A down-scaled Self-Supervised approach for image classification: SimCLR

Joseph Beasse*, Mohamed Yassine Kabouri*
Enseirb-Matmeca
1 Avenue du Dr Albert Schweitzer, 33400 Talence, France
{jbeasse, mkabouri}@bordeaux-inp.fr

Abstract

In this work, we propose a down-scaled version of the Self-Supervised Contrastive Learning framework SimCLR, tailored for the CIFAR-10 dataset and using a less complex base model, ResNet18. Our approach adapts contrastive learning techniques for environments with limited computational resources and smaller datasets. We investigate the effect of data augmentations and batch sizes on the learning capabilities of the model. The findings suggest that contrastive learning can still yield significant improvements in feature representations and classification accuracy, offering a promising avenue for efficient learning in constrained scenarios. Finally, we use Vision Transformers (ViTs) to extract features, and we found encouraging results for future investigation.

1. Introduction

Self-Supervised Learning [1] (SSL) has emerged as a powerful paradigm in machine learning, especially for tasks where labeled data is scarce or expensive to obtain, In the context of medical imaging, where the cost of annotating data is significantly high, the participation of human experts becomes imperative. To illustrate the magnitude of this challenge, consider the ImageNet dataset. If an individual were to annotate images at a rate of one per minute without any breaks for two years, including essential activities like sleeping and eating, it would still require 22 years and 10 months to complete the task[1].

The idea behind SSL is to extract and learn representations from data itself. This is done by defining a pretext-task where a Deep Learning model try to learn representations about the unlabeled data by trying to solve this pretext-task. The only reason of defining a pretext-task is to learn representation. A pretext-task apply a transformation to the input data, An example of such transformations includes rotations [11] or jigsaw transformations [10]. The Deep Learning model objective is to predict characteristics of the transformation from the transformed input data. Subsequently, the same Deep Learning model can be used in downstream tasks, such as classification, object detection in a Computer Vision view, with only a fine-tuning step required. Figure 1 summaries the idea behind SSL with pretext-tasks.

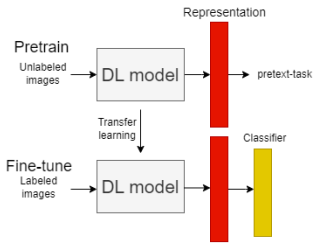

Figure 1. Self-Supervised Learning with pretext-tasks overview.

SSL techniques, particularly those based on Contrastive Learning, have shown remarkable success in learning robust feature representations from unlabeled data. SimCLR, a leading framework in this domain, leverages large-scale datasets and complex models to learn representations that are transferable to various tasks. However, the application of such frameworks in resource-constrained environments remains a challenge. Our study explores the implementation of SimCLR on the CIFAR-10 dataset with ResNet18 as the base encoder. Finally, we trained three Vision Transformer (ViT) models on four classes of ImageNet (5200 images) to see if the properties of ViT will help extract good representations. We detail our methodology, the challenges faced, and the resulting performance enhancements achieved through Self-Supervised Learning. The source code can be accessed at https://github.com/mKabouri/contrastive-learning.

2. Related Work

The advancement of Self-Supervised Learning (SSL) has been crucial in utilizing unlabeled data to learn meaningful representations. The SSL domain has seen a shift from

*Equal contribution
[1] https://www.pinecone.io/learn/series/image-search/imagenet/

heuristic approaches to methods that enable models to understand data semantics autonomously. One of the primary drivers of this shift has been contrastive learning, which has been effective in differentiating between distinct data instances [4, 7, 8].

PIRL, for Pretext-Invariant Representation Learning by Misra et al. [8], used Jigsaw as a pretext-task. The Jigsaw task involves dividing the image into nine patches and introducing perturbations by randomly permuting these patches. Earlier studies employed the Jigsaw [10] task as a pretext-task, involving predicting the permutation from the perturbed input image. This necessitates the learner to build a representation that is covariant to the introduced perturbation. Misra et al. [8] incorporate the established Jigsaw pretext-task in a manner that promotes the invariance of image representations to the perturbation of image patches. Moreover, PIRL uses a memory bank of negative samples to be used in the contrastive learning.

SimCLR by Chen et al. [4] marked a significant milestone by demonstrating that the choice of data augmentations and architectural considerations could lead to representations rivaling those obtained by supervised learning. The framework's ability to scale with increased batch sizes opened up new directions in training deeper models with large datasets. However, the reliance on large batch sizes poses challenges in resource-constrained scenarios, motivating the exploration of more efficient training methods.

Complementing the contrastive paradigm, He et al.'s MoCo [7] introduced the concept of a momentum-based moving average encoder, reducing the necessity for large batch sizes and enabling a more accessible entry point for SSL in environments with limited computational capability. This approach aligns with our interest in adapting SSL methods for smaller datasets and less powerful models.

While contrastive methods have dominated the SSL landscape, alternative approaches like BYOL [6] offer a different perspective by eliminating the need for negative pairs in the learning process. This non-contrastive approach, along with clustering-based methods like those proposed by Caron et al. [2], showcases the breadth of strategies in the field and informs our understanding of the versatility of SSL.

Our investigation is situated in this context of expanding SSL beyond large-scale environments. By integrating insights from leading methods and adapting them to a constrained setting, our work seeks to contribute to the democratization of SSL, making it more applicable and relevant to a wider array of real-world scenarios where computational resources and labeled data are limited.

3. Methodology

3.1. Contrastive Learning Framework

Our research adapts the SimCLR framework to a more accessible computational setting, utilizing a scaled-down version that employs a ResNet18 model and the CIFAR-10 dataset. The primary goal is to maintain the essence of contrastive learning while adjusting the scale to suit environments with limited resources.

Contrastive learning involves learning to encode similar (positive) pairs closer together in the representation space while pushing dissimilar (negative) pairs further apart. This objective is realized through a contrastive loss function, which operates on pairs of augmented images derived from the same source image, referred to as positive pairs, and augmented images from different source images, or negative pairs. The challenge lies in selecting augmentations that preserve the critical features of the images while providing enough variation to facilitate robust learning.

In our framework, we generate positive pairs through a set of stochastic data augmentation techniques, including random cropping, horizontal flipping and color jittering. Each image in a batch is passed through these augmentations to create two correlated views, which the model then projects into a shared representation space using a convolutional neural network (resnet18) followed by a multi-layer perceptron (MLP). The contrastive loss is applied to these representations, pulling together the embeddings of positive pairs and pushing apart those of negative pairs across the batch.

The innovation of our approach lies in the adaptation of the contrastive learning process for smaller datasets and models. By fine-tuning the balance between model capacity and dataset complexity, we demonstrate the potential of self-supervised learning even in settings that traditionally lack the scale of data or computational power presumed necessary for such tasks.

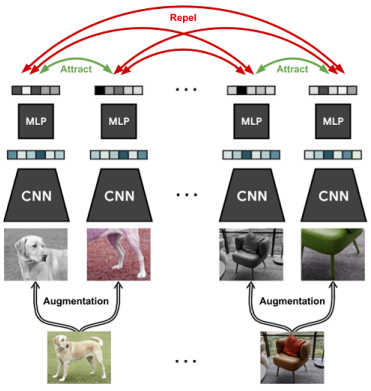

Figure 2. Visualization of the contrastive learning process with positive and negative pairs.

The effectiveness of our model is further explored through various experiments, assessing the quality of the learned representations and their transferability to downstream tasks.

3.2. Contrastive Learning Process

The contrastive learning process is graphically represented in Figure 2, which encapsulates the core idea of this self-supervised approach. For each data example x, two correlated views $\tilde{x}_i$ and $\tilde{x}_j$ are generated by applying two distinct augmentation transformations sampled from the same family T. These views are then fed into the base encoder network $f(\cdot)$ to obtain representations h_i and h_j. A projection head $g(\cdot)$ maps these representations to the space where the contrastive loss is applied.

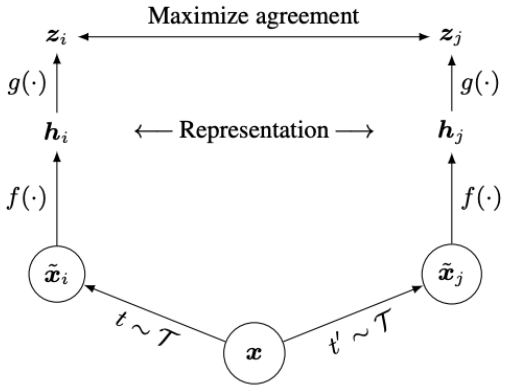

Figure 3. The contrastive learning framework, where two augmented views of the same image are processed through an encoder and a projection head to maximize agreement. Adapted from [4].

The objective is to maximize agreement between the projections z_i and z_j of the positive pair, while minimizing it for negative pairs. The contrastive loss function employed, typically a variant of Noise Contrastive Estimation (NCE), encourages the encoder to learn invariant features under the defined augmentations.

After training, the projection head $g(\cdot)$ is discarded, and the encoder $f(\cdot)$, along with its output representations h, is utilized for downstream tasks. This design ensures that the learned representations are transferable and beneficial for subsequent classification or recognition tasks.

3.3. Contrastive Loss Function

The Contrastive Loss function is the key component of our framework, directing the model to distinguish between similar (positive) and dissimilar (negative) pairs. For a given positive pair of augmented images, z_i and z_j, the loss function is formulated as follows:

$$\mathcal{L}_{i,j} = -\log\left(\frac{\exp(\text{sim}(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} 1_{[k\neq i]}\exp(\text{sim}(z_i, z_k)/\tau)}\right)$$

In this equation:

- $\text{sim}(z_i, z_j)$ is the cosine similarity between the representations z_i and z_j, which is computed as the dot product between the normalized vectors of z_i and z_j.
- τ is the temperature scaling parameter, it is a crucial hyperparameter that adjusts the separation of positive pairs from negative ones.
- The denominator sums over all negative pairs, ensuring that z_i is contrasted with every other representation in the batch except for itself. The indicator function $1_{[k\neq i]}$ is 1 for all $k \neq i$ and 0 otherwise, preventing a trivial solution where all representations collapse into a single point.

The contrastive loss hence encourages the model to pull together the representations of positive pairs while pushing apart those of negative pairs. This self-supervised task ensures that the model learns robust feature representations that are beneficial for downstream tasks such as classification, even in the absence of explicit labels.

4. Dataset and Preprocessing

Our framework operates on the CIFAR-10 dataset, which comprises 60 000 images of dimensions 32×32, categorized into ten classes. 50 000 of those images are in the training set and 10 000 in the test set.

Data augmentation serves an important role in the context of contrastive learning, particularly when dealing with diverse image datasets. The objective is to simulate realistic transformations that an image might naturally encounter. This approach is guiding the model to discern robust features that are invariant to such changes and to abstract away misleading cues. For instance, in a dataset where images of cats are frequently accompanied by grass, the model might erroneously associate the green background with the concept of 'cat'. Through strategic augmentations—such as color jittering—our method aims to mitigate this bias by diminishing the model's reliance on color as a distinguishing feature. By doing so, the model is encouraged to focus on more stable and generalizable attributes.

The data augmentation strategy implemented is as follows:

Algorithm 1 Selection of a data augmentation composition

```
r ← select random value in [0, 1[
if r < 0.5 then
  return Compose([
    Crop(scale=(0.7, 1.0), ratio=(0.8, 1.2)),
    Resize(size=ORIGINAL_SIZE),
    RandomHorizontalFlip(proba=0.3)
  ])
else
  return Compose([
     RandomHorizontalFlip(0.3),
     ColorJitter(brightness=0.1, contrast=0.1, saturation=0.1, hue=0.05)
  ])
end if
```

The algorithm dynamically constructs an augmentation pipeline by random selection, ensuring that each image is exposed to a range of transformations. This variability is crucial for self-supervised learning, where the model is encouraged to learn invariant and discriminative features without relying on labels.

To preprocess the images, we first resize them to a uniform dimension of 224×224 to accommodate the architecture of our downscaled SimCLR model (and thus the layers of the resnet18 models). We then normalize the images based on the calculated mean and standard deviation values across the dataset, which aids in model convergence by ensuring that the input data distribution is centered and scaled appropriately.

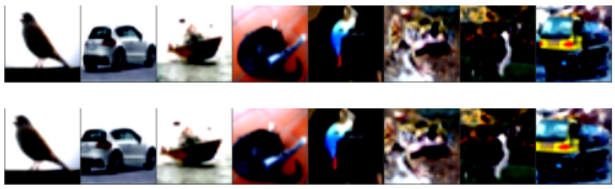

Figure 4. Original vs. Random Crop

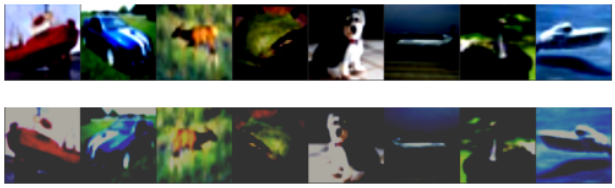

Figure 5. Original vs. Color Jitter & Horizontal Flip

Figures 4 and 5 illustrate the effect of the selected transformations on sample images. Random cropping and resizing induce spatial variability, while color jittering and horizontal flipping simulate changes in color dynamics and orientation, respectively. These transformations are carefully chosen and fine-tuned to ensure that the model learns from meaningful alterations without being misled by overly aggressive distortions that could degrade the learning process.

4.1. Model Architecture

Our model architecture is inspired by the Siamese network design, which is particularly effective for learning from pairs of examples. It is structured to employ a ResNet18 base encoder followed by a projection head, which consists of fully connected layers. This setup is pivotal in the contrastive learning framework, as it serves to encode the input images into a representation space where contrastive loss can be effectively applied.

The ResNet18 encoder captures the essential features of the input images through its deep residual learning framework. This choice of network is due to its ability to learn rich representations with a relatively low computational demand, making it suitable for down scaled applications like ours. The encoder transforms each input image into a feature representation h_i, which encapsulates the informative patterns necessary for distinguishing between different images.

The projection head further maps these representations into a space where the contrastive loss can maximize the agreement between different augmentations of the same image and minimize it between different images. The projection head consists of multiple layers, each adding a level of abstraction to the representation, culminating in the output z_i, which is used for the contrastive loss calculation.

This architectural choice facilitates the learning of representations that are invariant to the augmentations applied to the input images, which is a core objective of contrastive learning methods. The following figure illustrates the flow of data through our model:

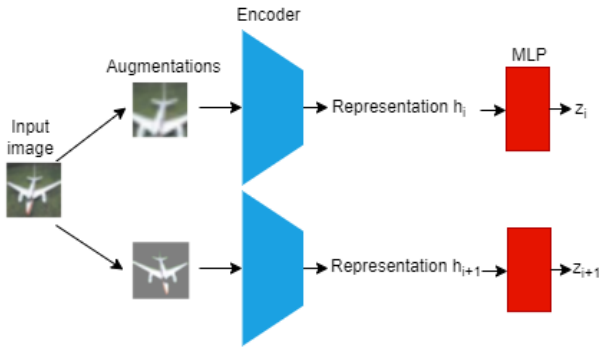

Figure 6. The Siamese network architecture with a ResNet18 encoder and a projection head.

4.2. Model Training

The training process is a critical component of our implementation, focusing on optimizing the encoder and the projection head to minimize contrastive loss. The Adam optimizer was chosen for its efficiency, simplicity of implementation and its adaptive learning rate capabilities, which are crucial for the convergence of our model.

Our training algorithm operates on batches of data, thus choosing an appropriate batch size is very important for the convergence of our model. It not only influences the stability and speed of the training process but also affects the generalization ability of the model. A larger batch size provides more negatives samples and more accurate estimate of the gradient, but it also requires more computational resources and can lead to a sharper, possibly less generalizable minima. Our experiments underscore the critical balance required in choosing a batch size that facilitates a steady decline in loss, ensuring efficient model training without compromising the quality of the learned representations, the results are shown later in the paper.

We apply two distinct augmentations to each image to generate a pair of correlated views. These augmented images are then fed into the model, which computes their representations. The contrastive loss is calculated for each pair, encouraging the model to learn to minimize the distance between representations of augmentations from the same image while maximizing the distance between different images. The batch loss is the average of these contrastive losses, which is used to perform a gradient descent step to update the model parameters.

The pseudocode for our training process is outlined in Algorithm 2, which details the operations performed in each training epoch.

Algorithm 2 Training process

for batch **in** train_data **do**
  $batch\_loss \leftarrow 0$
  for k **in** $range(len(batch))$ **do**
    $augm\_image[2k] \leftarrow augmentation1(batch[k])$
    $augm\_image[2k + 1] \leftarrow augmentation2(batch[k])$
    $z[2k], z[2k + 1] \leftarrow model(augm\_image[2k], augm\_image[2k + 1])$
  end for
  for $k \in range(len(batch))$ **do**
    Calculate $L[2k, 2k + 1]$ and $L[2k + 1, 2k]$
  end for
  $batch\_loss \leftarrow \frac{1}{2*len(batch)} \sum_{k=0}^{len(batch)} L[2k, 2k+1] + L[2k + 1, 2k]$
  Update encoder and projection head
end for

The evolution of the contrastive loss over the training epochs is depicted in the graph below. As illustrated, the loss consistently decreases, indicating that the model is learning effective representations over time.

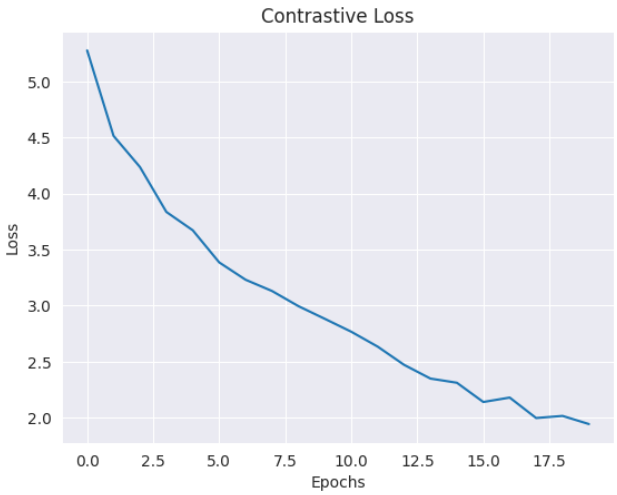

Figure 7. Contrastive loss over 20 epochs, demonstrating the model's learning progression.

This graph validates the effectiveness of our training regime, showing a clear trend of improvement as the model becomes better at distinguishing between positive and negative pairs.

5. Experiments and Results

In our pursuit to replicate and adapt the SimCLR framework for a downscaled model and dataset, we meticulously followed the methodologies detailed in the original SimCLR paper [4]. We aimed to investigate whether the benefits of contrastive learning could be preserved when applied to a smaller-scale problem, such as the CIFAR-10 dataset, using a ResNet18 model as opposed to the larger ResNet50.

5.1. Model Evaluation

To evaluate our model, we employed two main visualization techniques: t-SNE and nearest neighbor search. These methods provided us with insights into the quality of the features learned by our model.

5.1.1 t-SNE Visualization

t-Distributed Stochastic Neighbor Embedding [9] (t-SNE) allowed us to visualize the high-dimensional feature vectors in a two-dimensional space. The resultant plots, however, did not exhibit well-defined clusters for different classes as anticipated, indicating that our model may not be learning as discriminative features as desired.

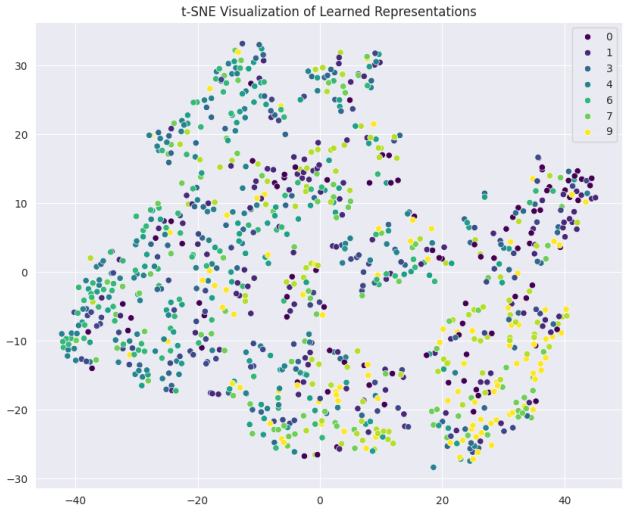

Figure 8. t-SNE visualization of learned representations.

5.1.2 Nearest Neighbor Search

Similarly, the nearest neighbor search was employed to assess the feature representation capabilities of the model. The idea was to verify if the representations could cluster similar images together. As depicted in Figure 9, the closest images were not always of the same class, suggesting that the feature space was not separating the classes effectively.

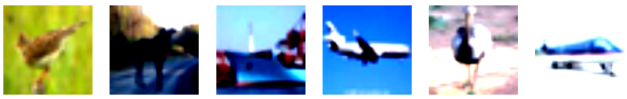

Figure 9. Nearest neighbor search results.

5.2. Discussion

The experimental outcomes revealed that our downscaled adaptation of the SimCLR framework was unable to replicate the results of the original study. A closer examination suggests that this disparity could stem from the intrinsic differences in model complexity and dataset characteristics.

Firstly, the choice of ResNet18 as the backbone for our contrastive learning model introduces a reduction in complexity when contrasted with the original ResNet50 architecture. The impact of this simplification is twofold: it not only restricts the model's capacity to encode nuanced features.

Furthermore, the CIFAR-10 dataset, with its lower resolution and reduced variability in comparison to the expansive and diverse ImageNet collection, presents a less challenging landscape for the model. Consequently, the learned representations may lack the sophistication necessary to generalize across a broader spectrum of visual domains.

Lastly, the optimization of the model was constrained by the computational resources at our disposal, leading to a truncated training regimen and limited hyperparameter exploration. The necessity for extended training durations, larger batch sizes emerges as a critical consideration for future researches in this domain.

Our work also aimed to determine if these embeddings can enhance classification accuracy, providing valuable insights for deploying contrastive learning in limited-resource scenarios.

6. Comparative analysis

6.1. Multi Layer Perceptron Head

The Multi Layer Perceptron (MLP) head is a pivotal component of our contrastive learning framework. It serves as the classification layer that maps the representations learned by the encoder to the label space.

6.1.1 Baseline Model

The baseline MLP model is designed as a straightforward feedforward neural network, crucial for establishing a reference point for performance assessment. It comprises a flattening layer that transforms the input into a one-dimensional tensor, followed by a fully connected layer (fc1) that maps the flattened input to a hidden layer with a specified size. The ReLU activation function introduces non-linearity, essential for complex pattern recognition. Finally, another fully connected layer (fc2) projects the features from the hidden layer to the output size, corresponding to the number of classes in the dataset. The simplicity of this architecture is deliberate, providing a clear benchmark against which we measure the efficacy of more complex models.

6.1.2 Enhanced CLR Model

In contrast to the baseline, our enhanced CLR model integrates the embeddings obtained from our SimCLR-based self-supervised learning framework. The MLP head in this configuration receives a rich, abstract representation of the input images, encoded by the base ResNet model and refined through contrastive learning. This approach leverages the distilled knowledge encapsulated within these embeddings, aiming to amplify the model's ability to discern and classify images with a higher degree of accuracy. The expectation is that the enhanced model will not only outperform the baseline in terms of classification metrics but also demonstrate the practical value of self-supervised learning in real-world applications.

6.2. Classification Task

The classification task aimed to leverage a Multi-Layer Perceptron (MLP) to categorize images from the CIFAR-10 dataset (60 000 images). We explored the performance impact of different input representations on the classification accuracy. These representations included raw images, embeddings generated by our downscaled SimCLR model, principal component analysis (PCA) reduced embeddings, and t-Distributed Stochastic Neighbor Embedding (t-SNE) reduced representations.

6.3. Results

The comparative analysis of classification accuracies is depicted in the bar chart below (see Figure 10). The baseline MLP model, trained directly on raw image data, achieved a test accuracy of 35.77%, setting a foundational benchmark for classification performance. Notably, the use of embeddings from our SimCLR model as input to the MLP resulted in a marked improvement, with accuracy climbing to 48.63%. This enhancement underscores the effectiveness of contrastive learning in extracting meaningful feature representations.

Conversely, the MLP trained on PCA-reduced embeddings attained a lower accuracy of 39.67%, indicating a potential loss of critical information during dimensionality reduction. Similarly, t-SNE embeddings, which further compress the feature space into a two-dimensional manifold, yielded an accuracy of 28.72%. This suggests that the nature of the transformation and the level of dimensionality reduction critically influence the classification capabilities of the MLP.

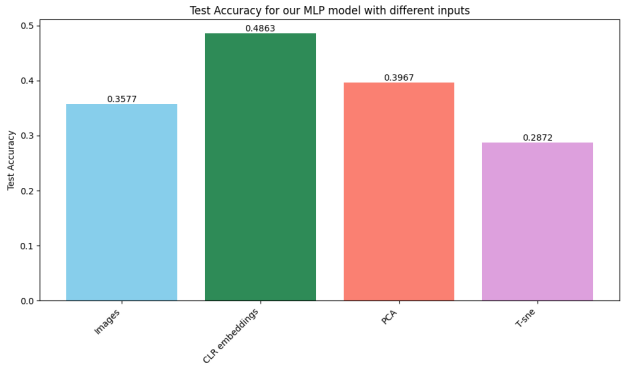

Figure 10. Test accuracy of the MLP model using different input representations. The comparison highlights the impact of feature extraction and dimensionality reduction techniques on classification performance.

These experiments show that the embeddings from our adapted SimCLR model significantly elevate the MLP's performance, validating our approach's effectiveness in this specific downstream task of image classification.

6.4. Batch Size Experiments

The size of the batch during training is a critical hyperparameter in the field of contrastive learning. As outlined in the seminal SimCLR paper, larger batch sizes tend to provide more accurate and stable gradients, potentially leading to better learned representations.

6.4.1 Methodology

In our experiments, we explored the impact of various batch sizes on the performance of our contrastive learning model. We incrementally adjusted the batch size, observing the corresponding changes in the loss landscape and the model's ability to generalize from the learned representations. This iterative process was key to determining the optimal balance between computational resources and learning efficacy.

6.4.2 Results

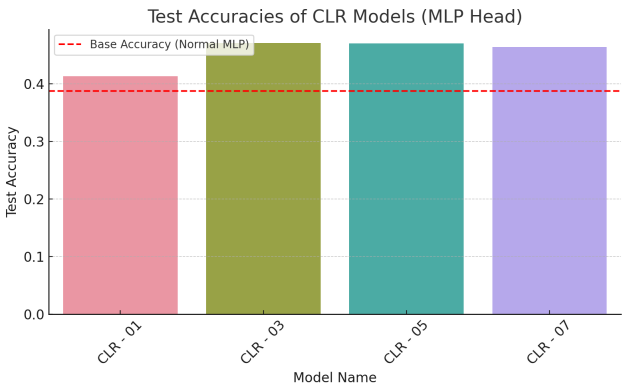

Figure 11. Test accuracies of CLR models with varying batch sizes (The number represents the power of 2 of batch size).

The results, illustrated in Figure 11, demonstrate a clear trend: larger batch sizes are associated with increased test accuracy. This observation is consistent with the assertions made in the original SimCLR research, reinforcing the notion that more extensive batches facilitate superior feature learning. However, it is important to note that increasing batch sizes can overcharge the computational load, often resulting in GPU memory constraints. This limitation prevented us from replicating the larger batch sizes utilized in the original experiments (more than 8192).

7. Vision Transformers

In recent years, transformer models have revolutionized the field of Machine Learning. Originally introduced by Vaswani et al. [12], transformers have become the cornerstone of state-of-the-art models, showcasing unprecedented performance in various tasks such as language translation,

text summarization. In computer vision, Dosovitskiy et al. [5] introduced Vision Transformer (ViT) as an alternative to convolutional neural networks. They necessitate higher computational resources and demand a larger volume of training data. By treating images as sequences of patches and leveraging self-attention mechanisms, ViT excels at capturing long-range dependencies, offering an understanding of the visual content. In DINO by Mathilde Caron et al. [3], they make the observation that features extracted by self-supervised ViT contain explicit information about the semantic segmentation of an image. This pushes us to experiment with ViT to extract representations. We replace ResNet18 by a ViT in the encoder part. We train three ViT models on four classes of ImageNet dataset (5200 images) with the same hyperparameters, except the number of attention heads and the number of layers that we vary as shown in 1. A batch size of 40 and a patch size of 16, chosen for divisibility by 224. The learning rate was set to 0.0001 over 100 epochs for each training. Additionally, a dropout rate of 0.2 and an embedding dimension of 768 were specified.

Heads	Layers	Loss	Training Time
6	6	$3.26 \rightarrow 2.70$	10h 52min
8	8	$3.31 \rightarrow 2.71$	11h 41min
12	12	$3.45 \rightarrow 2.74$	13h 28min

Table 1. Effect of different configurations on model performance.

From table 1, we think that depth and width of a Vision Transformer does not have any impact on training with a small dataset. Because the three models converge to the same loss value after 100 epochs of training. Moreover, We do a fine-tuning step on training data of CIFAR-10, and evaluating the model on the validation data of CIAFR-10. The results

8. Future Research Directions

Subsequent studies may aim to refine the efficiency of self-supervised learning in contexts with limited data or computational resources. The exploration of alternative architectures, augmentation strategies and hyperparameter optimization, could further advance the field. Ultimately, applying these findings to real-world scenarios will underscore the utility of such methods in practical applications.

9. Conclusion

In this paper, we have embarked on an exploratory journey to adapt the SimCLR framework for more resource-constrained environments, specifically targeting the CIFAR-10 dataset with a ResNet18 encoder. Our investigation was motivated by the desire to democratize self-supervised learning, making it accessible and practical for scenarios where computational resources are limited.

Throughout our research, we examined the impact of various hyperparameters on model performance, with a particular focus on the role of batch size in contrastive learning. Our experiments provided insightful revelations about the trade-offs between computational demand and learning efficacy, highlighting the challenges of GPU memory constraints when scaling batch sizes.

Furthermore, we ventured beyond traditional architectures by integrating a Vision Transformer with our contrastive learning framework. This innovative approach yielded promising results, suggesting a potential new direction for future research in self-supervised learning.

Our efforts culminated in valuable contributions to the understanding of self-supervised learning within the constraints of smaller datasets and computational resources. While we achieved a certain level of success in our objectives, the journey does not conclude here. The findings from our study lay the groundwork for subsequent research, inviting further exploration into the optimization of contrastive learning frameworks for various applications.

In closing, we affirm the significance of self-supervised learning as a potent tool for feature extraction and representation learning, particularly in fields where data is scarce or the cost of annotation is prohibitive. As the field continues to evolve, we anticipate a future where these methodologies are not just the privilege of high-resource settings but become a staple across diverse computational landscapes.

References

[1] Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes, Adrien Bardes, Gregoire Mialon, Yuandong Tian, Avi Schwarzschild, Andrew Gordon Wilson, Jonas Geiping, Quentin Garrido, Pierre Fernandez, Amir Bar, Hamed Pirsiavash, Yann LeCun, and Micah Goldblum. A cookbook of self-supervised learning. pages 3–4, 2023. 1

[2] Mathilde Caron, Piotr Bojanowski, Armand Joulin, Matthijs Douze, Ivan Laptev, Cordelia Schmid, and Jean Ponce. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in Neural Information Processing Systems (NIPS)*, 2020. 2

[3] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 8

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 2, 3, 5

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner,

Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *International Conference on Learning Representations (ICLR)*, 2021. 8

[6] Jean-Baptiste Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, et al. Bootstrap your own latent: A new approach to self-supervised learning. *arXiv preprint arXiv:2006.07733*, 2020. 2

[7] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[8] Misra Ishan and van der Maaten Laurens. Self-supervised learning of pretext-invariant representations. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[9] van der Maaten Laurens and Hinton Geoffrey. Visualizing data using t-sne. *The Journal of Machine Learning Research*, page 85, 2008. 5

[10] Noroozi Mehdi and Favaro Paolo. Unsupervised learning of visual representations by solving jigsaw puzzles. *European Conference on Computer Vision (ECCV)*, 2016. 1, 2

[11] Gidaris Spyros, Singh Praveer, and Komodakis Nikos. Unsupervised representation learning by predicting image rotations. *International Conference on Learning Representations (ICLR)*, 2018. 1

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems (NIPS)*, volume 30. Curran Associates, Inc., 2017. 7