# Chapter 47
# Self-supervised Learning with Deep Neural Networks for Computer Vision


**Tan Huan Xi Gregory, Neo Souw Chuan, and Shen Bingquan**

**Abstract** Self-supervised learning has gained popularity in recent years due to a need to avoid the expensive costs of large-scale data annotation. This field has had huge developments in recent months, with the state-of-the-art self-supervised learning methods achieving results that surpass their supervised counterparts on the ImageNet dataset. In this study, we investigate and implement 2 forms of self-supervised learning: Momentum Contrast (MoCo) and autoencoders, which are contrastive and generative methods, respectively. Through several experiments, we analyze the quality of the latent representations that are learnt by each method and assess whether they allow for increased performance when labels are scarce. We also propose a dual head network for self-supervised learning, combining elements from both the above methods and study the effectiveness of such a method.

## 47.1 Introduction

Deep neural networks, specifically convolutional neural networks, have seen great success in Computer Vision, with networks such as ResNet being able to classify images with great accuracy. The current convention for training deep neural networks is supervised learning, where the network is trained using a large amount of labeled data, split into train, test and validation sets. However, a bottleneck is arising in this process. Annotating and labeling image data remains a time consuming and expensive process that is prone to human error. Furthermore, supervised networks suffer from generalization errors and are prone to adversarial attacks. As such, self-supervised learning is an increasingly popular field in machine learning, which is the process of utilizing the large amounts of unlabeled data available to extract effective representations from images. In recent months, notable successes in self-supervised

T. H. X. Gregory (✉) · N. S. Chuan
Hwa Chong Institution, Singapore, Singapore
e-mail: 171432j@student.hci.edu.sg

S. Bingquan
DSO National Laboratories, Singapore, Singapore



H. Guo et al. (eds.), *IRC-SET 2021*, https://doi.org/10.1007/978-981-16-9869-9_47

representation learning with images include MoCo [1], SIMCLR [2], BYOL [3] and SimSiam [4]. The latent representations learnt through self-supervised pre-training can then be transferred into downstream tasks, such as Object Detection, Semantic Segmentation and Image Classification. For example, a model trained using self-supervised learning can allow for fewer labels to be used to achieve high classification accuracy. This can be especially helpful in fields where labeled data is incredibly limited, such as medical imaging or robotics. Our aim is to investigate and explore various methods for self-supervised learning and propose our own method.

In this paper, we study 2 different methods of self-supervised learning: MoCo and autoencoders. These 2 training methods vary greatly, as MoCo utilizes contrastive learning approach and is a more recent, state-of-the-art method, whereas autoencoders are a more traditional form of self-supervised learning, which uses a generative approach. We also propose a dual head network, which adds an additional decoder head to the query encoder of the MoCo model so as to produce a weighted loss that combines the contrastive loss of MoCo with the reconstruction loss from the autoencoders, to see if combining these two contrastive and generative would result in a more robust model. We use ResNet18 as our backbone for the models and train them on the CIFAR-10 dataset.

We then evaluate their performance on downstream image classification tasks to analyze how strong the learnt representations are. We also study how this classification accuracy scales with the amount of labels, so as to find out if self-supervised learning is effective at reducing the amount of required labels for high classification accuracy.

## 47.2 Related Work

A large body of research in Computer Vision is dedicated to training convolutional neural networks without the use of labeled data. There have been many proposed methods for self-supervised learning, which can be broadly split into two categories: Pretext Tasks and Contrastive Learning.

### *47.2.1 Pretext Tasks*

One of the most popular methods for self-supervised learning for Computer Vision involve automatically creating "pseudo-labels" based on data alone and training the deep neural networks on auxiliary tasks using these labels. Many pretext tasks have been used for self-supervised learning. One example is autoencoders, which involves the pretext task of reconstructing an image based on a latent representation, treating the original image as the pseudo-label. Other works include colorization of grayscale images [5] and predicting the rotation of an image [6]. Even though these methods

have proven effective at learning image representations, the representations are far from their supervised counterparts and are unable to achieve state-of-art performance.

### *47.2.2 Contrastive Learning*

Recent developments in self-supervised learning mostly involve contrastive self-supervised learning, focusing on the task of Instance Discrimination for training. In short, Instance Discrimination task treats each image and its augmented version as its own class. A contrastive loss, such as InfoNCE [7], is used to maximize the similarity between the different augmentations of the same image, while at the same time minimize the similarity between the augmented images and entirely different images. Very aggressive image augmentation has proven to be integral to the success of the instance discrimination task. Many works have taken varying approaches to this task. CPC [7] and CPC v2 [3] propose using patches within an image as instances. Wu et al. [8] uses a memory bank to store a large amount of negative instances, which MoCo [1] expands upon by maintaining a queue of negative examples. SIMCLR [2] and BYOL [3] draw negative instances from the same mini-batch as the positive sample, requiring large batch sizes for successful representations to be learnt. SimSiam [4] uses a Siamese architecture, proposing that doing so eliminates the need for the negative examples used in SIMCLR [2] and the momentum encoder used in BYOL [3] and MoCo [1]. Currently, contrastive self-supervised learning methods have state-of-the-art performance, with some methods even surpassing supervised methods on the ImageNet dataset.

## 47.3 Methods

### *47.3.1 Dataset and Model*

In this study, we used the CIFAR-10 dataset, which contains 50 000 $32 \times 32$ images consisting of 10 classes of different objects and animals. The dataset was chosen due to its popularity and relatively small size in comparison to other commonly used benchmark datasets such as ImageNet, which contains over 1 million images. This smaller dataset size made it feasible for us to train our models, given that we lacked access to powerful hardware for training. We also chose ResNet18 [9] as the main encoder network to be trained using the self-supervised training method. We chose this model as opposed to the more commonly used ResNet50 as its shallower depth was sufficient for a relatively small dataset like CIFAR-10. The model architecture was slightly changed to better suit the CIFAR-10 dataset, with the first MaxPooling layer being removed.

### 47.3.2 Experiment 1: Momentum Contrast

One of the state-of-the-art methods for the self-supervised learning that we have chosen to study in detail is Momentum Contrast (MoCo) [1], a method that uses contrastive learning. MoCo defines self-supervised contrastive learning as a dictionary lookup problem, where encoders are trained to encode similar images into embeddings that are similar to each other and dissimilar to the embeddings of different images. MoCo trains a visual representation encoder by matching an encoded query q to a dictionary of encoded keys using a contrastive loss [7]. The dictionary is built as a queue, which is constantly updated by enqueuing the current batch and dequeuing the oldest batch. The use of a dictionary decouples the number of keys, K, from the batch size, allowing for a large number of keys without a large batch size, which is memory-intensive. The keys are encoded by a slowly progressing momentum encoder, which derives its weights from moving average of the weights of the query encoder for consistency. Denoting the $\theta_k$ as the parameters of the momentum encoder $f_k$ and $\theta_q$ as the parameters of the query encoder $f_q$, $\theta_k$ is updated by:

$$\theta_k \leftarrow m\theta_k + (1 - m)\theta_q$$

where $m$ is the momentum hyperparameter. It was found that a relatively large momentum worked much better suggesting that a slowly progressing key encoder is integral to the use of a dictionary (Figs. 47.1 and 47.2).

In the training procedure, the MoCo model randomly performs a series of strong augmentations (Color Jitter, Gaussian Blur, Random Resized Crop, Random Horizontal Flip, Random Grayscale) on the same image twice to produce two different views of the same image, $x^q$ and $x^k$, which are then fed into the query

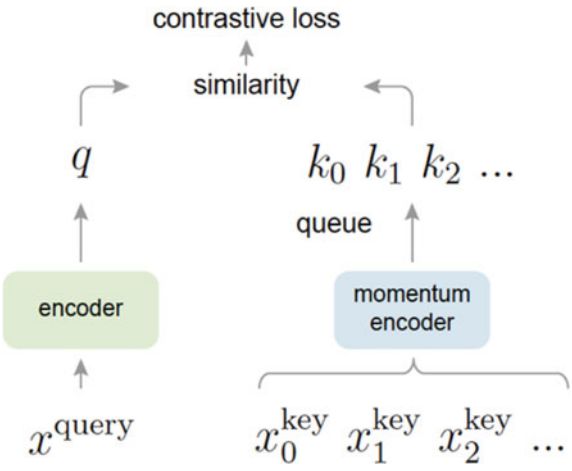


**Fig. 47.1** MoCo maintains a dictionary of encoded keys which serve as negative samples for the contrastive loss function

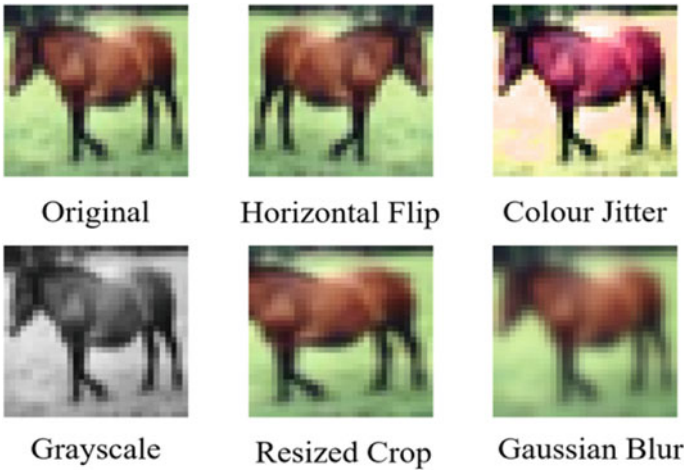


**Fig. 47.2** Augmentations used on CIFAR-10 images for MoCo

encoder $f_q$ and momentum encoder $f_{k,}$ respectively, to produce encoded query $q$ and its positive key $k^+$.

A contrastive loss function known as InfoNCE [7] is then applied. The function is defined as follows:

$$L_q = -\log \frac{\exp(q \cdot k^+/\tau)}{\Sigma_{i=0}^{K} \exp(q \cdot k/\tau)}$$

where $\tau$ is a temperature hyperparameter. The sum is over one positive sample and K negative samples from the dictionary. This loss can be seen as the log loss of a $(K + 1)$-way softmax-based classifier that tries to classify $q$ as $k^+$. Minimizing this loss function helps the query encoder to learn to discriminate between the positive and negative samples, thus allowing it to learn strong image representations in a self-supervised manner. After training, the query encoder is kept as a feature extractor for downstream tasks. We trained our MoCo model using ResNet18 [9] as the base encoder, with $K = 4096$. The linear head was replaced with a projection MLP head which contained a hidden layer with 512 neurons and a ReLU activation, a feature from [2] that improved performance. A batch size of 256 was used and the model was trained for 800 epochs using SGD with momentum as the optimizer. The learning rate was initially set at 0.03 and was divided by 10 at epoch 120 and epoch 160.

## 47.4 Results and Discussion

We evaluated the quality of the learnt representations by freezing the ResNet18 encoder and training a single fully-connected layer on top of it to classify the images. Our linear classification protocol is as follows: the linear layer on top of the ResNet18 was trained for 100 epochs, using SGD optimizer with momentum 0.9. The learning rate was set at 0.3 and divided by 10 at the 60th and 80th epoch. No image augmentations were applied. As a comparison, a fully supervised ResNet18 was also trained using basic augmentations (Random Resized Crop, Horizontal Flip). We also trained a linear layer on top of a randomly initialized ResNet18. The optimizer, learning rate and learning rate schedule were the same for all tests (Table 47.1).

The above results prove that MoCo is indeed a highly effective method for self-supervised learning as it achieves a high linear classification accuracy, one that is rather close to the accuracy of a fully supervised network. This is consistent with the successful results that other contrastive self-supervised learning methods such as SIMCLR and BYOL have achieved on the ImageNet dataset (Fig. 47.3).

The t-SNE plot of the output of from the final Average Pooling Layer of the model shows clear clustering of the classes in the CIFAR-10 dataset, with the ship, truck, airplane and automobile classes being the most linearly separable. This reaffirms that a good representation has been learnt, allowing the encoder to properly discriminate between images of each class.

### *47.4.1 Experiment 2: Autoencoder*

Autoencoders are one of the earliest forms of self-supervised learning where the image is compressed into a small latent space through an encoder before being decompressed back into the image with a decoder. The loss is then taken by taking the mean squared error of the reconstructed image and the original image. For self-supervised learning, the more common approach is to use a denoising autoencoder rather than a regular autoencoder, as shown in [6]. The input images are corrupted with either noise or occlusion and the autoencoder is expected to restore the image to its original form. The architecture of the denoising encoder is identical to that of a basic autoencoder, with the only difference being that the input image is a corrupted one.

**Table 47.1** Results of linear classifier for MoCo

| Training method | Train accuracy/% | Test accuracy/% |
|---|---|---|
| MoCo | 86.59 | 86.62 |
| Fully supervised | 91.61 | 93.08 |
| Randomly initialized | $36.21 \pm 0.47$ | $34.14 \pm 0.95$ |

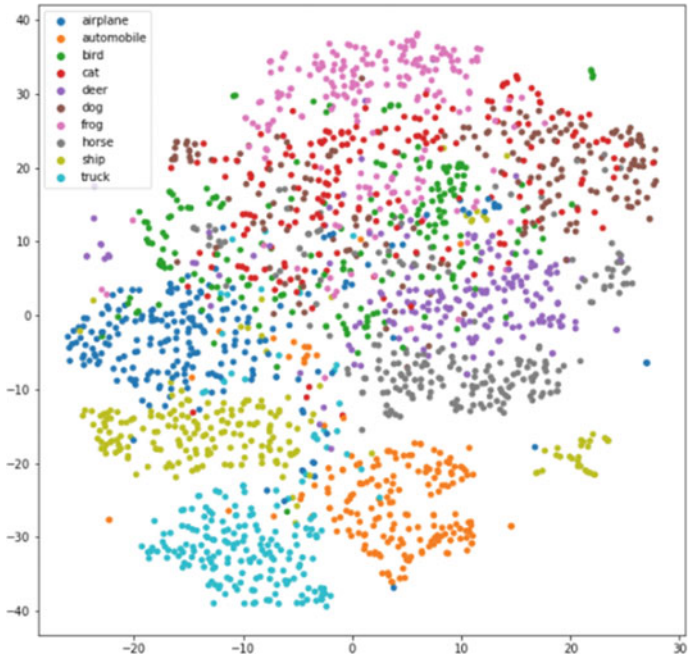


**Fig. 47.3** t-SNE visualization of representations learnt by MoCo

We implemented our autoencoders using ResNet18 [9] as the base encoder to allow for a consistent comparison with MoCo. For the decoder portion of the autoencoder, we simply reversed the layers of the regular ResNet18, with the addition of up sampling layers, which used interpolation to resize the compressed latent representation back to the size of the original image. For a more in-depth study, we tested 3 variations of autoencoders: a basic autoencoder, a denoising autoencoder and a denoising autoencoder with strong augmentations. The first denoising autoencoder applied pixel-level Gaussian noise to then images (with a standard deviation of 0.15). We only applied gaussian noise as opposed to other works which cut out portions of the image, as the low resolution of CIFAR-10 would render such image corruption ineffective. The second denoising autoencoder applied the same strong augmentation that was used for our MoCo model (shown in Fig. 47.2). Each autoencoder model was trained for 200 epochs with Adam optimizer with an initial learning rate of 0.001, which was divided by 10 at the 80 and 160 epochs.

## 47.5 Results and Discussion

We evaluated the effectiveness of autoencoders for self-supervised representation learning by applying the same linear classification method that we used for MoCo.

After training, the decoder portion of the autoencoder was discarded and the encoder was frozen. A linear layer was then trained on top of these frozen features, using the same hyperparameters as the linear classification for MoCo (Table 47.2).

From the results above, autoencoders prove to be a less effective method for self-supervised learning, with classification accuracies significantly lower than MoCo. However, the linear classification accuracy of all 3 autoencoder methods do still show a large improvement in comparison to the randomly initialized network, indicating that image representations have indeed been learnt by the encoder. The results also indicate that the quality of the learnt representations tend to increase with the amount of augmentation added to the image, even if the increased augmentations harm performance on the pretext task, as shown below. The increase from the basic autoencoder to the denoising autoencoder is relatively small, whereas the increase from the denoising autoencoder to the denoising autoencoder with strong augmentation is quite large. This indicates the importance of strong image augmentation in self-supervised learning and shows that it is incredibly helpful not only for contrastive methods, but also for generative ones (Figs. 47.4, 47.5 and 47.6).

Even though the performance of the autoencoder with the strongest augmentation on the reconstruction task was the poorest, the learnt representations proved to be the strongest, showing that the performance on the pretext task and the performance on downstream tasks can be completely unrelated.

**Table 47.2** Results of linear classifier for autoencoders

| Training method | Train accuracy/% | Test accuracy/% |
|---|---|---|
| Regular autoencoder | 54.79 | 54.83 |
| Denoising autoencoder | 58.50 | 58.62 |
| Autoencoder with augmentation | 67.71 | 67.73 |
| Fully supervised | 91.61 | 93.08 |
| Randomly initialized | $36.21 \pm 0.47$ | $34.14 \pm 0.95$ |

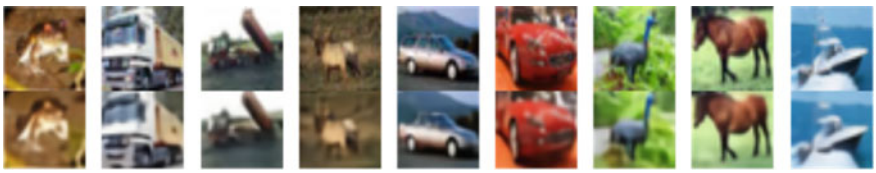

**Fig. 47.4** Original images (top) and reconstructed images (bottom) for basic autoencoder

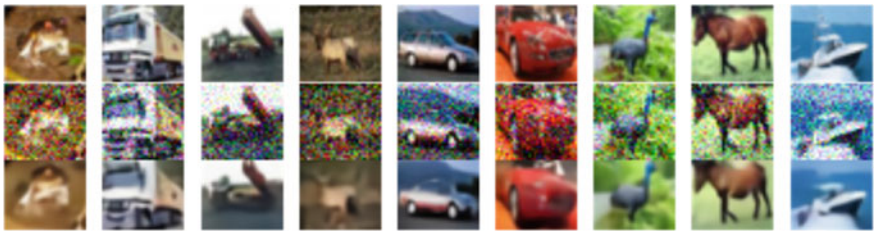

**Fig. 47.5** Original images (top), noisy images (middle) and reconstructed images for denoising autoencoder

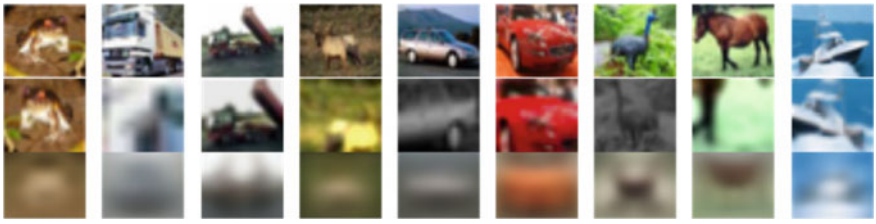

**Fig. 47.6** Original images (top), augmented images (middle) and reconstructed images for autoencoder with augmentation

### *47.5.1 Experiment 3: Dual Head Network*

After studying and implementing the above methods of self-supervised learning, we wanted to explore the possibility of combining multiple methods (pretext task and contrastive) to see if it would result in better performance. To study this, we created a dual head network, which feeds the encoded query from the query encoder of the MoCo network into a decoder to produce an MSE loss. This loss is then combined with the contrastive loss of the original MoCo network via the following equation:

$$L = \alpha L_{\text{contrastive}} + (1 - \alpha) L_{\text{MSE}}$$

where α is the hyperparameter adjusting the weightage of each individual loss function (Fig. 47.7).

Due to the inclusion of the regressive MSE loss in the loss function, the learning rate had to be lowered from 0.03 to 0.003 to prevent the loss from increasing exponentially. We trained the model with SGD optimizer with momentum 0.9 for 800 epochs, dividing the learning rate by 10 at the 160th, 240th and 320th epoch.

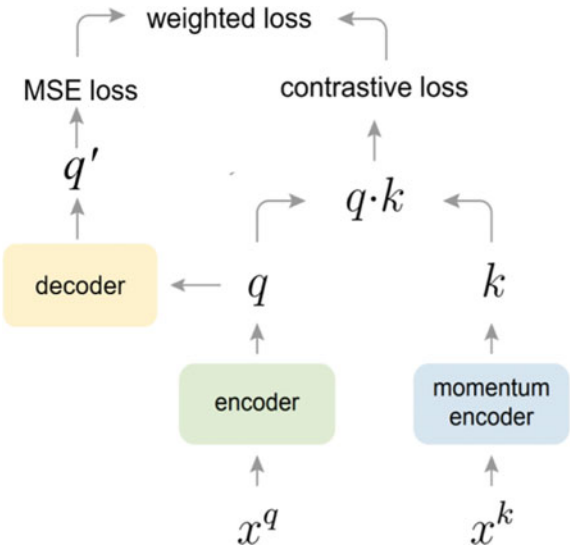


**Fig. 47.7** Diagram explaining the structure of the dual head network

## 47.6 Results and Discussion

The results above show that as alpha values increase and the influence of the contrastive loss increases, the accuracy increases, then stagnates after alpha = 0.7. This is contradictory to what we expected as we hoped that the addition of the autoencoder would increase the accuracy since new latent features are detected allowing for better classification. However, the results show otherwise, and we hypothesize that this is due to the addition of the two loss functions, which results in the gradient descent being in two not necessarily similar directions. Concatenating the outputs of the alpha = 1.0 and alpha = 0.0 and classifying the combined features yielded an accuracy of 73.6%, which was very close to the accuracy of the alpha = 1.0 network, suggesting that the features learnt by the autoencoder and MoCo sections ended up learning were similar. The 10 × lower learning rate used could also be a factor contributing to the dual head network being less effective at learning image representations than a pure MoCo. The lower learning rate had to be used to prevent the regressive MSE Loss from exploding, but may have caused the model to get stuck in local minima and become unable to effectively minimize the loss function. Possible improvements to this dual head network include having separate weights for the two loss functions, to give the MSE Loss a smaller weightage in order to prevent it from exploding even with a greater learning rate. Further tuning of the learning rate and momentum hyperparameters and use of other optimizers such ADAM and AdaGrad could also help to avoid local minima (Fig. 47.8).

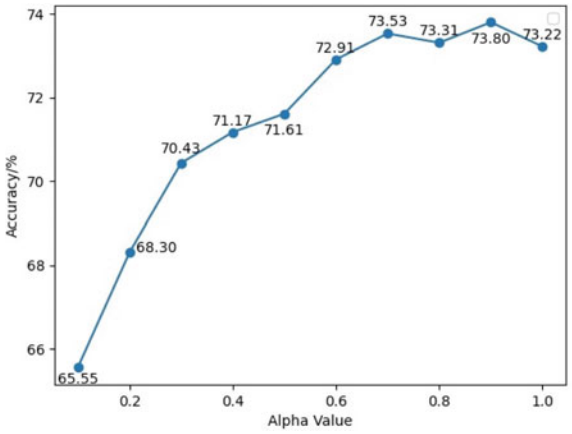


**Fig. 47.8** Graph of test accuracy against alpha value for dual head network

### *47.6.1 Experiment 4: Varying Amounts of Labels*

After implementing various methods of self-supervised learning, we wanted to analyze the effectiveness of the trained encoders on one of the key goals of self-supervised representation learning: reducing the reliance on large amounts of labeled data. We ran tests on 4 different label benchmarks: 1000, 2000, 4000 and 10,000 labels. We ensured that an equal number of labeled images from each of the 10 classes. We chose to run this test with our MoCo network, the autoencoder with augmentation, as well as the dual head network with an alpha value of 0.5, which meant an equal representation of both loss functions. We once again applied our evaluation protocol that was used for the linear classification with all the labels, training a single fully-connected layer on top of the frozen encoder. Since the labeled images were chosen at random, we ran the test 5 times, each with a different random seed. This was benchmarked against a supervised ResNet18, which was trained with the basic augmentations of Horizontal Flip and Random Resized Crop.

## 47.7 Results and Discussion

As the above results show, the accuracy of the self-supervised methods is much greater when the amount of available data is sparse. With 1000 labels, the accuracy of the encoder trained with MoCo is almost double that of the supervised method. Even the autoencoder achieves greater accuracy at this label benchmark. This clearly shows that self-supervised learning can lead to far superior results with limited data. The results also prove that the accuracy of the supervised method greatly increases with more labeled data, whereas the accuracy of the self-supervised methods only increase slightly. It is important to note that the main reason for this trend is that only the linear head of the self-supervised encoders was fine-tuned to the labels, with

**Table 47.3** Results for linear classification using varying amounts of labeled data

| Model | 10 k labels | 4 k labels | 2 k labels | 1 k labels |
|---|---|---|---|---|
| MoCo | 81.80 ± 0.11 | 80.64 ± 0.21 | 79.78 ± 0.28 | 78.87 ± 0.19 |
| Autoencoder with augmentation | 62.64 ± 0.46 | 60.73 ± 0.46 | 58.42 ± 0.54 | 56.22 ± 0.80 |
| Fully supervised | 83.10 ± 0.11 | 74.12 ± 0.59 | 62.96 ± 1.32 | 49.45 ± 0.75 |
| Dual head (alpha = 0.5) | 67.00 ± 0.21 | 64.56 ± 0.17 | 62.42 ± 0.26 | 59.57 ± 0.50 |

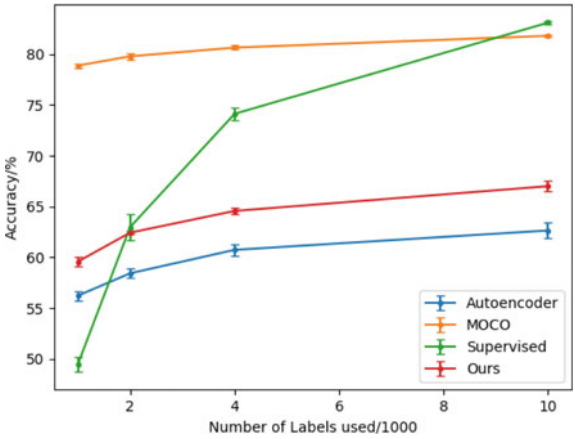


**Fig. 47.9** Graph of classification accuracy against amount of labeled images, including all accuracy with all 50 k labels

the rest of the network remaining frozen. This is also the reason for the accuracy with all the labels being greater with the supervised method. We surmise that the classification accuracy would be even greater if the entire model was unfrozen and allowed to be fine-tuned on the labels, rather than only the final layer, as such results have been shown with other self-supervised training methods on the ImageNet dataset [7]. Using the pre-trained self-supervised weights as an initialization and training the entire model would likely surpass the supervised baseline by a significant margin at all label benchmarks (Table 47.3 and Fig. 47.9).

## 47.8 Conclusion

In this paper, we implemented MoCo and 3 variations of autoencoders for self-supervised representation learning. We verify that contrastive self-supervised learning is highly effective for learning deep image representations, with performance on image classification coming close to the supervised benchmark. Our study also shows that contrastive self-supervised learning is also effective with lower resolution images (32 × 32 images in CIFAR-10, compared to 224 × 224 images in the more commonly used ImageNet) and with shallower models (ResNet18 rather than the standard ResNet50). We also proposed a method for integrating contrastive

methods with generative methods, through adding a secondary head. Although Dual Head networks do not provide a promising result, we believe that it is due to the combination of Autoencoder and MoCo, which results in similar latent representations thus preventing it from being able to achieve higher accuracy. Due to similar representations being learnt despite the networks using two different methods for self-supervised learning, we also believe that there is a need for models to focus on smaller features as well.

## 47.9 Future Work

Even though our dual head network using a decoder head did not help performance, we believe that more research can be done in combining generative and contrastive methods for self-supervised learning, as well as training the encoder to recognize alternative representations as well. The decoder could be replaced with other pretext tasks, such as a self-supervised CutMix [10] which criteria is how accurate the encoder can recognize the replaced section of the image allowing it to learn a diversity of representations.

## References

1. He, K., Fan, H., Wu, Y., Xie, S., & Girshick, R. (2020, March 23). Momentum contrast for unsupervised visual representation learning. Retrieved December 21, 2020, from https://arxiv.org/abs/1911.05722
2. Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020, March 30). A simple framework for contrastive learning of visual representations. Retrieved June 30, 2020, from https://arxiv.org/abs/2002.05709
3. Grill, J., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., . . . Valko, M. (2020, June 13). Bootstrap Your Own Latent: A New Approach to Self-Supervised Learning. Retrieved June 30, 2020, from https://arxiv.org/abs/2006.07733
4. Chen, X., & He, K. (2020, November 20). Exploring simple Siamese representation learning. Retrieved December 26, 2020, from https://arxiv.org/abs/2011.10566
5. Zhang, R., Isola, P., & Efros, A. (2016, October 05). Colorful Image colorization. Retrieved December 24, 2020, from https://arxiv.org/abs/1603.08511
6. Gidaris, S., Singh, P., & Komodakis, N. (2018, March 21). Unsupervised Representation Learning by Predicting Image Rotations. Retrieved December 24, 2020, from https://arxiv.org/abs/1803.07728
7. Hénaff, O., Srinivas, A., De Fauw, J., Razavi, A., Doersch, C., Eslami, S., & Oord, A. (2019, December 06). Data-efficient image recognition with contrastive predictive coding. Retrieved June 30, 2020, from https://arxiv.org/abs/1905.09272
8. Wu, Z., Xiong, Y., Yu, S., & Lin, D. (2018, May 05). Unsupervised feature learning via non-parametric instance-level discrimination. Retrieved December 26, 2020, from https://arxiv.org/abs/1805.01978v1

9. He, K., Zhang, X., Ren, S., & Sun, J. (2015, December 10). Deep Residual Learning for Image Recognition. Retrieved December 22, 2020, from https://arxiv.org/abs/1512.03385
10. Yun, S., Han, D., Oh, S., Chun, S., Choe, J., & Yoo, Y. (2019, August 7) CutMix: Regularization strategy to train strong classifiers with localizable features. [online] Retrieved December 28, 2020, from https://arxiv.org/pdf/2003.05991.pdf