Analyzing Performance of SimCLR on Different Image Types

Kira Kaplan
Smith College
kakaplan@smith.edu

***Abstract*—Self-supervised learning is an up and coming field of machine learning that relies on an algorithm to train a model to detect patterns from an unlabeled dataset. It's become popular in recent years due to the expensive, and time consuming process of curating labeled, training datasets. However, whether the current technology is at a level where it can sufficiently be deployed as a generalized model remains to be seen. The Simple Framework for Contrastive Learning for Visual Representations (SimCLR) is one algorithm that has shown substantial promise in it's ability to detect and classify objects, even outperforming it's supervised counterparts in some downstream tasks. In this paper a SimCLR model pre trained on an ImageNet dataset is used to classify sets of images from three different TensorFlow datasets (tf_flowers, imagenette, and cifar100). The model performs best on the imagenette dataset and has slight classification errors on the tf_flowers dataset. However, the model repeatedly fails to classify the lower quality images from the cifar100 dataset. This is indicative that SimCLR and other self-supervised learning algorithms may not yet be suitable for direct deployment and need either adjustments to the algorithm to account for low quality images, or to be fine-tuned on a separate dataset before put into use.**

***Index Terms*—Self-supervised Learning; Image Classification; SimCLR**

I. Introduction

Computer vision applications have revolutionized the way we process information and work through datasets. Researchers are now first turning to machine learning and deep learning algorithms to work through their datasets, greatly reducing the manual processing time previously required. These algorithms can mostly be split into two broad categories: supervised, and unsupervised learning. Supervised learning is an approach characterized by the utilization of labeled data to train a model, while unsupervised learning doesn't require labeled datasets and relies on the model itself to detect it's own patterns in the data. While supervised learning has generally been shown to produce better performing models, it requires manual annotations to be made for large datasets. The amount of annotated training datasets available for use is limited and they are time-intensive and expensive to create. Therefore, unsupervised learning approaches are an alternative that is quickly being improved upon to address this issue.

A. Self-supervised Learning

One form of unsupervised learning is what is known as self-supervised learning (SSL). [1] describes this approach as a semi-automatic process in which the model obtains labels from the data itself to then predict part of the data from other parts. It is considered a "recovery" approach where hidden portions of the data are predicted utilizing visible portions [2]. There are two main steps present in SSL that consist of pretext and downstream tasks.

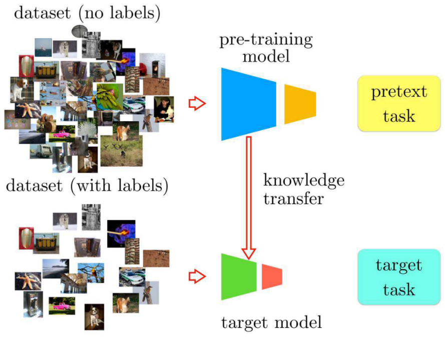

Fig. 1. Fig 1. The basic workflow of self supervised learning. SSL models are trained on unlabeled data and then fine-tuned or tested on their specific downstream test with labeled data [3].

1) Pretext tasks: Pretext tasks are also known as "surrogate" or "proxy" tasks [4] and are a means to achieve the final purpose of the model. They generate pseudo labels for the data through a process P in order to produce a labeled source dataset

$$\bar{D}_s = \{x_i, y_i\} = P(D_s) \tag{1}$$

where $\bar{D}_s$ represents the original, unlabeled dataset, x_i represents an image in the dataset, and y_i represents the pseudo label generated by the pretext task [5]. There are many different algorithms implemented as pretext tasks, some of which are described later in this paper.

2) Downstream tasks: Downstream tasks are sometimes known as "secondary", "primary", or "target" tasks [2] and define the model's main purpose. These refer to tasks such as object detection, object recognition, image classification, semantic segmentation, natural language processing, and human action recognition to name a few [2], [6], [7].

II. Existing Methods

Existing self-supervised learning algorithms can be split up into four main categories that describe the pretext task used including context-based methods, contrastive learning, generative algorithms, and contrastive generative methods [4].

A. *Context-based Methods*

Context-based methods include method such as transformation, jigsaw, and colorization that rely on the contextual relationship between samples.

B. *Contrastive Learning*

Contrastive learning (CL) is focused on measuring the similarity and dissimilarity between images and often relies on methods such as cluster discrimination and instance discrimination [1], [4]. Momentum Contrast (MoCo) and the Simple Framework for Contrastive Learning for Visual Representation (SimCLR) are two CL algorithms that have been shown to outperform their supervised pre-trained model counterparts in some downstream tasks [8], [9].

C. *Generative Algorithms*

Generative algorithms focus on the reconstruction and are specifically used in masked image modeling (MIM) methods [1], [4]. This involves training the model to fill in missing or masked data based on the patterns it learns itself from the data [5]. Algorithms such as the bidirectional encoder representation from image transformers (BEiT) [6], masked autoencoder (MAE) [10], context autoencoder (CAE) [7], and simple framework for masked image modeling (SimMIM) [11] have all shown great promise in their ability to perform downstream tasks such as object detection, semantic segmentation, and image classification.

D. *Contrastive-generative Methods*

Contrastive learning and generative algorithms (specifically MIM methods) have been the dominant players in the field [4], but each has their own downfalls. Contrastive models are prone to overfitting issues, while generative models have difficulty with data scaling and have data-filling challenges [12]. This final category for pretext tasks works to combine the best of both these methodologies as a contrastive generative method. It minimizes the contrastive aspects of samples and then reconstructs the inputs [1], drawing from both contrastive and generative methods. One example of this method is in Generative Adversarial Networks (GAN) [13] which involve a two step training process with the generation of fake samples and then an attempt to distinguish them from real samples [1].

TABLE I

EXAMPLES OF SELF-SUPERVISED ALGORITHMS USED FOR THE FOUR MAIN CATEGORIES OF PRETEXT TASKS.

Context-based	Contrastive Learning	Generative	Contrastive-generative
Colorization	SimCLR [9]	BEiT [6]	iBOT [14]
Jigsaw	MoCos [8]	MAE [10]	CMAE [15]
Transformation	Barlow Twins [16]	CAE [7]	SiameseIM [17]

III. OPEN CHALLENGES

A. *Current Challenges*

Self-supervised learning alleviates many of the barriers surrounding traditional, supervised machine learning, specifically in regards to creating large, expensive training datasets. However, current limitations still exist in the field of self-supervised learning that may deter future use. One of the major challenges right now in self-supervised learning is determining the best approach to use for a downstream task. An optimal SSL algorithm doesn't exist for a specific task, and the decision usually requires much experimentation and consideration of the dataset to determine the best method [4]. Future research is required to investigate a suitable solution for deciding which SSL method to utilize for the best results without running multiple experiments. Additionally, although some SSL algorithms have been shown to outperform their fellow supervised models [6]–[10], less information exists on how SSL models might perform on less visually distinctive and refined datasets, such as those that experience illusion difficulties and blurriness.

B. *Research Ideas*

This study provides a preliminary comparison of the performance of SimCLR on different test datasets. Specifically, it looks at the model's ability to run inference on images from a different dataset than it was trained on and on images that may be of reduced quality. The results of this experiment, evaluated qualitatively, will show whether future improvements to SimCLR and other contrastive learning methods are required to achieve performance levels suited for continued use.

IV. CONCEPT TO CODE

A. *Datasets*

1) Training Dataset: The SimCLR model used for this study was trained by [9] on the ImageNet ILSVRC-2012 dataset. This is a dataset comprised of images from 1,000 object classes for a total of 1.28 million available images [18]. These images span a wide range of different classes, making the SimCLR pre-trained model ideal as a generalized image classifier.

2) Testing Datatsets: The three example datasets used for testing come from TensorFlow. The tf_flowers dataset is comprised of up-close images of flowers including sunflowers, dandelions, daisies, and tulips [19]. The imagenette dataset is a subset of the ImageNet dataset and contains 10 different classes. The final test dataset is cifar100 which consists of 100 classes and 20 superclasses [20]. These images are significantly more blurred than that of the other two datasets.

B. *Running the Code*

The code can be run by going to the following GitHub folder: `https://github.com/MHC-FA24-CS341CV/beyond-the-pixels-emerging-computer-vision-research-topics-fa24/blob/main/code/13-self-supervised-learning` and opening the file in Google Colab. Then select "Run All" from Runtime and wait for the process to complete. The outputted images display the models' predictions alongside the true labels.

C. Results

The SimCLR model performed best at classifying the images from the imagenette dataset (Fig. 2).

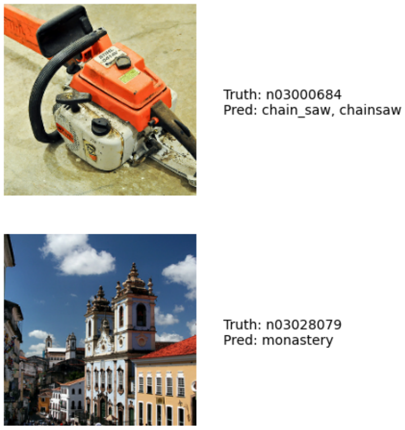

Fig. 2. Results of pretrained SimCLR model classifying images from the imagenette dataset.

This is not surprising since the model was originally trained on the larger ImageNet dataset. This means it might already be familiar with many of the images and the labels of this test dataset. The model had varying results with the tf_flowers dataset. It was generally able to classify an image as some type of flower, although the type of flower was often incorrect (Fig. 3).

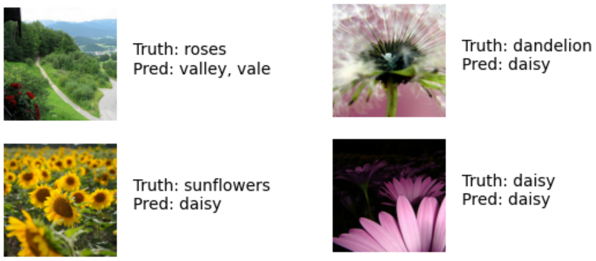

Fig. 3. Results of pretrained SimCLR model classifying images from the tf_flower dataset.

The cifar100 dataset had the worst results, with the model incorrectly classifying almost all of the images. Although most of the classes shown in the test images were similar to those that are labeled in ImageNet, the blurriness of the cifar100 images may have contributed to the model's inability to identify them (Fig 4.).

This indicates that self supervised learning algorithms such as SimCLR still have a ways to go before they are able to perform as ready to use, generalized models. Avenues of improvement should be investigated to account for low quality images as well as for identifying classes that might not have been seen in the original training dataset.

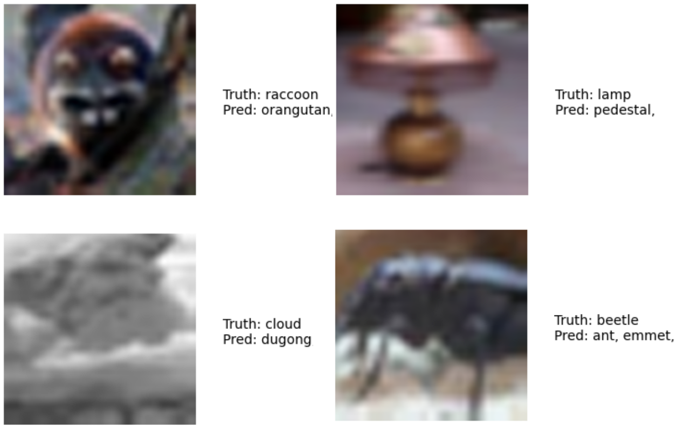

Fig. 4. Results of pretrained SimCLR model classifying images from the cifar100 dataset.

V. Future Work

Next steps may include altering image colorization to see if it impacts the models ability to accurately classify an image. Additionally, more datasets should be included as test subjects to get a more robust sample of SimCLR's performance. Finally, models with different pretext tasks, such as generative algorithms (MIM methods), contrastive-generative methods, and contex-based methods should be explored with the same test datasets used for SimCLR to compare performances on different image types and styles. This is one step closer to defining an optimal SSL algorithm for different downstream tasks.

References

[1] X. Liu, F. Zhang, Z. Hou, L. Mian, Z. Wang, J. Zhang, and J. Tang, "Self-supervised learning: Generative or contrastive," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 1, p. 857–876, Jan. 2023.

[2] V. Rani, S. T. Nabi, M. Kumar, A. Mittal, and K. Kumar, "Self-supervised learning: A succinct review," *Archives of Computational Methods in Engineering*, vol. 30, no. 4, p. 2761–2775, May 2023.

[3] M. Noroozi, A. Vinjimoor, P. Favaro, and H. Pirsiavash, "Boosting self-supervised learning via knowledge transfer," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. Salt Lake City, UT: IEEE, June 2018, p. 9359–9367. [Online]. Available: https://ieeexplore.ieee.org/document/8579073/

[4] J. Gui, T. Chen, J. Zhang, Q. Cao, Z. Sun, H. Luo, and D. Tao, "A survey on self-supervised learning: Algorithms, applications, and future trends," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 46, no. 12, p. 9052–9071, Dec. 2024.

[5] L. Ericsson, H. Gouk, C. C. Loy, and T. M. Hospedales, "Self-supervised representation learning: Introduction, advances, and challenges," *IEEE Signal Processing Magazine*, vol. 39, no. 3, p. 42–62, May 2022.

[6] H. Bao, L. Dong, S. Piao, and F. Wei, "Beit: Bert pre-training of image transformers," no. arXiv:2106.08254, Sept. 2022, arXiv:2106.08254. [Online]. Available: http://arxiv.org/abs/2106.08254

[7] X. Chen, M. Ding, X. Wang, Y. Xin, S. Mo, Y. Wang, S. Han, P. Luo, G. Zeng, and J. Wang, "Context autoencoder for self-supervised representation learning," no. arXiv:2202.03026, Aug. 2023, arXiv:2202.03026. [Online]. Available: http://arxiv.org/abs/2202.03026

[8] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," no. arXiv:1911.05722, Mar. 2020, arXiv:1911.05722. [Online]. Available: http://arxiv.org/abs/1911.05722

[9] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A simple framework for contrastive learning of visual representations,” no. arXiv:2002.05709, July 2020, arXiv:2002.05709. [Online]. Available: http://arxiv.org/abs/2002.05709

[10] K. He, X. Chen, S. Xie, Y. Li, P. Dollár, and R. Girshick, “Masked autoencoders are scalable vision learners,” no. arXiv:2111.06377, Dec. 2021, arXiv:2111.06377. [Online]. Available: http://arxiv.org/abs/2111.06377

[11] Z. Xie, Z. Zhang, Y. Cao, Y. Lin, J. Bao, Z. Yao, Q. Dai, and H. Hu, “Simmim: A simple framework for masked image modeling,” no. arXiv:2111.09886, Apr. 2022, arXiv:2111.09886. [Online]. Available: http://arxiv.org/abs/2111.09886

[12] Z. Qi, R. Dong, G. Fan, Z. Ge, X. Zhang, K. Ma, and L. Yi, “Contrast with reconstruct: Contrastive 3d representation learning guided by generative pretraining,” no. arXiv:2302.02318, May 2023, arXiv:2302.02318. [Online]. Available: http://arxiv.org/abs/2302.02318

[13] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” no. arXiv:1511.06434, Jan. 2016, arXiv:1511.06434. [Online]. Available: http://arxiv.org/abs/1511.06434

[14] J. Zhou, C. Wei, H. Wang, W. Shen, C. Xie, A. Yuille, and T. Kong, “ibot: Image bert pre-training with online tokenizer,” no. arXiv:2111.07832, Jan. 2022, arXiv:2111.07832. [Online]. Available: http://arxiv.org/abs/2111.07832

[15] Z. Huang, X. Jin, C. Lu, Q. Hou, M.-M. Cheng, D. Fu, X. Shen, and J. Feng, “Contrastive masked autoencoders are stronger vision learners,” no. arXiv:2207.13532, Jan. 2024, arXiv:2207.13532. [Online]. Available: http://arxiv.org/abs/2207.13532

[16] J. Zbontar, L. Jing, I. Misra, Y. LeCun, and S. Deny, “Barlow twins: Self-supervised learning via redundancy reduction,” no. arXiv:2103.03230, June 2021, arXiv:2103.03230. [Online]. Available: http://arxiv.org/abs/2103.03230

[17] C. Tao, X. Zhu, W. Su, G. Huang, B. Li, J. Zhou, Y. Qiao, X. Wang, and J. Dai, “Siamese image modeling for self-supervised vision representation learning,” no. arXiv:2206.01204, Nov. 2022, arXiv:2206.01204. [Online]. Available: http://arxiv.org/abs/2206.01204

[18] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, “Imagenet large scale visual recognition challenge,” no. arXiv:1409.0575, Jan. 2015, arXiv:1409.0575. [Online]. Available: http://arxiv.org/abs/1409.0575

[19] T. T. Team. (2019, jan) Flowers. [Online]. Available: http://download.tensorflow.org/example_images/flower_photos.tgz

[20] A. Krizhevsky, “Learning multiple layers of features from tiny images,” 2009.