Different contrastive self-supervised learning techniques for unlabelled dataset

Robert Melikyan [*] **Jash Doshi** [*]

Abstract

Supervised and unsupervised learning problems are pillars of machine learning and deep learning algorithms but it is not always possible to label all the data manually. Need of the algorithms which can create some relation in the feature space of the dataset of labelled images with the unlabelled images to automate the data labelling process and increase the accuracy of our model at the same time arose. Here, for a semi supervised learning task, different state of the art algorithms were used, which helped us compare the behaviours of those algorithms. in which teams a model using new semi and self supervised techniques to achieve a classification task where only 25,600 labelled images with 800 classes, where 32 images of each classes were given. More 25,600 images were provided as validation set and 5,12,000 images were given as unlabelled images for the project. The aim was to try different techniques and achieve maximum accuracy on the given data.

Keywords:

1. Introduction

1.1. Objective

The project was given in a competition format. The conventional classification tasks are easy to achieve results with and there are many state of the art models available which can give as good as human observation. But the main challenge of this competition was the fact that there were only 32 labeled images per class available, which makes training deep convolutional neural networks much more difficult. This is why we used techniques such as transfer learning and data augmentation to improve the test accuracy of the model. There are other research going on in field of self and semi supervised learning to solve the same problem. Throughout different phases of this project, we tried different methods and succeeded to increase accuracy in each submission.

1.2. Data

The images provided were 96 x 96 coloured images. The division of the data was,

1. 512k unlabelled images,
2. 25,600 labeled training images (32 examples, 800 classes),
3. 25, 600 labeled validation images (32 examples, 800 classes).

2. Literature review

We tried different approaches with each leader board and finally ended up using the MoCo as our final submission.

For the first leader board we went through some research paper in which the idea of pseudo labelling was introduced. Where after training on the labeled image dataset, and after achieving satisfactory accuracy you try to predict the labels for your unlabelled images and then try to label those with the highest confidence score. After each such iteration, you add those newly pseudo labelled images in your original training dataset and try to train your model again with this new training dataset. Repeat the process till you get good results.
We initially tried this approach but realized that this naive approach was taking too much resources to train and the increase in accuracy at each step after pseudo-labelling step was 0.1%. Which was not that good and in case of wrong labels the training would harm the model parameters.

After which we moved to first training our unsupervised images for some pre-training task (1). In the paper (1) that we referred, they used the augmented views of the same image and then tried to train the model by minimizing the loss which was calculated on the bases of the output features of those 2 images and then taking log softmax of those features. The pseudo code for that is given below.

We used ResNet50 as our backbone model for this approach. Now, after the unsupervised learning as shown from algorithm 1. We then trained the model on our labelled dataset by freezing all the layers except the last 2 layers of the model and re-initializing those 2 layers.

Algorithm 1 Self-Classifier PyTorch-like Pseudocode

```
N: number of samples in batch
C: number of classes
aug(): random augmentations
For x in loader:
  x1, x2 = aug(x), aug(x)
  logits1, logits2 = model(x1), model(x2)
  log_y_giv_x1 = log_softmax(logits1, dim=1)
  log_y_giv_x2 = log_softmax(logits2, dim=1)
  x1_giv_y = softmax(logits1, dim=0)
  x2_giv_y = softmax(logits2, dim=0)
  l1 = - (x2_giv_y * log_y_giv_x1).sum() / N
  l2 = - (x1_giv_y * log_y_giv_x2).sum() / N
  L = (l1 + l2) / 2
  L.backward()
  optimizer.step()
```

After training this model 300 epochs for unsupervised and 100 epochs for supervised, with batch-size 512 and 256 respectively. We were able to reach initial 16% accuracy.

In the paper(1), they had used CIFAR-10 dataset for the experiment and then further divided that into test, validation and then unlabelled. But since we know that there were only 10 classes and we had 800 classes.

Thus, to improve accuracy further we took these results as our baseline results and tried to improve upon these results using other techniques.
After we moved to some advance approaches. SimCLR (2) was one of them. It again uses a similar approach to compute the loss based on similarity, but SimCLR uses a contrastive loss called “NT-Xent loss”, which makes the learning faster and more efficient for unlabelled dataset. Again we trained this model for 200 epochs for unsupervised and 100 epochs for supervised, with 512 and 256 batch size, we were able to get 26.5% accuracy for validation on the labelled dataset. After which we went on to make modifications in that model to try to improve accuracy, by training it for ResNet101 model, but we were only able to get only so far till 28%.

Our subsequent attempts included an adaptation of SwAV (3) given that convergence was faster and it was computationally lighter, however preliminary runs saw little improvement from the SIMCLR method. Other than the clustering methods mentioned previously, distillation methods such as BYOL (4) were initially considered but disregarded after a long computational runtime on the hardware available to our team. Thus our team settled on focusing contrastive methods as our choice of self-supervised learning, with the preferred algorithm of choice being MoCo (5). As shown in Figure 1 (5), MoCo maintains a dictionary and lookup architecture for unsupervised contrastive learning.

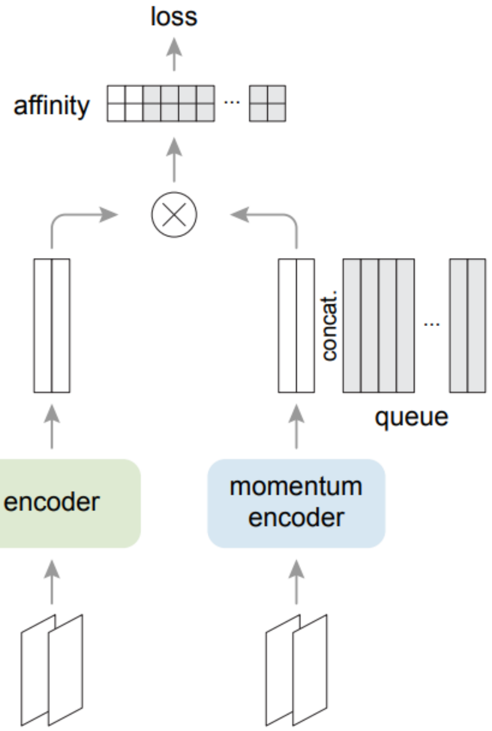

Figure 1. MoCo Architecture

Despite the added complexity compared to models such as SimCLR, this team hypothesized that the online nature of the algorithm would lead to better tuning especially once run across the custom classes. After a run of unsupervised training we took that model and then ran it again on a supervised learning block by fine tuning the last layers of the network.

3. Preliminary Results and Modifications

3.1. Unsupervised Training

Initial runs of the MoCo algorithm were run in an unsupervised manner before being joined with ResNet50, which serves as the backbone of our supervised architecture. Given a maximum of 2 GPUs, we trained our unsupervised model for 200 epochs, 250 and 300 epochs.

Despite larger iterations, convergence was ineffective leading to our empirical finding of an optimum of 300 epochs having run for around 50 hours. We noticed that the loss was almost constant between the iterations of epochs from 250 to 300. When we used the unsupervised model trained for 200 epochs it gave us around 32% accuracy. So we used the one trained till 250 epochs, which ended up giving around 34.8% accuracy and the final unsupervised model trained for 300 epochs gave us the validation accuracy at this point was 36.5%. As seen from the trend we decided that training the model further would not help us to improve the results drastically. Our guess was that it would take us around 39% but our team hypothesized that additional representative labels would improve the model accuracy quickly and this would also help us generalize the learned features more if we aim to use dataset with different distribution it would still give us a satisfactory results if we avoid over-training on the existing data.

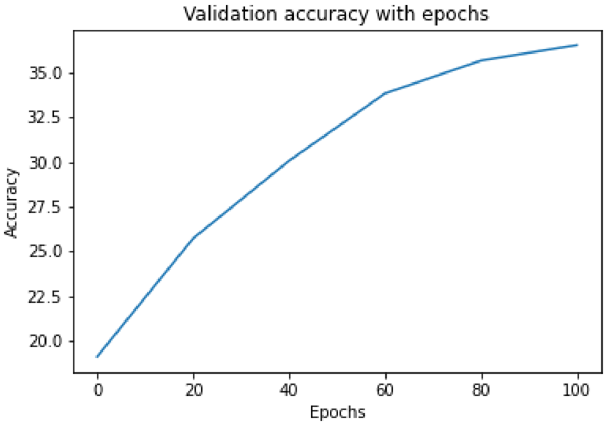

Figure 2. Accuracy over 25,600 original images for MoCo

3.2. Label Request

To improve our model, our team would be able to request an additional 12,800 images to be labelled. From the varying methods of image selection, such as edge case selection or contrastive pairing, literature relating to representative samples (6) and the effectiveness of random sampling (7) under these circumstances our decision was to approach our own challenge with a similar approach.

In other words, our methodology for image selection was to randomly sample images across the spectrum from points of both high and low energy in order to maintain the representative structure of the data. Our choice of said methodology was also partially driven by the concern that selection of edge cases would lead to internal biases in the model and an intractable manifold.

Our hypothesis included that a representative random sample would address this concern as well as potentially a robust transferable model.

3.3. Supervised Training and Fine Tuning

Both prior and post the labelling request, supervised training was conducted to attempt gauge our model's effectiveness. Our architecture of choice was ResNet50 (8) and our approach would be to fine tune the outer layers specifically for our task.

Given our resources of 2 GPUs, number of allocated GPU hours and reasonable runtime, we settled on a hyperparameter of 100 epochs for all supervised layer training.

As mentioned, prior to the addition of new labels, our accuracy was 36.5%, this increased to 37.32% after the addition of new labels.

4. Evaluation

The resulting outcomes of the preceding experimentation are summarized in *Table 1*.

Overall the selection of MoCo was grossly beneficial to the accuracy of the model (almost by double). However a notable insight is that the additional data only marginally improved the predictive power of the model. But while visualizing correctly and wrongly classified images using clustering analysis we observed that the images classified wrongly have really similar features representation of the class it has been assigned to. Meaning that in feature space correct classes for these wrongly classified images were really near in some cases. As for 800 classes the vector space and their decision boundaries overlap for some classes, we could conclude that it was not possible to achieve a perfect model using MoCo architecture.

Another particular insight may be that the method of label selection may not have been as effective for the given architecture chosen of unsupervised learning. Indeed the models in the referred papers for our decision of random sampling were not applied to contrastive methods such as MoCo, so perhaps the compatibility between such methods is low and may be better suited for potentially clustering methods like SIMCLR or SwAV. However this remains to be tested.

Another potential point of insight may be whether the architectural choice of MoCo was a limiting point. Given the time constraints of the competition part of our decision to choose MoCo was to be able to adapt it accordingly with enough time to deliver a model.

However given more time another path of interest could be one with SwAV combined with Transformers, to test whether the fast convergence of the former could be boosted by the flexibility of the latter.

Table 1. Classification accuracies for adapted MoCo with representative random sampling labels.

MODEL TRAINING	ACCURACY	IMPROVEMENT
BASELINE	16%	-
SIMCLR(RESNET50)	26.5% ± 0.1	+10.5
SIMCLR(RESNET101)	28% ± 0.2	+12
ORIGINAL LABELS(MOCOV2)	36.5% ± 0.1	+20.5%
NEW LABELS (MOCOV2)	37.32%± 0.1	+0.82%

5. Conclusion

The concluding evidence suggests that using the Momentum contrast for unsupervised visual representation learning algorithm, otherwise known as MoCo (5) is an effective baseline model for self-supervised learning and can be improved by representative random sampling.

A batch size of 256 combined with training epochs of 300 for unsupervised learning and 100 on supervised learning running on ResNet50 with the rest set to the default hyperparameters mentioned in the original paper, all show promising results and adequate performance.

Potential limitations may be the compatibility between label selection and the contrastive architecture used in our model. Further experimentation may be necessary to further explore these implications.

6. Acknowledgements

This effort wouldn't have been possible, first and foremost, without the insight from Yann LeCun and the patience of Alfredo Canziani.

The diligence and thoroughness of Jiachen Zhu and Vlad Sobal immensely assisted in bridging the theory and experience gap. Neeraj Yadav and Tin Luu's insightful comments over the duration of the course also assisted in better comprehending both the material and practical intuition.

Finally we would like to thank the team at NYU for facilitating the educational and guest speaking opportunities that enriched this course. This team is humbled by the opportunity to participate and contribute, and look forward to further learn and refine our approaches.

References

[1] A. B. Elad Amrani, "Self-supervised classification network," 2021. [Online]. Available: https://arxiv.org/pdf/2103.10994.pdf

[2] T. Chen, S. Kornblith, M. Norouzi, and G. E. Hinton, "A simple framework for contrastive learning of visual representations," *CoRR*, vol. abs/2002.05709, 2020. [Online]. Available: https://arxiv.org/abs/2002.05709

[3] M. Caron, I. Misra, J. Mairal, P. Goyal, P. Bojanowski, and A. Joulin, "Unsupervised learning of visual features by contrasting cluster assignments," *CoRR*, vol. abs/2006.09882, 2020. [Online]. Available: https://arxiv.org/abs/2006.09882

[4] P. H. Richemond, J.-B. Grill, F. Altché, C. Tallec, F. Strub, A. Brock, S. Smith, S. De, R. Pascanu, B. Piot, and M. Valko, "Byol works even without batch statistics," 2020.

[5] K. He, H. Fan, Y. Wu, S. Xie, and R. B. Girshick, "Momentum contrast for unsupervised visual representation learning," *CoRR*, vol. abs/1911.05722, 2019. [Online]. Available: http://arxiv.org/abs/1911.05722v3

[6] F.-T. Hong, W.-H. Li, and W.-S. Zheng, "Learning to detect important people in unlabelled images for semi-supervised important people detection," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[7] E. Nowak, F. Jurie, and B. Triggs, "Sampling strategies for bag-of-features image classification," *Computer Vision – ECCV 2006 Lecture Notes in Computer Science*, p. 490–503, 2006.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.