

# Self-Supervised Learning for Image Classification with Limited Labeled Data

Noel Nilsson

Umeå University

Deep Learning: Methods and Applications

March, 2025

## **Abstract**

This report explores the implementation and effectiveness of a self-supervised learning approach for image classification tasks with limited labeled data. The project uses contrastive learning to pre-train a MobileNetV2 model on the CIFAR-10 dataset without labels. The pre-trained encoder is then fine-tuned by training on labeled data with supervised learning using different percentages of labeled data (1%, 5%, 10%, 20%, 30%, 50%, and 100%). The result shows that self-supervised learning can improve performance with less labeled data. Results showed that 20% labeled data fine-tuning on a pre-trained model is comparable to a supervised model trained on 100% labeled data. This suggests that self-supervised learning can be an effective way to train models with limited labeled data.

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>Method</b>	<b>5</b>
2.1	Dataset . . . . .	5
2.2	Data preprocessing . . . . .	5
2.2.1	Data augmentation . . . . .	5
2.3	Model Architecture . . . . .	5
2.4	Training . . . . .	6
2.4.1	Self-supervised pre-training . . . . .	6
2.4.2	Supervised Fine-Tuning . . . . .	7
2.5	Baseline Supervised Model . . . . .	7
2.6	Evaluation . . . . .	7
<b>3</b>	<b>Results</b>	<b>8</b>
3.1	Self-Supervised Pre-Training . . . . .	8
3.2	Fine-Tuning with Limited Labeled Data . . . . .	8
3.3	Comparison with Supervised Baseline . . . . .	9
<b>4</b>	<b>Discussion</b>	<b>10</b>
4.1	Self-supervised training time . . . . .	10
4.2	Limitations and Challenges . . . . .	10
4.3	Other Solutions . . . . .	11
4.3.1	MoCo . . . . .	11
4.3.2	BYOL . . . . .	11
<b>5</b>	<b>Conclusion</b>	<b>12</b>

# 1 Introduction

Deep learning has improved computer vision in many areas, for example image classification, object detection, and segmentation. To achieve high performance, deep learning requires large amounts of labeled data for supervised learning. Labeled data can be expensive, time-consuming, and difficult to make. This presents a challenge for real-world applications where labeled data can be scarce.

Self-supervised learning has recently become a promising approach to reducing the amount of labeled data required [1]. Instead, self-supervised learning uses a pretext task to learn meaningful representations from unlabeled data. The model can then be fine-tuned on a small amount of labeled data and potentially achieve comparable performance to a fully supervised learning model [4].

This project implements and evaluates a self-supervised learning approach for image classification using contrastive learning. The main objectives are:

- Implementing a self-supervised learning model using contrastive learning and data augmentation.
- Test the pre-trained model on various percentages of the labeled data.
- Compare the result with a fully supervised learning model.

This project uses the CIFAR-10 dataset [5]. The dataset consists of 60,000 32x32 color images in 10 classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck.



Figure 1: CIFAR-10: classes

## 2 Method

### 2.1 Dataset

The CIFAR-10 dataset consists of 60,000 32x32 color images in 10 classes with 6,000 images per class [5]. The data set is divided into training and test images. The training has 50,000 images, and the test has 10,000 images.

A subset of the labeled training data was created in varying percentages to evaluate the effectiveness of self-supervised learning. The percentages are: 1%, 5%, 10%, 20%, 30%, 50%, and 100%. The subsets were created randomly to maintain class balance.

Due to a problem with the 10,000 test images, the training data was split into 40,000 training images and 10,000 test images.

### 2.2 Data preprocessing

The model was pre-trained using a contrastive learning approach inspired by SimCLR [1]. Data augmentation is used to create different views of the same images. Having different augmented views of the same image helps the model learn that the augmented views should have similar representations regardless of their differences. The model thereby teaches itself to focus only on the important features, resulting in the model having a good understanding of the images.

#### 2.2.1 Data augmentation

Images are augmented to create different views of the same image by this augmentation pipeline [1]:

- Random cropping followed by resizing to 96x96 pixels
- Random brightness adjustment
- Random contrast adjustment
- Random saturation adjustment
- Random hue adjustment
- Random horizontal flipping

### 2.3 Model Architecture

The model used the MobileNetV2 architecture as the backbone [6]. MobileNetV2 is a convolutional neural network consisting of 2,257,984 parameters. The backbone was initialized with imagenet weights.

The encoder architecture consists of:

- MobileNetV2 backbone (without the top classification layer)



Figure 2: Data augmentation examples

- Global Average Pooling layer
- Projection head with two dense layers:
  - Dense layer (units = 256, ReLU activation)
  - Batch normalization layer
  - Dense layer (units = 128, output)

Once the pre-training is completed, the projection head is replaced with a classification head to be fine-tuned on the labeled data:

- Dropout layer (rate = 0.5)
- Dense layer (units = 10, with softmax activation)

## 2.4 Training

The training procedure consists of two parts. The first part is self-supervised pre-training on augmented images, and the second part is supervised fine-tuning on labeled data.

### 2.4.1 Self-supervised pre-training

During this training stage, the model learns important features from unlabeled images using conservative learning. The model was pre-trained for 75 epochs with a batch size of 128. Here is the process:

- Each training step is trained on the two augmented images of each original image in the batch (256 augmented images).
- The two augmented images are encoded into a 128-dimensional embedded vector.
- Compute the contrastive loss of pairs of augmented images and pairs with different original images.
- Update weights in the encoder using Adam optimizer (learning rate of 0.001)

The contrastive loss function is a normalized temperature-scaled cross-entropy loss function [1]. Its goal is to push pairs with the same original images closer and pairs with different original images away in the embedded vector space.

### 2.4.2 Supervised Fine-Tuning

Once the encoder is pre-trained with contrastive learning, the encoder is fine-tuned for image classification. The encoder is trained on different percentages of labeled data. The model was fine-tuned for 50 epochs with early stopping if validation accuracy did not increase for four consecutive epochs. Here is the process:

- Load the pre-trained encoder weights
- Freeze the first 100 layers of the MobileNetV2 backbone
- Add a classification head on top of the encoder
- Train the model on labeled data using cross-entropy loss
- Use Adam optimizer to update weights (learning rate of 0.0001)

The first 100 layers of the model were frozen during fine-tuning to keep all the important features it learned during pre-training. The early layers were frozen because they capture general features like edges and colors, while the deeper layers handle the specific classification representations. By freezing the first layers, the model gets a strong foundation followed by a task-specific head. Additionally, a reduced learning rate was used when validation accuracy plateaued.

## 2.5 Baseline Supervised Model

For comparison, a supervised model was trained without encoder weights using 100% of the labeled data. The model has the same architecture as the fine-tuned model and is initialized with imagenet weights. The model was trained similarly as the supervised fine-tuning but with all layers trainable.

## 2.6 Evaluation

The models were evaluated using the classification accuracy on the validation set. To understand the effectiveness of self-supervised pre-training, models are compared with different percentages of labeled data during supervised fine-tuning together with the baseline supervised model.

## 3 Results

### 3.1 Self-Supervised Pre-Training

The initial contrastive loss during self-supervised pre-training is 0.7883 loss. The loss then decreases rapidly in the beginning and then gradually stabilizes towards the end of the training process. The final training loss is 0.0209, indicating that the model learned meaningful presentations from the unlabeled data. Figure 3 shows the contrastive loss during training.



Figure 3: Contrastive loss over 75 epochs

### 3.2 Fine-Tuning with Limited Labeled Data

In total seven models were trained on different percentages of labeled data and one fully supervised model. Figure 4 shows the training accuracy over the epochs for every model. By analyzing the plots, a smaller amount of labeled data results in the model overfitting the training data. This suggests that either the training dataset is not fully representing the dataset or an adjustment in regularization is needed. By examining the models trained on more labeled data, the overfitting problem disappears, indicating the regularization is balanced.

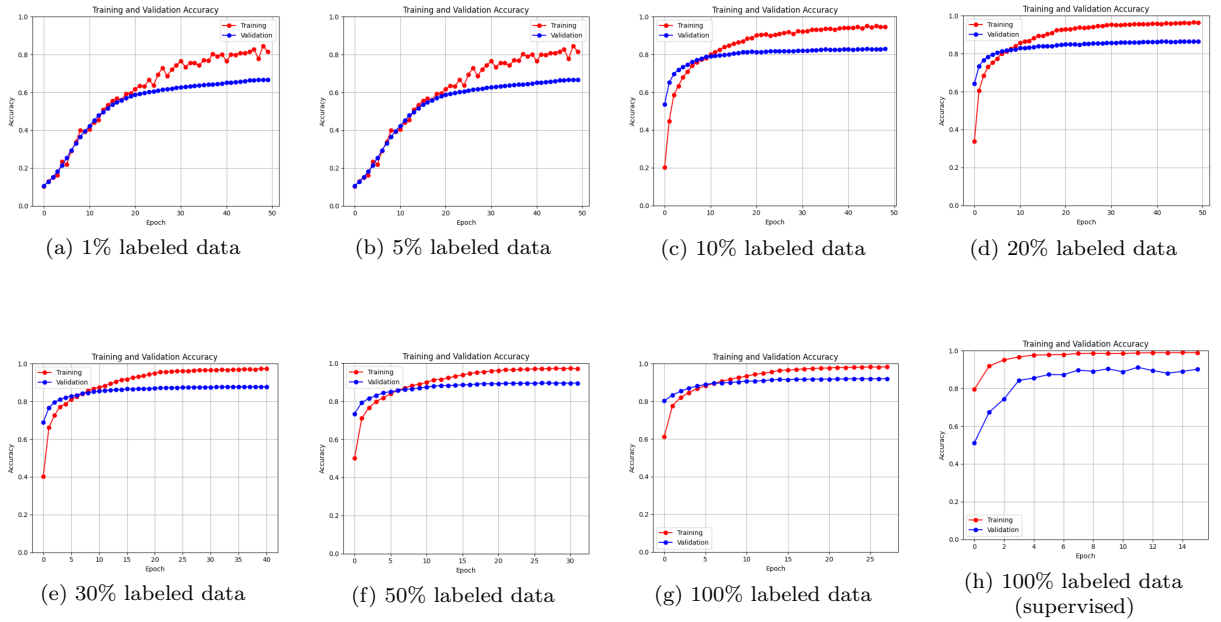


Figure 4: Training accuracy plots for the 8 models

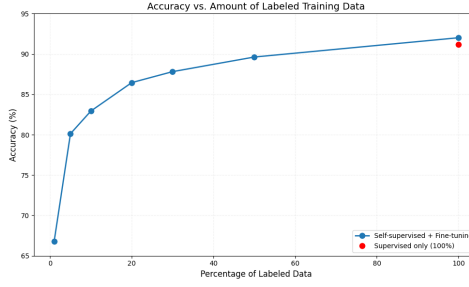


Figure 5: Accuracy vs % labeled data

Table 1 shows the validation accuracy of the eight models. Figure 5 shows a graph of labeled data used in training and the validation accuracy the model achieved. This plot shows how rapidly the validation accuracy increases with a small amount of labeled data. The accuracy then stabilizes and increases slowly. This showcases that a self-supervised model does not require a large number of labeled data to perform comparably with a supervised model.

Labeled Data	Accuracy (%)
1%	66.76
5%	80.12
10%	82.93
20%	86.45
30%	87.81
50%	89.62
100%	92.01
100% (Supervised only)	91.19

Table 1: Validation accuracy (%) with different percentages of labeled data

### 3.3 Comparison with Supervised Baseline

The result showed that the self-supervised model that trained on 100% of the labeled data was best at 92.01%. The model outperformed the supervised model at 91.19%, this indicates the pre-training provided a better initialization of weights to represent important features for the supervised classification.

## 4 Discussion

In this project, self-supervised learning showed its potential by showcasing the ability to learn with a limited data. With only 1% of the labeled data the model managed to achieve an validation accuracy of 66.76%. This achievement is impressive since the labeled data only consists of approximately 40 images per class, whereas the validation data consists of around 1000 images per class.

Self-supervised learning provides great benefits by not requiring a lot of labeled data to perform well. It enables datasets that would be impossible to label feasible by only having to label a part of the dataset. Depending on how powerful the model needs to be, it might not need a lot of labeled data. Due to the fact that the model increases rapidly in performance in a small number of labeled data, the model might only require 10% labeled data (fig 5).

This learning method has the potential to benefit various things by making it applicable in areas where labeled data is scarce and expensive. An example of this is climate monitoring. A large volume of satellite images exists, and they keep getting larger. The complexity of the climate with different specific environmental conditions in different locations makes it difficult and time-consuming to label data. By using a self-supervised model, the amount of label data can be drastically cut, saving both time and money. This would enable better climate monitoring to prevent disasters such as floods and wildfires [4].

### 4.1 Self-supervised training time

The total training time for pre-training and fine-tuning for the self-supervised model far exceeds the supervised model’s training time. The pre-training has double the batch size since it has two augmented images per original image compared to normal supervised learning. This makes the time per epoch double for the pre-training as for the normal supervised training. Additionally, the self-supervised model needs to be fine-tuned with labeled data, which also adds to the total training time.

### 4.2 Limitations and Challenges

MobileNetV2 was used as the backbone for the model [6]. It is a relatively small, 2,257,984-parameter model but powerful. A more robust model might lead to higher performance by capturing more important representations. Due to limited resources in time and hardware capacity, no other larger backbone models were tested.

CIFAR-10 is a dataset consisting of 60,000 images [5]. Compared to real-world applications, it is small and simple. The effectiveness of self-supervised learning can vary depending on the size and complexity of the data set. Due to limited resources, no other dataset was tested.

As shown in the result, models trained on a smaller amount of labeled data tended to overfit more than models trained on more labeled data. This suggests that model architecture adjustment might lead to better performance depending on how much labeled data the model is fine-tuned on.

## 4.3 Other Solutions

The project started with using rotating images as the pre-training task. Rotating images is a simple task where the training data is rotated and labeled with the degrees rotated. This project used the rotation  $[0, 90, 180, 270]$  degrees. During the pre-training, the model tries to predict the rotations of the images. The model, therefore, has to capture meaningful representations of the dataset to tell the image's rotation. This task was shown not to be effective for this dataset, resulting in a lack of performance.

### 4.3.1 MoCo

Momentum Contrast (MoCo) is a similar self-supervised learning method that builds on top of contrastive learning. MoCo uses previous batches' augmented image features to use as examples of pairs with the different original images. This allows the model to train on more comparisons without needing a larger batch size. MoCo would be suitable for this project since it enables pre-training with larger batch sizes. A larger batch size could potentially increase the model's performance by comparing more pairs in pre-training without requiring too much memory [2].

### 4.3.2 BYOL

Bootstrap Your Own Latent (BYOL) is a different pre-training approach to contrastive learning. During training BYOL does not need examples with different original images. Instead, BYOL uses two networks. An online network that gets updated normally during training. A target network that is a copy of the online network but updates slowly instead.

Through training, BYOL creates two different views of an image; then, the online network tries to predict the first view and what the target network will output for the second view. The online prediction is then trained to match the target network's prediction for the second view. After this, the process is repeated with the views swapped. Interestingly, this acts like a self-prediction game and often outperforms methods that use examples with views from different images. BYOL could be a great and interesting alternative to the chosen method in this project [3].

## 5 Conclusion

This project demonstrated how effective self-supervised learning can be for image classification with limited labeled data. It showed how self-supervised learning was applied to the CIFAR-10 dataset with a backbone model, MobileNetV2 [6]. The contrastive learning approach for the pre-training gave promising results, helping the model to learn important features [1]. This resulted in the self-supervised model achieving comparable performance to a fully supervised model with fine-tuning on a small fraction of the labeled data.

The results indicate that self-supervised learning requires significantly less labeled data. With only pre-training on 20% of the labeled data, the model achieved a validation accuracy of 86.45% after fine-tuning. This result is comparable to the fully supervised model trained on 100% of the labeled data, which achieved 91.19% validation accuracy. Contrastive pre-training also captured important features that a supervised model would not. This was demonstrated when the self-supervised model outperformed the supervised model with 100% labeled data, achieving a performance of 92.01% compared to 91.19%.

The rapid improvement of the self-supervised model performance with a small amount of labeled data, particularly in the 1-20% range, suggests that the model has already captured the most important features and needs to be recalibrated with a classification head. This could be valuable when data is scarce and expensive when building deep learning models.

However, self-supervised learning has limitations to consider. One is that the training time for self-supervised models takes substantially longer because it requires both pre-training and fine-tuning. Additionally, the pre-training is more challenging to implement, and different approaches may be ineffective for certain datasets.

Future work could explore alternative pre-training techniques such as MoCo [2] or BYOL [3]. These learning approaches are more advanced than the contrastive learning implemented in this project and can increase performance. Exploring different backbone architectures and optimizing hyperparameters could also increase the performance.

In conclusion, self-supervised learning is a promising approach to addressing the challenge of limited labeled data in deep learning applications. By utilizing the abundance of unlabeled data, models could be built more efficiently and require significantly less labeled data while maintaining competitive performance.

## References

- [1] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” in *International Conference on Machine Learning*, 2020, pp. 1597–1607.
- [2] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, “Momentum contrast for unsupervised visual representation learning,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 9729–9738.
- [3] J.-B. Grill, F. Strub, F. Altché, C. Tallec, P. H. Richemond, E. Buchatskaya, C. Doersch, B. A. Pires, Z. D. Guo, M. G. Azar, et al., “Bootstrap your own latent: A new approach to self-supervised learning,” *arXiv preprint arXiv:2006.07733*, 2020.
- [4] X. Zhai, A. Oliver, A. Kolesnikov, and L. Beyer, “S4L: Self-supervised semi-supervised learning,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1476–1485.
- [5] A. Krizhevsky, G. Hinton, et al., “Learning multiple layers of features from tiny images,” Technical report, University of Toronto, 2009.
- [6] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, “MobileNetV2: Inverted residuals and linear bottlenecks,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4510–4520.