Re-implementation and Analysis of SimCLR: A Contrastive Learning Framework

Deha Ay
Data Science Institute
Columbia University
NY, United States
da3109@columbia.edu

Simran Padam
Department of Statistics
Columbia University
NY, United States
sdp2158@columbia.edu

Yixiao Xiong
Department of Statistics
Columbia University
NY, United States
yx2771@columbia.edu

***Abstract*—This research reevaluates and adapts the SimCLR framework (A Simple Framework for Contrastive Learning of Visual Representations), initially introduced by Chen et al., for a practical application in the ECBM 4040 course at Columbia University. Tailored to align with our technical resources and learning goals, our rendition strives to corroborate Chen et al.'s original results and deepen our understanding of contrastive learning in the sphere of deep learning. The paper focuses on the challenges faced during implementation and model refinement and conducts a comparative analysis with established supervised learning approaches. The primary objective of this study is to shine a light on the practicality and relevance of the SimCLR practice, examining its potential as a viable substitute for existing methodologies. While due to computational restrictions we were not able to get close results to the paper, we have acknowledge the potential of the model even with the very limited data and computing power out team held.**

I. Introduction

In our student-led project, we're tackling a popular challenge in machine learning and computer vision: learning to recognize and understand images without needing a lot of labeled data. We're using contrastive learning, a new method that teaches computers to tell the difference between similar and different images. This technique is effective for making computers better at tasks like recognizing what's in a picture or finding objects, without needing or requiring complicated data or memory banks.

Our team takes the study by Chen et al. called "A Simple Framework for Contrastive Learning of Visual Representations" (SimCLR, 2020) as a focal point of this paper. This study was significant because it made learning image representations simpler and more effective. The original paper focused on important methods like data augmentation to vary and enrich the data, using a complex part of the model that can learn on its own (nonlinear transformation), and the benefits of using a lot of data and longer training times. Original authors used this method on a huge dataset called ImageNet and got really impressive results in learning without much supervision with human-created labels.

Compared to other methods in machine learning, like those developed by Hinton, Kingma & Welling, and Goodfellow, SimCLR is a very interesting new option. It's special because it can teach computers to understand images without needing a lot of labeled data. Due to our restrictions in computation power, after multiple trials of trying to utilize the tiny-imageNet-200, we decided to use the CIFAR-10 dataset instead of the bigger ImageNet dataset as a starting point. CIFAR-10 has different types of images but is a much smaller data archive, making it a good choice for testing SimCLR in a more limited way. We hope to show that we can still learn a lot about images even with smaller datasets and less powerful computers.

Our goal is to initially implement the SimCLR method on the CIFAR-10 dataset, (with 10 different classes of images). We want to see how well SimCLR works on this dataset with the limited resources we have, how changing different settings affects the results, and how our findings compare to the original SimCLR study. We hope to prove that Chen et al.'s findings are correct and also to add new information about how well SimCLR works with different types of data and in situations where we don't have as many resources.

II. Literature Review

A. Inspiration

The paper "A Simple Framework for Contrastive Learning of Visual Representations" [1] discusses recent progress in self-supervising representation learning and possible new approach to train self-supervising model in visual representation more efficiently. The authors of the article examined relative works and discovered that most training methods requires specialized model architectures (see [3], [4]) or large memory banks (see [5]). They proposed a new method named **simCLR**, which is a simple framework for contrastive learning of visual representations. The framework adopts the normalized temperature-scaled cross entropy loss, which makes the algorithm more efficient comparing to all the existing complex approach, as it requires neither specialized architechures nor large memory banks. In addition, **simCLR** has better performance regarding the accuracy of self-supervised representation learning on ImageNet.

A main requirement of **simCLR** is that the framework requires large batch sizes during training process. In the paper, they experiment with batch sizes ranging from 256 to 8192, which could be an issue with limited computer capacity. As later discussed.

Another important factor to **simCLR** is that it requires longer training to achieve it's optimal structure. In the paper, the authors train the model with 90, 500, and 1000 epochs for ResNet-50 and ResNet-50 (4x), respectively. However, it is noticable that the improvement in result with respect to longer training process is $\sim 0.5\%$, which suggests that shorter training process may not affect the outcome to a significant degree.

B. Related Work

1) Discussion on Batch Size: Given limited computer capacity, it is beneficial to look for the effect of smaller batch size and shorter training process on contrastive learning of visual representation. In [2], the authors delve into the connection between **simCLR** and batch size. They notice that **simCLR** relies deeply on larger batch sizes during training process. Once the number is decreased, there is a noticeable drop in the final accuracy.

Based on the current contrastive learning model for visual representation learning, the article aims to improve the efficiency to a next level. It notices that the reliance on larger batch sizes is due to the direct adoption of the gradient estimated by mini-batch estimator in **simCLR**, which is a biased estimator that eventually affect the optimization error in **simCLR**. They claim tht **simCLR** suffers from an optimization error at least in the order of $\mathcal{O}(1/\sqrt{B})$ for the objective's gradient norm.Hence, to address the issue, the paper proproses a new approach to the estimation of gradients that will ensure the optimal optimization error. The authors create a memory-efficient Stochastic Optimization algorithm for solving the Global objective of Contrastive Learning of Representations, which they named **SogCLR**.

SogCLR, as the name suggested, optimizes over global contrastive objectives, which allows the model to work efficiently with smaller batch sizes during training process. The authors suggest that the choice of loss function significantly impacts the model's performance and scalability.

2) Discussion on Data Augmentation: It is also worthwhile to explore the relation between contrastive learning of visual representation and data augmentation. The paper [1] includes multiple data augmentation methods.i.e., crop and resize, discoloration, flip, and gaussian blur. In this paper [6], the authors notice the importance of data augmentation for self-supervised learning and acknowledge that simply apply stronger data augmentation could deteriorate the model's performance. So they propose a new framework to address the issue, Contrastive Learning with Stronger Augmentations, **CLSA**.

CLSA adopts a loss function different from both **simCLR** and **SogCLR**. It uses a distributional loss to transfer knowledge from weakly augmented views to strongly augmented views, which improves the distinction between data while maintain the data integrity. The **CLSA** framework can be easily adapt to any concurrent constrastive learning framework, which allows constrastive learning of visual representation to make even better performance.

3) Section Conclusion: It is worth pointing out that the choice of loss function plays a major role in the structure and performance of constrastive learning model. As the three frameworks proposed **simCLR**, **SogCLR**, and **CLSA** all adopt a distinct loss function for their objective. Further study in the connection between loss function and objectives for contrastive learning is suggested.

III. Methodology

With the main goal of learning meaningful and robust representations of data without relying on explicit labels, it addresses the problem of unsupervised representation learning. While traditional supervised learning relies on labeled datasets for model training, acquiring such labeled data can be difficult, costly, or infeasible in numerous real-world scenarios.

SimCLR is designed to extract valuable representations directly from unlabeled data, enhancing its versatility and applicability in situations where labeled data is limited. To put it simply, it trains the encoder for generating representations. This pre-training can serve as a strong initialization for downstream tasks, where fine-tuning on smaller labeled datasets can lead to improved performance. This is particularly advantageous in situations where acquiring labeled data is resource-intensive or impractical.

The SimCLR algorithm involves various steps to learn representations of the data. The aspects of SimCLR include data augmentation, encoder, projection head, and contrastive loss which are described as follows:

1) Data Augmentation: The idea behind data augmentation is to ensure the generalization capabilities of the learned representations. In this approach, two data-augmented images are generated of the same sample. The paper specifies to sequentially apply three augmentations: random cropping and outputting back to the original size of the image, random color distortions, and random Gaussian Blur. The random component enhances the generalizability of the image to learn representations.

2) Encoder Model: Encoder architecture uses a deep neural network based on a Siamese network architecture. [8] Siamese networks learn a similarity function as part of two (or more) identical subnetworks. The encoder maps the input instances into high-dimensional space. In this paper, they have adopted a simple and a popular model called Resnet. During the training process, we utilized the pre-trained weights of the base layers from ResNet, $f(.)$, and the final layer was modified to match the requirements of our specific task. The Global Average Pooling layer, $h_i = f(x_i)$, is finally used on the output.

3) Projection Model: The encoder model's output undergoes processing through a projection head, which constitutes a Multi-Layer Perceptron (MLP), $z_i = g(h_i)$, featuring a single hidden layer using ReLU activation. The main idea for the projection model is to reduce the dimensionality of the representations. In this work, final representations are generated of output size 128.

4) Contrastive Loss: The fundamental concept behind Contrastive learning involves generating positive and negative pairs of samples from the input data, to prompt the model to bring positive pairs closer together in the representation space and simultaneously push negative pairs farther apart. This step is defined as a Contrastive prediction task.

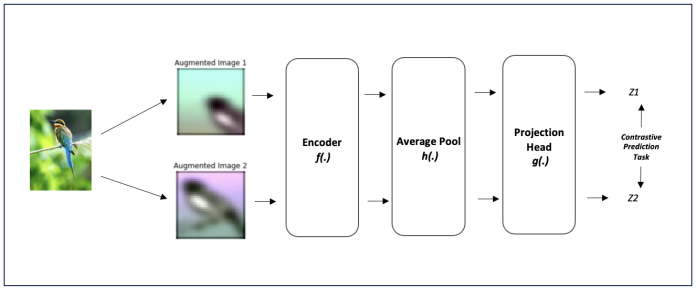

Fig. 1. SimCLR Framework

We utilize these key concepts to learn the representations as shown in Figure 1 [9]. The methodology is applied to random samples of the training images to define a contrastive prediction task. If N random samples are selected, augmentation will produce $2N$ data points. The contrastive learning task is applied on a positive pair which are similar to each other and the rest are considered as negative pairs. The paper uses NT-Xent (Normalized temperature-scaled cross-entropy loss) as the default loss function.

IV. Discussion

A. Comprehensive Analysis of the Low Accuracy in SimCLR CIFAR-10 Implementation

In our application of the SimCLR model to the CIFAR-10 dataset throughout this paper, we observed notably low accuracy levels, roughly around $\sim$ 20%. This section aims to dissect the potential reasons behind this underperformance, drawing comparisons with the original SimCLR implementation detailed in the paper and analyzing deviations in our approach.

Deviations from the Original SimCLR Implementation:

1) ***Training Duration and Intensity****:*

- *Original Paper's Approach:* The SimCLR framework, as described in the original paper, was trained extensively, often spanning hundreds to thousands of epochs. This lengthy training period is pivotal in self-supervised learning paradigms for the model to iteratively refine and optimize its parameters. Since the model is a self-supervised learning that feeds off of data augmentation, the risk of overfitting is very slim, therefore long epochs of training are crucial for higher accuracy results.
- *Our Implementation:* In contrast, our model underwent a significantly truncated training phase, limited number of epochs with 100 batches each. This brief training period is a fraction of what SimCLR typically requires, insufficient for the model to reach its learning potential.

2) ***Dataset and Quality****:*

- *Original Paper's Approach:* Originally, the work was implemented on ImageNet with over 14 million images across more than 20,000 categories. The quality of images is better which ensures better model performance.
- *Our Implementation:* In this paper, we have attempted to replicate all the steps elucidated in the original work. Due to computing resources and limited time constraints, we have initially tried a smaller subset of ImageNet known as tiny-imagenet-200. However the training phase have resulted in the memory issue due to the size; therefore, we substituted with a well-known smaler dataset such as CIFAR10. The CIFAR10 dataset is accessible through the Keras library. Each image is of size (32, 32, 3) with 10 classes such as Bird, Frog, Truck,.etc.

3) ***Batch Size and Sample Diversity****:*

- *Original Paper's Approach:* SimCLR's effectiveness largely stems from its use of large batch sizes. This is essential to providing a diverse range of negative samples, crucial for the model to effectively learn contrastive features.
- *Our Implementation:* Due to computational and memory constraints, we significantly reduced the batch size, directly impacting the model's learning efficiency and its ability to distinguish and classify from a wide range of features.

4) ***Compute Resources and Training****:*

- *Original Paper's Approach:* In the original work, higher compute resources are used such as 128 TPU v3 cores, which take approximately 1.5 hours to train ResNet-50 with a batch size of 4096 for 100 epochs.
- *Our Implementation:* The project necessitated numerous iterations, leading to the training of the models on both the Google Cloud Platform and a local machine. We utilized Python and Tensorflow for implementing the methodology. Despite our efforts, we were unable to secure Virtual Machines with sufficient memory and GPU capabilities in several regions we explored; they were all inactive. Consequently, we primarily relied on the computational resources of our local machines.

5) ***Resolution and Image Detail****:*

- *Original Paper's Approach:* SimCLR in the paper handled high-resolution images, allowing the model to learn from a rich set of pixel details, allowing richer architectures to enhance the encoder space.
- *Our Implementation:* CIFAR-10 images are considerably lower in resolution (32x32). Our model, therefore, had less detailed information to learn from, which strictly affected its ability to extract nuanced features and its adaptability to other datasets.

6) ***Learning Rate and Optimization****:*

- *Original Paper's Approach:* The original SimCLR model was tuned with an optimized learning rate, essential for effective learning over extended epochs.

- *Our Implementation:* The limited number of epochs in our case necessitated a more precise and potentially higher learning rate, which we may not have achieved, affecting the model's performance.

V. Result

In this segment, we showcase the outcomes for the **simCLR** framework, which was trained using various encoders. *Figure 2* displays the highest accuracy we achieved utilizing multiple models. Each encoder undergoes training with a total of 100 batches in every epoch. The batch sizes presented in the subsequent tables indicate the sizes used during the training of the encoders, with the term 'epoch' referring to each epoch of supervised training. Adam optimizer is utilized in the training across all of the models.

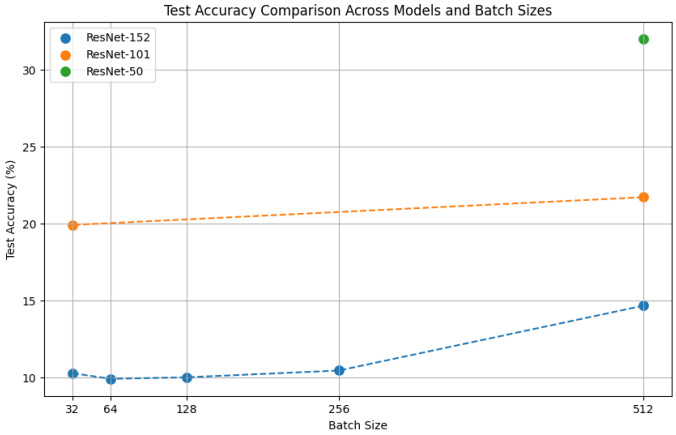

Fig. 2. Accuracy Comparison Across Our Models

A. ResNet-50

For the contrastive learning model with ResNet-50 as its encoder, trained with epoch = 3 and batch size = 512. We achieve the following metrics.

TABLE I
ResNet-50 with Different Classification Model Training Result

Tr Time	Epoch	B.Size	C.Size	Acc. (%)
2 hr	10	512	(20,20)	Tr: 23.0, Tst: 20.0
2 hr	90	512	(20,20)	Tr: 53.0, Tst: **32.0**

As shown, the contrasive representation learning model takes an extensive amount of time to train, yet only results in a training accuracy of 53% and a test accuracy of 32%, much lower comparing to the 64% accuracy in [1]. The cause is due to limited memory and computing capacity, which forced us to adopt a much shorter training period that in turn affects the resulting accuracy. The connection between **simCLR**'s accuracy and training batch size is discussed in Batch Size.

B. ResNet-101

For the contrastive learning model with ResNet-101 as its encoder, trained with different 3 epochs for various batch sizes. We achieve the following metrics in Table II.

TABLE II
ResNet-101 with Different Contrastive Training Results

Tr Time	Epoch	B. Size	C. Size	Acc. (%)
1 hr	10	32	(24,24)	Tr: 12.5, Tst: 12.9
1 hr	10	32	(20,20)	Tr: 19.7, Tst: 19.9
2 hr	10	32	(24,24)	Tr: 20.59, Tst: 19.5
9 hr	10	512	(20,20)	Tr: 21.6, Tst: **21.7**

C. ResNet-152

For the contrastive learning model with ResNet-150 as the encoder, trained for 3 epochs with the 100 batch sizes. We achieve the following metrics:

TABLE III
ResNet-152 with Different Contrastive Training Results

Tr Time	Epoch	B. Size	C. Size	Acc. (%)
1 hr	10	32	(20,20)	Tr: 10.14, Tst: 11.46
1.2 hr	10	64	(20,20)	Tr: 9.9, Tst: 9.09
1.5 hr	10	128	(20,20)	Tr: 10.0, Tst: 10.00
2 hr	10	256	(20,20)	Tr: 10.44, Tst: 10.07
3 hr	50	512	(20,20)	Tr: 19.06, Tst: **16.64**

Low Accuracy Across Batch Sizes: The accuracy for the ResNet-152 model, regardless of the batch size, is consistently around 10%, which is no better than random guessing for a 10-class problem (CIFAR10). This suggests that the model is not learning effectively.

Memory Overpass with Larger Batch Size: In our examination of the ResNet-152 training procedure, we encountered computational limitations. Attempting to train the original model with an increased batch size or more epochs for the encoder resulted in memory constraints.Our based model is trained for 3 epochs with 100 batches on each epoch. ResNet-152, being a more intricate model with deeper layers, typically requires extended training duration and a greater number of epochs to achieve effective convergence. However, in our implementation, surpassing a certain epoch threshold triggered memory issues.

This restriction likely hinders the model's ability to fully exploit its sophisticated learning potential. Additionally, training with a batch size exceeding 512 posed memory challenges, suggesting possible inefficiencies in our approach to data and model management during the training process. This indicates that our current setup might not be ideally configured to meet the demands of more complex models, especially concerning memory utilization and the efficiency of data handling.

VI. Conclusion

In this paper, we delve into the existing frameworks for contrastive learning of visual representation. We focused on **simCLR** while also exploring other frameworks like **SogCLR** and **CLSA**. For our model, we adopt the **simCLR** framework and implement it with cifar10 dataset. We adopt the original loss function, the normalized temperature-scaled cross entropy loss, and trained three different models with ResNet-50, ResNet-101, and ResNet-152 as their encoder, respectively.

We also explored the effect of different training metrics and data augmentation methods.

Due to limited computer capacity, we are unable to achieve the accuracy in the article. Yet this project is an invaluable experience to all of us. Throughout the course of our project, we encountered several technical and interpersonal challenges, including but not limited to the difference in systems, version mismatch, and misunderstanding. Overcoming those issues taught us the importance of effective and frequent communication. As even technical problems can be addressed by clarification and collaboration. This hands-on experience with a large model greatly enhanced our understanding in deep learning and improved our coding and debugging skills. Moreoever, we learned the invaluable lesson on the importance of communication, time management, and teamwork, which would benefit us in our future careers, whether academically or professionally.

ACKNOWLEDGMENT

In this section we want to acknowledge the website or code chunk we used for reference in the report.

For data augmentation, function gauss2D() and function gaussFilter() are constructed on the foundation from these threadgaussian filter in python [7]. Function color_distortion() and color_drop() are based on the pseudo functions listed in the appendix of the paper [1].We also extend our gratitude to the detailed explanation and implementation of the NT-Xent loss function provided by Towards Data Science [10], which greatly aided in our understanding and application of this crucial component in our study.

REFERENCES

[1] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A Simple Framework for Contrastive Learning of Visual Representations," arXiv:2002.05709 [cs.LG], 2020.

[2] Z. Yuan, Y. Wu, Z. Qiu, X. Du, L. Zhang, D. Zhou, and T. Yang, "Provable Stochastic Optimization for Global Contrastive Learning: Small Batch Does Not Harm Performance," arXiv:2202.12387 [cs.LG], 2022.

[3] P. Achman, R. D. Hjelm, and W. Buchwalter, "Learning representations by maximizing mutual information across views," in Advances in Neural Information Processing Systems, pp. 15509–15519, 2019.

[4] O. J. Hénaff, A. Razavi, C. Doersch, S. Eslami, and A. v. d. Oord, "Data-efficient image recognition with contrastive predictive coding," arXiv:1905.09272 [cs.LG], 2019.

[5] I. Misra and L. van der Maaten, "Self-supervised learning of pretext-invariant representations," arXiv:1912.01991 [cs.CV], 2019.

[6] X. Wang and G.-J. Qi, "Contrastive Learning with Stronger Augmentations," arXiv:2104.07713 [cs.LG], 2021.

[7] Stack Overflow. "How to obtain a Gaussian filter in Python," *Stack Overflow*, https://stackoverflow.com/questions/17190649/how-to-obtain-a-gaussian-filter-in-python, 2020.

[8] Towards Data Science. "Siamese-networks". *Towards Data Science*, https://towardsdatascience.com/a-friendly-introduction-to-siamese-networks-85ab17522942

[9] https://medium.com/popular-self-supervised-learning-methods-simclr/popular-self-supervised-learning-methods-simclr-and-swav-74f49d3a267

[10] Towards Data Science. "NT-Xent: Normalized Temperature-scaled Cross Entropy Loss Explained and Implemented in PyTorch." *Towards Data Science*, https://towardsdatascience.com/nt-xent-normalized-temperature-scaled-cross-entropy-loss-explained-and-implemented-in-pytorch-cc081f69848