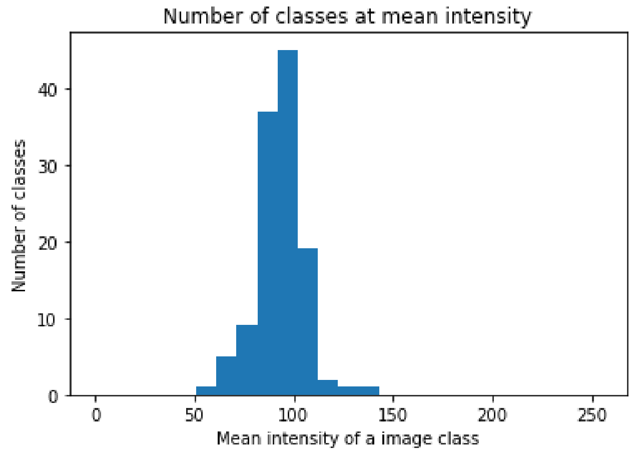

Figure 7: Histogram with 25 bins showing the distribution of the mean pixel intensity per class

4. Subsets

There is a smaller dataset which is the demo dataset which was provided before the full dataset was available and that is going to be the larger subset because 84 thousand images take a while to train the model on. It has the same number of labels and groups of data. The test set does have class labels and has labels for which subcategory the image is in. Although some parts of the dataset are inconsistent with other parts of the dataset. Like the number of images that it says are in the dataset and how many there actually are 2000 difference for a dataset of 6000 images. Using the demo dataset, I have created a smaller dataset with only 15 classes. The reason I choose 15 classes is that for when training there should be 1 class being trained at a time which is hypothetically more like a real-world application of continual learning. To get less than 1000 images for the small subset I filtered out any of the large classes with more than 100 images while keeping a similar amount of class imbalance.

4.1 Demo dataset figures

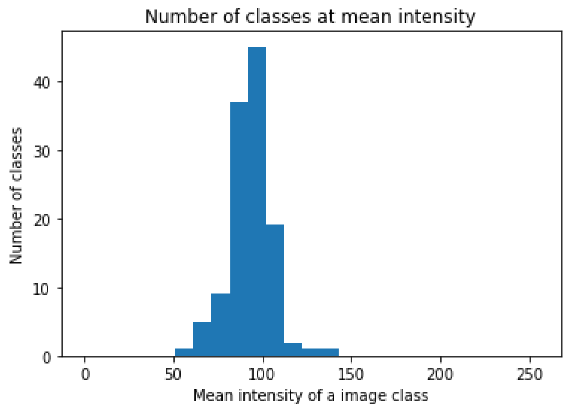

Figure 8: Histogram of the number of images in a class showing the distribution of the images

Table 2. Label distribution

Statistic	Value
Mean	136.40
Standard Deviation	112.56
Median	95.5

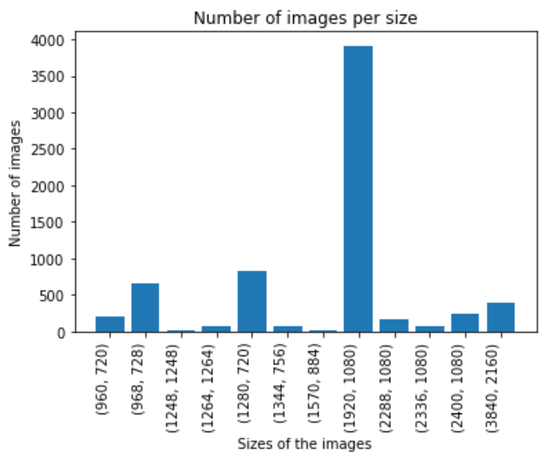

Figure 9: Bar graph showing the distribution of image sizes in the dataset

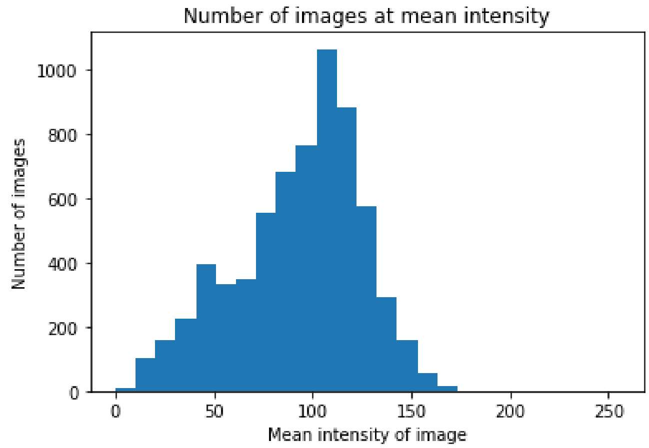

Figure 10: Histogram with 25 bins showing the distribution of the mean pixel intensity per image

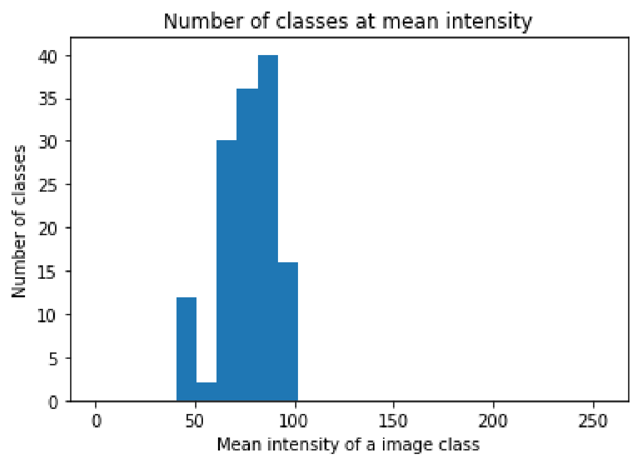

Figure 11: Histogram with 25 bins showing the distribution of the mean pixel intensity per class

References

[1] Lomonaco, V., Pellegrini, L., Cossu, A., Carta, A., Graffieti, G., Hayes, T. L., ... & Maltoni, D. (2021). Avalanche: an end-to-end library for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 3600-3610).

[2] Pellegrini, L., Zhu, C., Xiao, F., Yan, Z., Carta, A., De Lange, M., ... & Vazquez, D. (2022). 3rd Continual Learning Workshop Challenge on Egocentric Category and Instance Level Object Understanding. *arXiv preprint arXiv:2212.06833*.

[3] Lomonaco, V., & Maltoni, D. (2017, October). Core50: a new dataset and benchmark for continuous object recognition. In *Conference on Robot Learning* (pp. 17-26). PMLR. https://vlomonaco.github.io/core50/

[4] She, Q., Feng, F., Hao, X., Yang, Q., Lan, C., Lomonaco, V., ... & Chan, R. H. (2020, May). Openloris-object: A robotic vision dataset and benchmark for lifelong deep learning. In *2020 IEEE international conference on*

robotics and automation (ICRA) (pp. 4767-4773). IEEE. https://lifelong-robotic-vision.github.io/dataset/object