Open World Recognition in Image Classification

Gabriele Bruno Franco s282245
Anna Di Lorenzo s287952
Shannon Mc Mahon s289958

Abstract

Despite the recent success of neural networks, their capability is frequently limited to closed world scenarios in which it is assumed that the semantic concepts a model has to recognize is limited to the number of classes seen during training. Several works have investigated the scenario known as Open World Recognition (OWR) in order to break these limiting assumptions. In the aforementioned framework, the model must be able to: i) learn new classes incrementally; ii) recognize when classes do or do not belong to the knowledge it already has (open set); and iii) add these classes to its knowledge once data for these categories is provided.
In this project we implement and study the knowledge distillation strategy to address incremental learning challenges, and subsequently incorporate rejection capability into the models. The code for the project is available on Github [1].

1. Introduction

As humans, our vision systems are inherently incremental. This means that when new visual information is incorporated existing knowledge is preserved. Consider a child familiar with the objects spoon and fork. If it sees chopsticks, it will retain this new knowledge without forgetting the previously learnt objects. The same cannot be said for vision-based applications [7]. These systems are trained in batch settings, in which all objects are known in advance. The objective moving forward is to find more flexible strategies that are able to incrementally learn new classes. In other words, as soon as new data becomes available for a class we want to be able to learn from it incrementally and avoid retraining the model from scratch, which can be computationally unfeasible. This scenario is known as class-incremental.
One possible technique consists in training classifiers from class-incremental data streams, e.g. using stochastic gradient descent optimization. Unfortunately this approach causes a quick deterioration of the classification accuracy. This effect is known as catastrophic forgetting. [9]. In practical terms new learning may alter weights involved in representing old learning, which in turn leads to inappreciable results.
In this paper we firstly reproduce the results of *ICaRL* [10], a proposed technique that allows the learning of new classes incrementally. Subsequently, having compared this method with other existing ones, we move to the *Open World scenario* in which we try to recognize whether or not a given sample belongs to the previously acquired knowledge, and, when appropriate, how to add classes to the knowledge of the system as soon as data for these categories is provided. Finally, we propose a possible modification to the defined model.

[1] https://github.com/gbrunofranco/MLDL

2. iCaRL

In the paper *iCaRL: Incremental Classifier and Representation Learning*, S.A. Rebuffi *et al.* address Catastrophic forgetting by use of the following:

- *Augmented Loss Function*: it combines a standard classification loss with a distillation loss. The first encourages improvements of the feature representation which results in good classification of new classes, while the second is a regularization term used to avoid the loss of previously learnt information;
- *Augmented Training set*: consists of training data and stored Exemplars. The latter are a representative set of samples from the distribution for each seen class. Their presence ensures that at least some information of the previous classes enters the training process;
- *Herding*: The number of exemplars per class must decreases as the number of considered classes grows (since the total number of exemplars remains constant throughout the process and equal to 2000), so a removal strategy of exemplars from the various exemplars sets is needed. To address this, iCaRL makes use of a prioritized construction of exemplars sets: exemplars are added to the set in an orderly manner, based on how good of an approximation it is of the mean vector for the given class. This way, to obtain the new

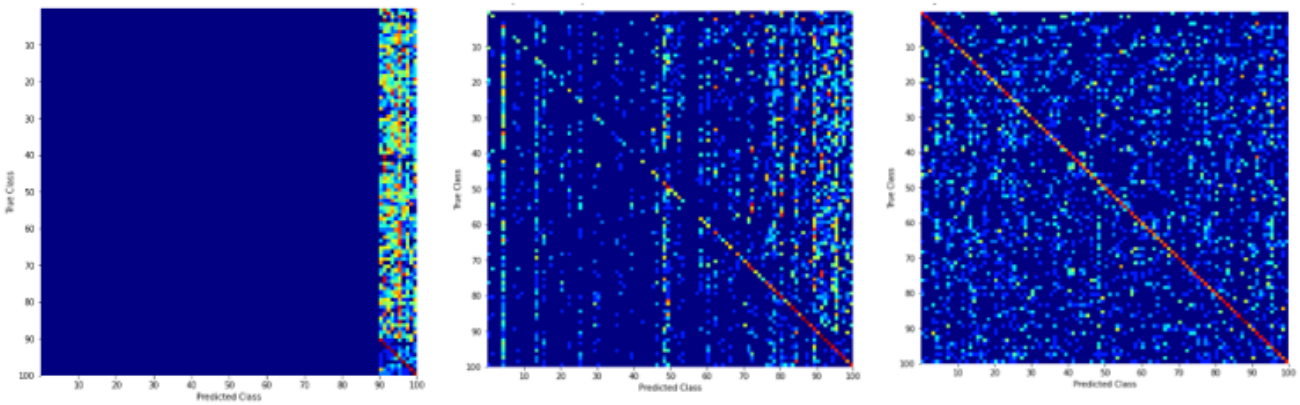

Figure 1. Confusion matrices for the three seen methods. From left to right: Fine-Tuning, Lwf, and iCaRL.

(smaller) exemplar sets we just need remove elements in fixed order starting from the end of the sets.

- *NME Classifier*: Nearest Mean of Exemplars multi-class classifier. Each class has a prototype vector which consists of the average feature vector of the exemplars of such class. Given an unlabeled image, it is assigned the label of the class that minimises the distance between the feature vector of the image and the prototype vector of a class. In mathematical terms, we have:

$$y* = \underset{y=1,\ldots,t}{\operatorname{argmin}} \|f(x) - \mu_y\|_2 \quad (1)$$

where μ_y is the average feature vector of class y calculated among the exemplars, x is the image, $f(x)$ the feature vector and $y*$ the class assigned to x.

3. iCaRL, Finetuning and LwF

This section is dedicated to replicating some of the results present in the original paper. The experiments are run on the CIFAR-100 dataset, using a training set of 10 by 10 class scenarios. Specifically the dataset is randomly divided in 10 sets of 10 classes, and each set is learned on a different training step. The testing set instead includes all the classes seen in the current learning step and also the previous training steps (e.g. after step 3 we test on the 30 known classes). As seen in the original implementation, data augmentation is performed by means of random horizontal splits and random cropping. A 32-layer ResNet [6] is implemented, with a learning rate of 2, which is divided by a factor of 5 after 49 and 63 epochs. The mini batch size is set to 128, the weight decay to *1e-4* and the maximum memory to 2000 (which corresponds to the maximum number of exemplars).
Figure 1 shows the confusion matrices for iCaRL and two other approaches: Finetuning and Learning Without Forgetting (LWF). Note that the entries were transformed by means of the *log(1+x)* function to emphasise the results.

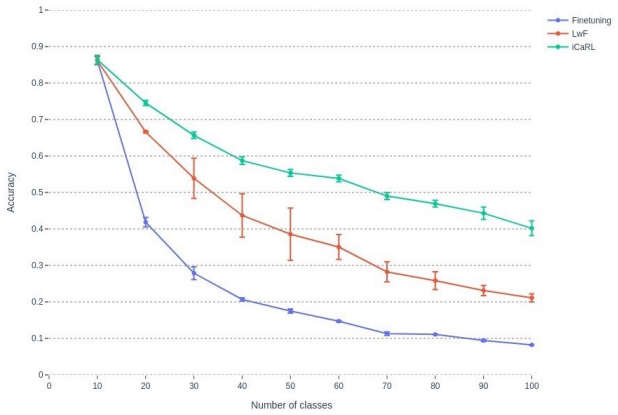

Figure 2. Accuracy plot for Fine-Tuning, LwF and iCaRL. Averages and standard deviations for each method were calculated by averaging the results obtained with 3 different seeds (the same 3 seeds were used for Finetutning, LwF, iCaRL so as to have comparable results).

- *Finetuning* (left) learns ordinary multi-class network without taking any measures to mitigate catastrophic forgetting. We use Binary Cross Entropy (BCE) as classification loss and the FC Layer (fully connected of the network) as classifier. We can see that all predicted class labels come from the last batch of classes that the network has been trained on;

- *Learning without forgetting* (middle) has a similar philosophy to that of iCaRL, as it makes use of an additional Distillation Loss term. Thus we have a BCE classification loss, a BCE distillation loss and FC Layer for classification purposes. However, LwF does not make use of the Exemplar Sets. The result is a confusion matrix with many more non zero entries towards the right (so for more recently learnt classes).

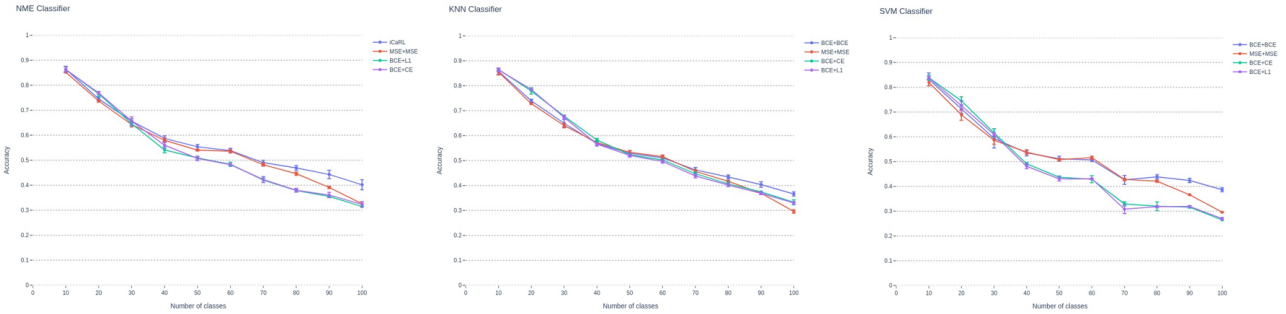

Figure 3. Accuracy plots for, from left to right, NME, KNN and SVM classifiers with various loss combinations.

- *iCaRL* (right) we see a homogeneous behaviour over all classes, both in terms of correct predictions (diagonal entries) and mistakes (non diagonal entries). This is due to the absence of intrinsic bias towards classes encountered early on or late in the learning process.

Figure 2 shows the trend of the accuracy for each batch and method. The quite evident gap between iCaRL and the other methods confirms the superior performance of this approach.

4. Ablation Study

iCaRL is a combination of NME Classifier and Binary Cross Entropy (BCE) loss used from both classification and distillation.
This section provides an ablation study that considers various combinations of classifiers, classification and distillation losses, with the aim of highlighting the gap between iCaRL and other possible choices.
As classifiers we chose:

- NME, as in iCaRL;
- KNN since like the former it also relies on the concept of neighbours, with k=10 neighbours;
- SVM with C=1 and RBF Kernel.

For the losses instead we used the following combinations of classification and distillation losses (in the given order), where y_i is the truth label and $\hat{y}_i$ is the prediction:

- L2 + L2 where $L_2 := \sum_{i=1}^{N}(y_i - \hat{y}_i)^2$ is the Least Squared Error;
- BCE + CE where $CE = -\sum_{i=1}^{N} y_i \log \hat{y}_i$ is the Cross-Entropy Loss;
- BCE + L1 where $L_1 := \sum_{i=1}^{N} |y_i - \hat{y}_i|$ is the Least Absolute deviation;
- BCE + BCE, as in iCaRL;

Each classifier was combined with the 4 pairs of losses, for a total of 12 combinations. Figure 3 shows the results per type of classifier. As expected, iCaRL outperforms all other choices.

The ablation study allowed us to also observe a particular phenomena.

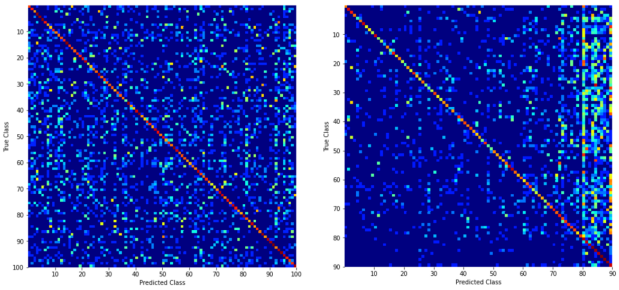

Figure 4. Confusion matrix at batch 10 for NME Classifier combined with BCE+BCE loss (left) and BCE+CE (right).

Specifically, Figure 4. shows that when the classification and distillation losses are of the same type, the confusion matrix is homogeneously distributed, while combining different losses leads to an unbalance towards the right (so more recently learnt classes) and in this specific case, a difference in accuracy of approximately 3.5% on the final batch. We think the observed imbalance is due to the fact that different losses can give results on different scales, so it would be appropriate to balance different classification and distillation losses by means of a weighing function. Such approach is not however inspected in this project.
As far as the hyperparameters for KNN and SVM are concerned, they were chosen after having performed a grid-search. For the training phase only the exemplars were used, in order to have a balanced dataset. Specifically 80% of this data was used for training and a 3-fold cross validation was implemented.
The results obtained on the test data are available in tables 1 and 2, and show that the accuracy of SVM and KNN increases respectively as C increases and K increases.

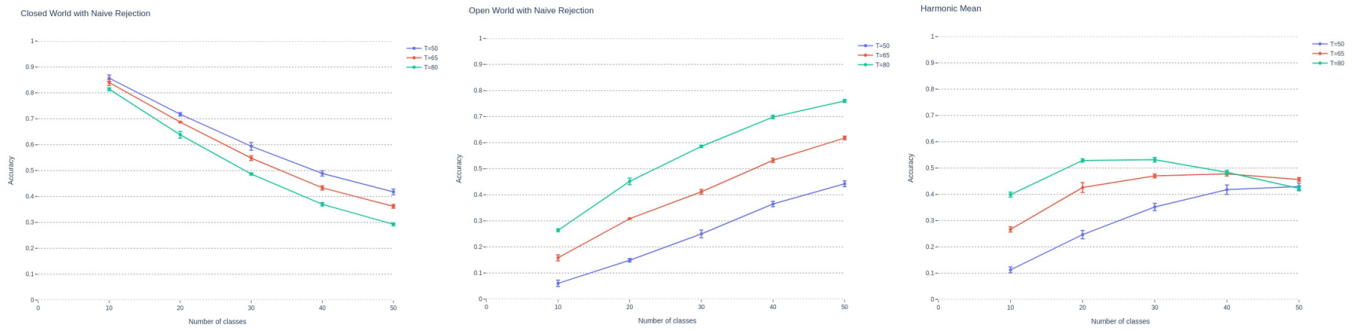

Figure 5. From left to right: Closed world with rejection, Open world with rejection and Harmonic mean of the previous scenarios for different values of the threshold T.

Batch	C=0.01	C=0.1	C=1
1	85.36 ± 0.46	84.1 ± 1.73	83.16 ± 2.59
2	43.31 ± 0.66	72.15 ± 1.86	71.18 ± 1.74
3	48.9 ± 0.9	61.42 ± 2.71	59.37 ± 3.90
4	48.15 ± 0.93	54.01 ± 1.14	53.55 ± 1.22
5	45.18 ± 0.95	51.67 ± 1.56	51.16 ± 0.98
6	44.07 ± 0.77	46.60 ± 1.39	50.67 ± 0.64
7	43.10 ± 1.01	44.18 ± 1.07	42.63 ± 1.80
8	40.44 ± 0.93	41.22 ± 1.25	43.86 ± 0.84
9	38.40 ± 1.23	38.96 ± 1.1	42.39 ± 0.85
10	36.39 ± 0.55	36.48 ± 0.59	38.64 ± 0.82

Table 1. Multi-class accuracy across all classes seen up to a certain point for SVM as the parameter C varies, with BCE loss for both classification and distillation.

Batch	K=4	K=7	K=10
1	85.6 ± 1.47	85.96 ± 1.55	85.83 ± 1.23
2	72.96 ± 0.68	74.11 ± 0.74	73.93 ± 0.67
3	63.25 ± 1.22	64.93 ± 1.36	64.95 ± 1.48
4	55.27 ± 1.5	56.52± 1.30	56.82 ± 1.03
5	50.93 ± 1.10	52.05± 1.38	52.90 ± 1.11
6	49.18 ± 0.88	50.56 ± 1.16	51.09 ± 1.09
7	44.22 ± 1.23	45.68 ± 0.98	46.27 ± 0.99
8	41.59 ± 0.93	43.09 ± 1.04	43.42 ± 0.78
9	38.72 ± 1.13	40.15 ± 1.26	40.39 ± 1.11
10	34.82 ± 0.88	36.23 ± 0.76	36.59 ± 0.84

Table 2. Multi-class accuracy across all classes seen up to a certain point for KNN as the parameter k varies, with BCE loss for both classification and distillation.

5. Open World

As previously stated, the second part of this project focuses on the Open World scenario. Incremental learning [2] [1] and open set recognition [8] [4] [5] are both problems addressed by the literature. However they are seldom solved together.

For this part of our study we divide the 100 classes of our data into two halves of 50 classes each, following BDOC's implementation [3]. The first half is dedicated to the training and testing phases of Closed World scenarios, whereas for the Open world scenario we use the first half for the training phase and the second half for testing.
A naive rejection strategy is implemented by fixing a threshold. For each probability assigned by the model, we check whether it is greater or lesser than such threshold. If it is higher, the model is considered highly confident and the class is considered to be known, otherwise the class is classified as unknown.

Let us analyse the considered scenarios more in detail:

- Closed World Without Rejection: this is the standard incremental learning scenario that has been considered thus far.
- Closed World With Rejection: analogous to the previous scenario with the addition of the previously addressed capability of rejecting a sample as unknown. Notice how since we do not provide samples of classes not seen during training, a perfect model should not reject any of them.
- Open World: in this setting the model is tested on the second half of the dataset, i.e. the unknown classes. In this case a prefect model should reject all samples.
- Open World harmonic mean: this scenario relies on the computation of the harmonic mean at each incremental step between the closed world with rejection scenario and the open set scenario.

The accuracy in the second and third settings are calculated as number of correct predictions over the total number of predictions.

As far as the choice of the threshold T is concerned, three different values were considered: 0.5, 0.65 and 0.80.

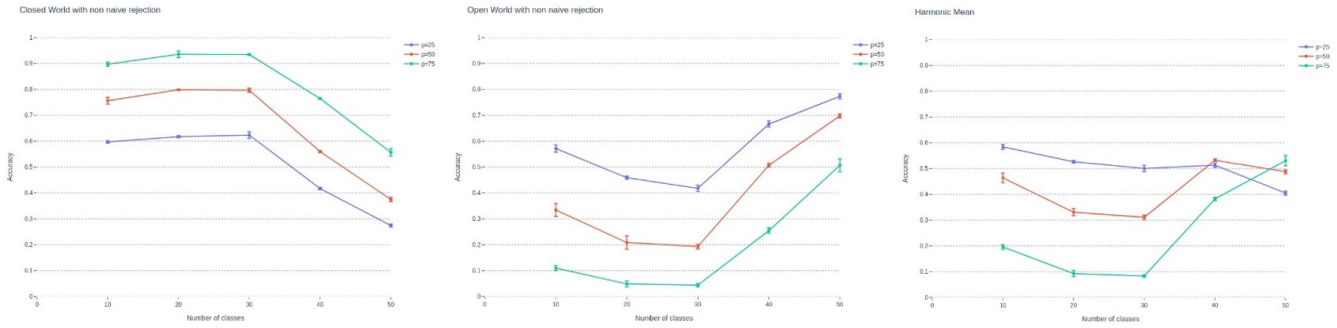

Figure 6. From left to right: Non naive closed world with rejection, non naive open world with rejection and Harmonic mean of the previous two scenarios for different values of p.

From Figure 5 we can see that closed world with rejection has a decreasing trend, contrary to the open world with rejection setting in which the trend is increasing. As expected, closed world performs better with a higher threshold T as the model is more confident in each prediction. The poor rejection that can be seen in the first steps of open world is probably due to fact that at this stage of the process only a few concepts are known, so the classification criteria is still very general, which makes determining whether an element is known or not challenging.
If we evaluate the harmonic mean of the two scenarios we find that the threshold T=0.8 yields more appreciable results on average, as can be seen in the third quadrant of Figure 5.

6. Proposed Modification

In this section we address our two proposed modification to the project.

6.1. Modifying the number of layers of the ResNet

Adding layers to a net generally means extracting more features. However, one must also consider the risk of overfitting the data. In fact, when all significant features have been detected, adding more layers leads to finding irregularities in the data, that in turn result in errors in our predictions. For this reason, we decided to study the trend of the accuracy of iCaRL as the number of layers of the ResNet increases, to find if there is a better choice for the number of layers.
Figure 7 shows the behaviour of iCaRL with different ResNets, respectively: ResNet32 (original), ResNet44 and ResNet56. Results obtained with the use of ResNet110 were not instead included, as they require changing hyperparameters to avoid the divergence of the network (a learning rate of an order of magnitude lower eliminated the divergence issue, however comparing the accuracy obtained on various ResNets using different learning rates would make the comparison futile).

As can be seen, using 44 layers allows us to achieve on average an increase of accuracy of 1.34% in the final batch, while with 56 layers we see a decrease in performance, probably due to the aforementioned motives. Since the modification gives a slight improvement, it is kept for the next step.

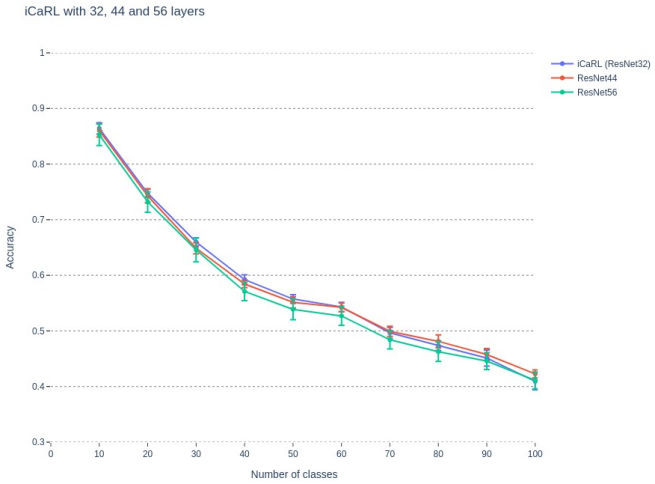

Figure 7. Accuracy plot of iCaRL with 32, 44 and 56 layers. These results were obtained by running the three options with 7 different seeds.

6.2. Modifying the naive rejection strategy

For the open world scenario, a non naive rejection strategy which takes inspiration from iCaRL's NME Classifier was introduced.
The steps to decide whether or not a sample should be rejected are the following:

1. For each class calculate the centroid, as seen in NME;
2. For each class calculate the distance between the elements of such class and the centroid, and store the p*th*

percentile of such distances, which will be our threshold value;

3. While predicting with NME check if the distance between the element and the centroid of the assigned class is greater than the threshold. If so, classify the sample as unknown, otherwise accept the prediction;

This modification relies on thresholds computed using available data, rather than fixed ones.

Figure 6 shows the behaviour of the adopted strategy in the closed world and open world setting, with also the harmonic mean reported in the right quadrant. The approach was tested with various values of p, in order to determine the one yielding better performances, which according to the harmonic mean is on average p=25.

The harmonic mean for such value of p was then compared to the best non naive approach, which we previously found to be given by a threshold of T=0.8. Figure 8. shows such comparison. On average, as reported in Table 3, the non naive strategy slightly outperforms the naive one, especially in the first batch, but still leaves room for improvement.

Batch	Harmonic Naive	Harmonic Non Naive
1	39.87 ± 0.92	58.42 ± 0.94
2	52.89 ± 0.62	52.64 ± 0.51
3	53.16 ± 0.88	50.04 ± 1.22
4	48.36 ± 0.07	51.30 ± 0.80
5	42.25 ± 0.09	40.50 ± 0.77
Average	47.31 ± 0.83	$\mathbf{50.58 \pm 0.85}$

Table 3. Comparison between the Naive Harmonic Mean for T=0.8 and the Non Naive Harmonic Mean for p=25.

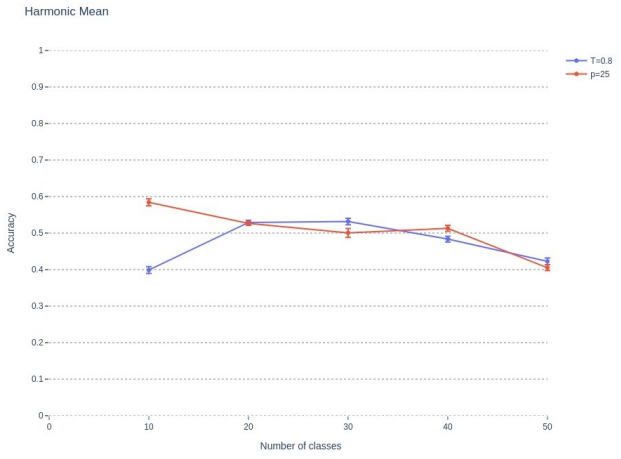

Figure 8. Comparison between the naive harmonic mean for T=0.8 and the non naive harmonic mean for p=25.

7. Conclusion

We have experimented that the use of a ResNet44 in place of a 32-layer one allows us to improve the performance of iCaRL.
Furthermore, we have understood that in the open world scenario thresholds learned from data are slightly more reliable.
Lastly, the learning procedure is computationally efficient and relies on data already given by the implementation of iCaRL. However, the models performances are still not satisfying.

References

[1] Raffaello Camoriano, Giulia Pasquale, Carlo Ciliberto, Lorenzo Natale, Lorenzo Rosasco, and Giorgio Metta. Incremental robot learning of new objects with fixed update time. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3207–3214, 2017.

[2] Raffaello Camoriano, Silvio Traversaro, Lorenzo Rosasco, Giorgio Metta, and Francesco Nori. Incremental semiparametric inverse dynamics learning. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 544–550, 2016.

[3] Dario Fontanel, Fabio Cermelli, Massimiliano Mancini, Samuel Rota Bulò, Elisa Ricci, and Barbara Caputo. Boosting deep open world recognition by clustering. *CoRR*, abs/2004.13849, 2020.

[4] Victor Fragoso, Pradeep Sen, Sergio Rodriguez, and Matthew Turk. Evsac: Accelerating hypotheses generation by modeling matching scores with extreme value theory. In *2013 IEEE International Conference on Computer Vision*, pages 2472–2479, 2013.

[5] Manuel Günther, Steve Cruz, Ethan M. Rudd, and Terrance E. Boult. Toward open-set face recognition. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 573–582, 2017.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[7] Ilja Kuzborskij, Francesco Orabona, and Barbara Caputo. From n to n+1: Multiclass transfer incremental learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2013.

[8] Fayin Li and H. Wechsler. Open set face recognition using transduction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(11):1686–1697, 2005.

[9] Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. volume 24 of *Psychology of Learning and Motivation*, pages 109–165. Academic Press, 1989.

[10] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. *CoRR*, abs/1611.07725, 2016.