SimCLR-UAL: Accelerated Contrastive Learning with Unsupervised Techniques and Active Learning Principles

Abstract

With the emergence of big data, manual labeling of images for classification tasks is increasingly cost-prohibitive. Supervised learning results in the most accurate classification accuracies but at the expense of human labeling, which consume tremendous amounts of time. Contrastive learning methods have grown due to their abilities to label data without human intervention prior to running image classification models. However, a core limitation of contrastive learning is the computational power required to generate labels quickly and accurately for large image datasets. The focus of this work was improving the implementation of a contrastive learning framework (SimCLR). The proposed method combines principles from other subfields of machine learning to improve SimCLR performance, namely unsupervised learning and active learning. This has resulted in a SimCLR based unsupervised active learning (SimCLR-UAL) implementation that dramatically reduces model classifier training times compared to SimCLR and greatly improves training accuracy. Different subsets of the CIFAR-10 dataset was used to validate this method. To date, this work is also one of the first to utilize vanilla deep convolutional autoencoders and hybrid models along with SimCLR in an attempt to achieve comparable testing accuracies at lower batch sizes using only a single GPU. This contrasts with other studies that mostly use ResNet models at higher batch sizes and via the use of multiple GPUs. As a result, this work suggests that the computational costs of SimCLR can still be reduced while maintaining relatively high image classification accuracy values.

Keywords: Unsupervised learning; Self-supervised learning; Contrastive learning; Active learning; Image classification.

1. Introduction

In Computer Vision (CV), several visual objectives exist such as semantic segmentation, reconstruction of missing pixels, image classification, and many others [1]. The purpose of image classification is to create vision models that can autonomously process and classify images into their pre-defined classes or labels. Three types of image classification consist of supervised learning (SL), unsupervised learning (UL), and more recently, self-supervised learning (SSL). In SL, vision models are trained on images where the classes are already predefined. After training, images unseen by the model are screened to evaluate how accurately they are assigned to their predefined classes. Curating data for SL models is the most time-consuming as all images are human-labeled. In UL, images are unlabeled, and models rely on uncovering hidden patterns within the data to classify images into their perceived classes without human annotations. Although the task of labeling images is mostly eliminated, different UL algorithms classify images differently based on clustering, anomaly detection, or community discovery objectives [2]. Other limitations of UL include lack of interpretability and model bias that stems from uncertainty in how the method algorithms handle out-of-distribution samples [3].

SSL is a subset of UL where the models learn meaningful patterns from unlabeled images by creating different versions of the existing images. SSL aims to reduce model bias and increase data interpretability with pretext tasks or the **pre-training phase** [2], [3]. A pretext task is one designed to learn more useful data characteristics prior to training an image classification model. An example of a pre-training phase is a model that generates in-distribution training samples [3]. Such samples can potentially result in more balanced amounts and representations of images for

each class than from UL techniques. Several SSL methods are categorized into contrastive, generative, and generative-contrastive learning according to [3]. Other methods include non-contrastive based models [4], [5] and vision transformers [6], [7], which do not always focus on image classification as its downstream task. For instance, DetCo is an SSL framework that is more effective for object detection while still maintaining classification accuracy [8]. t-SimCNE and SinSim are SSL frameworks that promote a better visualization of high-dimensional data and improve class separation of images [9], [10]. A popular SSL subfield for image classification, which is the focus of this work, is contrastive learning [11].

1.1. Background of Contrastive Learning

Contrastive learning (CL) is a modeling objective that maximizes agreement of images that are most similar and minimizes agreement of images that are most dissimilar and is a subset of self-supervised learning [12]. Models discriminate between augmented versions of images using different set of augmentations, loss functions, and architectures. The process of contrastive learning is as follows: Firstly, various projections of the same image are produced as determined by the user, which leads to an increased training set size. These generations include but are not limited to random cropping, rotation, flipping, Gaussian blurring, color jitters, to name a few [11]. Secondly, the original image serves as the anchor while the different views are paired based on similarity. A chosen deep convolutional neural network (DCNN) encoder (backbone) such as ResNet-18 extract some features of the most similar images and extract some features of the most dissimilar looking images with a loss function. These features are used to closely group similar images and remove dissimilar images as the similarities of the different views from the same image are learned. Thirdly, the images are projected into a shared embedding space (or the base header) where the pre-trained weights are transferred over to a different model for classification training [13]. The trained model is then used to evaluate the accuracy of classification with test images that the model has not seen yet.

State-of-the-art CL methods are closer to achieving performance accuracies to those of supervised learning methods [14]. However, a severe limitation of these methods is the computational cost accumulated throughout the process. Data generation or augmentation increases the amount of images fed into pre-training models and subsequently, the classification models. Choosing the best augmentation strategies should improve downstream classification tasks [14]. This also means that the quality of the samples that are augmented impact pre-trained and classification model performance [15]. Many contrastive learning methods such as SimCLR, BYOL, and momentum encoders rely on large batch sizes, which can extend the pre-training time for SSL models, lead to higher energy costs, and limit the usage of SSL methods to institutions with sufficient resources to train large models [16].

1.2. Research Problem

Only a few works have been proposed to reduce the computational costs associated with implementing CL. [16] proposes multiple strategies to reduce training times of CL methods that are model-agnostic while maintaining classification accuracies comparable to those CL methods without any speed-up strategies. The first strategy utilizes learning rate schedules that gradually increase to higher values and decays to small values while accommodating to large batch sizes. The second strategy increases flexibility in what resolution images are augmented during the creation of different image variations: For higher learning rates, lower resolution images are used and vice versa [17]. Implementing this ensures the pre-trained model learns from quality samples

based on learning rates. The third strategy quantifies the usefulness of the training samples resulting from the second strategy and selects the most meaningful augmentation pair that benefits pre-training performance. In short, constructing algorithms that adaptively adjusts the learning rate, resolution, and augmentation based on the image distributions effectively speed up the pre-training at least 2.3 times with five SSL frameworks. The authors of these works foresee that more benchmarks will be created with other backbones besides ResNet-50, which are the most popular backbones for CL techniques due to their abilities to capture detailed image features even with deeper convolutional layers. There are yet to be studies that use models shallower than ResNet-18.

[18] showed the minimum number of images necessary for pre-trained models to achieve higher generalization abilities while trying to address how probably approximately correct (PAC-learning) bounds have stronger predictive power than experimental results. This was done via evaluating the relationship between the sample complexity generated from the outputs of a pre-trained model and their effects on classification model performance. It was observed that computational costs scale linearly with the number of pre-trained samples, but it is possible to bound the complexity of models prior to pre-training. A consensus from this study and experiments conducted in [16], is that the careful selection of samples along with other hyperparameters in a CL model indeed affect the performance of pre-training, classification training, and finally the classification accuracy. Another open question from this study is not only choosing sufficiently complex samples but also deciding the appropriate batch size.

1.3. Research Objectives

Optimizing the pre-training speed and quality to improve classification accuracy with less powerful computational resources remains an open question. A broader development of efficient CL methods is needed for different users that may not have access to high-performance computing resources. This can potentially lead to saving hours and even days of compute time and energy costs. Thus, this study's objectives are as follows:

- Improve SimCLR [11] to reduce the number of model parameters with simpler DCNNs while reaching comparable classification accuracies with SL models.
- Use UL and AL techniques [19] to improve sample selection, which leads to faster pre-training speeds and loss with a minor trade-off in testing accuracy.

The rest of the paper is organized as follows. Section 2 summarizes gaps in multiple existing improvements to SimCLR and reviews active learning approaches relevant to improving SimCLR. Section 3 details the proposed methodology to improve SimCLR with unsupervised learning, active learning, and other data augmentation techniques. Section 4 details the implementation of the proposed algorithm and the pre-training parameters used in this study. Section 5 provides different analyses of the results and discusses some limitations of the proposed algorithm. Section 6 concludes the work and state its implications for the field of CL.

2. Literature Review

This section continues the review of contrastive learning works and active learning approaches from Section 1. In subsection 2.1, an overview of the SimCLR framework is explained. The components of this model that were leveraged to improve classification accuracies and training times is also summarized. In subsection 2.2, active learning approaches that improved components of SimCLR or can improve SimCLR's functionalities are summarized.

2.1. SimCLR Improvements

SimCLR is an early CL framework that work as follows: First, a pre-set stochastic data augmentation is applied to transform an image into two versions. Second, a similar base encoder (backbone) structure is used to extract features for each version and represent them as vectors. The vectors are then computed for similarity or dissimilarity with the original image using a contrastive loss function: The augmented image batches with the most similarity as the original image batches is brought closer together in the latent space (positive pair) while the least similar augmented image batches compared to the original batches are pulled away (negative pair) [11], [20]. A projection head (in the form of a fully connected dense layer) is optionally added to the pre-trained model for the classification task. In SimCLR, batch size is correlated with classification accuracy. Large batch sizes lead to unstable learning with stochastic gradient descent optimizers, so [11] used the layer-wise adaptive scaling (LARS) optimizer [21] and incorporated the normalized temperature scaled cross-entropy (NT-Xent) loss for optimized CL.

Despite SimCLR's ease of use, some limitations were yet to be addressed. Models were trained on TPUs and multiple GPUs due to their expensive computations from large batch sizes. Several pairs of data augmentations were also tried. The best pair of augmentations that led to the highest classification accuracy with the ImageNet dataset was random cropping and random color distortion [11]. Other data augmentation pairs were also studied, which involve a host of image noising, filtering, blurring, and color distortion techniques; however, there is yet to be a systematic grid or random search optimization to automatically determine the best pairing. Such an idea could increase pre-training speed but lower loss.

Some shortcomings of SimCLR that are addressed to outperform the original include the integration of SSL with UL methods [22], adding noise to images [23], using different loss functions and leveraging more positive pairs in the loss function calculations [24], [25], [26]. One of the earlier additions to SimCLR is the G-SimCLR algorithm, which utilized a k-means clustering and the latent space of a denoising autoencoder to facilitate the construction of pseudo-labels [22]. The result of these additions is an increased diversity of labels within the same batch, which led to performance enhancements because no two semantically similar images got treated differently. Additionally, unlike the original SimCLR where the images got sampled randomly, the pseudo labels obtained from the clusters minimized the risk of using similar images in the same batch, leading to more strategic learning of pairings with minimal bias [22]. While the study is optimistic that these additions reduce large batch sizes and false positives, future work entails applying this method to larger CV datasets such as ImageNet and CIFAR-100 and continually improving the efficiency of clustering algorithms contained in the autoencoder generated latent space as clustering runtimes can delay the overall pre-training process. A study from [23] observed that adding both symmetric and asymmetric noise for some of the pre-training samples could possibly improve model robustness because they also improve the diversity and quality of images across different batch sizes. The models improve their learning when exposed to different image classes that are modified with varying degrees of uniform noise. A limitation of this study is that still, the computational resources used were expensive with 12 GPUs used. [24] improved SimCLR performance via the supervised contrastive (SupCon) or a modify triplet loss than the previously used NT-Xent loss. A significant benefit of this function was the stability of different hyperparameter settings (learning rate, optimizer type) and data augmentation pairings. Another reason for creating the SupCon loss was due to the fact that the cross-entropy loss is not robust to noisy labels and generalizes poorly to the training data. Furthermore, this loss utilized multiple

positive pairs per original image of the same class instead of just one positive pair in the original SimCLR, which encouraged diverse learning of samples as shown by the gradient of the loss function and less sensitivity to diverse hyperparameters. This study still relied on more GPUs because of increased number of positive pairs. [25] also made use of several positive pairs called the Efficient Combinatorial Positive Pairing (ECPP) to boost SimCLR that matched the performance of SL methods from the ImageNet-100 dataset. ECPP boosted SimCLR is able to generate and process the most amount of differing image versions due to its combinatorial design. This method ultimately showed that different number of positive pairs for each image class leads to improved learning efficiency of SimCLR.

SimCLR improvements that do not always outperform the original SimCLR in terms of accuracy but for pre-training speeds include using smaller backbone models such as EfficientNetV2 paired with progressive learning [17]. Here, some pre-training examples are also scheduled from easy to hard instead of being picked randomly, which combats overfitting and bias towards a particular image label. This method also implements dropout layers during the pre-training step to further reduce computational costs. A study from [27] attempts to overcome large batch sizes by approaching contrastive learning via optimization techniques using their proposed memory-efficient SogCLR method. SogCLR with a batch size of 256 performed almost similarly as SimCLR with a batch size of 8192 with the ImageNet-1K dataset. The authors posited that implementing a global objective for contrastive learning and a stochastic algorithm for accelerated convergence allow minimized errors with smaller batch sizes. SogCLR can still be evaluated more with lower computational resources, other datasets, and with simpler CNN models other than the varying levels of ResNet models.

2.2. Recent advancements in Active Learning

Active learning (AL) is another subfield of machine learning where a model is designed to maximize the performance of a downstream task with the least amount of the most informative unlabeled images. This is to mitigate the labor-intensive cost of collecting millions of labeled data for SL models. Traditional AL approaches that do not require training have standard baselines for comparison, namely random selection of the best samples (random sampling), selection of samples with the smallest maximum activations (least confidence), selection of samples with the smallest separation between the closest class predictions (margin sampling), and selection of samples with minimal entropy loss if there are more than two image classes (uncertainty or entropy sampling) [28], [29]. Studies have combined two baseline approaches to create novel AL frameworks. For instance, [30] have created an adaptive weighted uncertainty sampling model, which balances random and uncertainty methods. This allowed the model to learn more informative features from additive manufacturing optical images to reduce the number of labeled images needed to increase classification accuracy tasks. Only 20-70% of the total training data were required for model gains.

Deep learning models were combined into the AL framework to extract more meaningful features from images for sample selection, creating the field of deep active learning (DAL) [31]. One study implemented deep learning models into AL with fully convolutional networks (FCNs), which were used as feature extractors iteratively trained to learn the most useful samples from an entire medical image training set [19]. Every iteration informs the FCNs the next best batch of images to label and use for classification. Several challenges existed for a solid DAL framework, such as quantifying the similarity between images, improving selections with augmented data, and speeding up model training [19]. As a result, DAL methods expanded to incorporate the latest network architectures such as ResNet models, latest augmentation techniques, and have worked

towards reducing bias in selecting samples [32]. Moreover, the studies have shown that incorporating data augmentation in AL training improves labeling efficiency, but measures need to be taken to accelerate training convergence. [33] also describe many AL frameworks that use less than 60% of provided training data. For medical image segmentation and classification, notable methods include the cost-effective AL benchmark, modified uncertainty sampling, and Bayesian CNNs [34]. Another challenge with AL is creating models that account for imbalanced datasets, where the number of samples per class vary from each other [35]. Studies have addressed class imbalance via estimating the probability of samples belonging to the majority or minority and augmented data as needed with different subsets of the CIFAR-10 dataset. Other DAL challenges consist of unstable performances due to the vanishing gradient problem of DCNNs [36], and the emerging practical usage of AL for data sharing amongst multiple machines [30], [37].

Using AL or DAL to boost CL continues to be an open problem. The integration of the two subfields are only beneficial for training datasets that require a lot of images to be labeled [29]. According to the CIFAR-10, CIFAR-100, and Tiny ImageNet datasets used in [38], SSL by itself improved image classification accuracy than AL methods for smaller datasets. [39] introduce an SSL combined AL framework that minimizes compute while boosting classification accuracy via selecting subsets from remote sensing datasets and combined sampling methods. These methods do not guarantee improved performance if the pre-training dataset is small. The performance gains of combining AL and SSL in one framework holds for larger datasets. Many studies are gearing towards using AL to select the most optimal data to label for SSL to realize their applicability for medical and manufacturing applications [40]. [41] integrated DAL with SSL in several ways only to find that the impacts of classification accuracy are marginal. However, there is still the opportunity to design better AL algorithms that can be integrated into state-of-the-art SSL models because existing AL methods still require too many labels. [42] designed ConAL to compile the most informative samples with the same class labels from an unlabeled pool and filter out mismatched augmented samples. A more effective query strategy was then built to annotate images at lower costs with many realistic datasets [43]. Such methods are promoting different algorithms that aim to select the most informative samples but can be improved with more thorough analyses of inter-cluster diversity. An example of another clustering-based AL can successfully manipulate the local and global topologies of high-dimensional feature spaces to assess and reduce complexity of AL algorithms while creating interpretable visualizations [44]. Combined CL and clustering methods [45] can also improve selection of the most informative samples and filter out uninformative ones, such as a small set of invasive larvae data that contained imbalanced labels with only 100 samples [46]. From these studies, it is clear that diverse and representative samples are needed to successfully implement AL with SSL.

2.3. Research Gaps and Opportunities

Upon a thorough review of contrastive learning works in Section 1, existing SimCLR improvements, and active learning advancements, multiple gaps and opportunities are identified. The majority of the SSL frameworks use ResNet DCNNs with skip connections that memorize information from previous layers. While skip connections mitigate the vanishing gradient problem, these models are computationally expensive. Random data augmentation does not always guarantee and can even hinder pre-trained models, which negatively affect image classification accuracy. Thus, a careful selection of augmentations must be decided to improve pre-training, training, and classification performance. SimCLR relies on large batch sizes, which may limit its use for hardware with lower memory sizes. It is yet to be explored **how many samples are needed**

for pre-training to realize higher training and testing generalizations. Combined AL and CL frameworks were trained on many GPUs and TPUs but there is the opportunity to produce a cost-effective algorithm that contains a smaller number of parameters than ResNet backbone models.

3. Methodology

In this work, the **CIFAR-10** dataset was used to validate the proposed SimCLR-UAL methodology. Section 3.1 details the benchmark methods used. Section 3.2 overviews this work's technical contributions and explains the model components of the proposed method.

3.1. Existing benchmark methods

Two backbone architectures for pre-training and training phases were used in this study: a vanilla three-layer DCNN (Figure 1) and a hybrid DCNN-SVM model inspired from [47] (Figure 2). For each SimCLR framework, the same data augmentation setup was used, balancing increased backbone complexity with the diversity of augmented samples [11]. For the Support Vector Machine (SVM) component of Figure 2, the C, gamma, and kernel parameters were applied in a 5-fold cross validation with grid search to penalize misclassification of training data and to capture more meaningful patterns for classification. The range of C values tried was [0.1, 1000] in 10 orders of magnitude, the gamma values tried was [0.01, 1000], and the kernels tried was the Radial Basis Function (RBF) and the linear function.

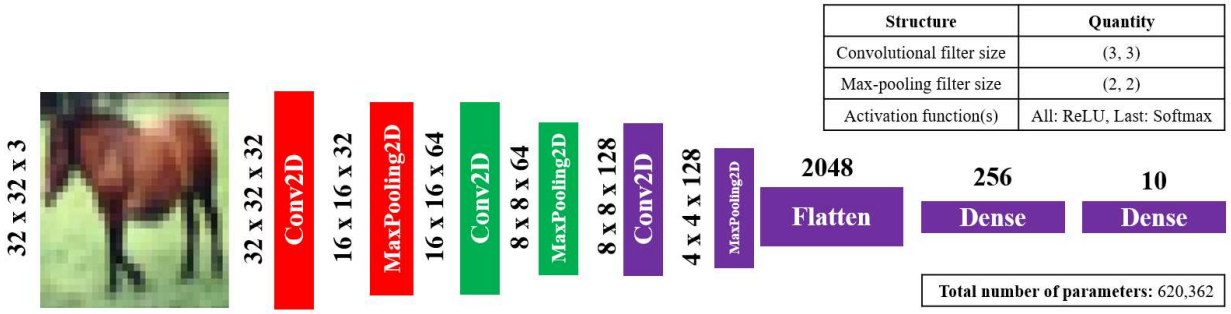

Figure 1. Three-layer DCNN backbone

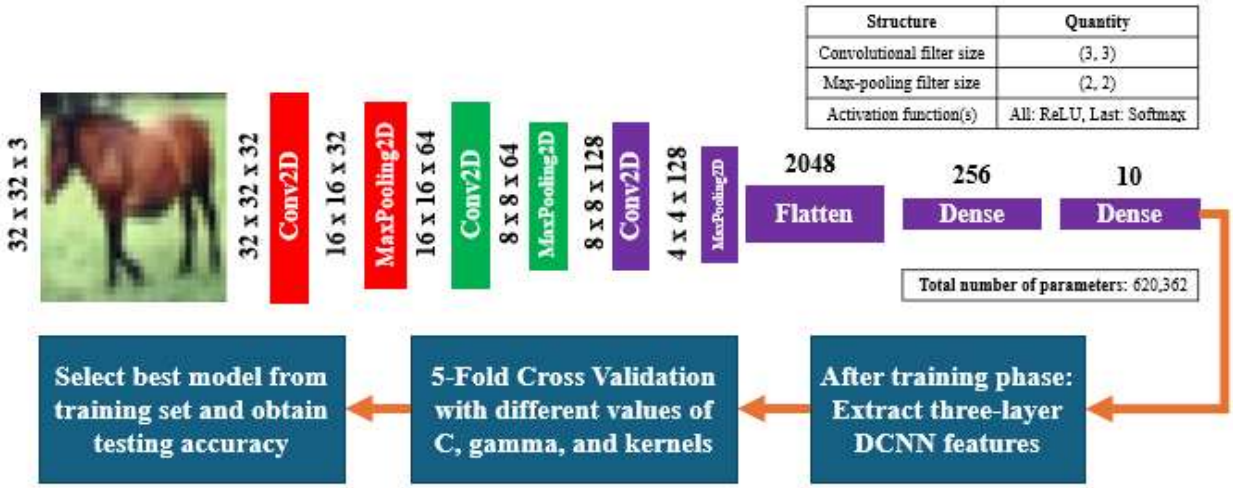

Figure 2. Three-layer DCNN-SVM backbone

Each image was center cropped to half of the image, brightness of each image was randomly set from no change to full brightness, and the contrast of each image was randomly set from no change to full darkness as shown in Figure 3.

Figure 3. Chosen data augmentation setup for both backbone models

For the image classification task, both models were trained at 100 epochs using a batch size of 256 with the Adaptive Moment Estimation (Adam) optimizer of 0.001 learning rate and lastly with the categorical cross entropy as the loss function to convert the softmax probabilities from the final fully connected dense layer into one of the ten data labels. 256 was selected as the batch size [27] due to the limitations of the hardware used in this work.

3.2. Proposed SimCLR-UAL Framework

This subsection proposes the SimCLR-UAL methodology for accelerating SimCLR training speeds and training accuracies by integrating unsupervised learning and active learning principles. This new methodology focuses on expediting training times while selecting the best representative samples for improving training accuracy.

The novelty of this methodology is to maximize the selection of the most informative images while minimizing the least important ones when labeling the training dataset. While SimCLR already decreases training times and accelerates convergence with the best positive pairs, SimCLR-UAL uses a selection criterion with unsupervised learning strategies to select only a few positive pairs that is the most important for model training. A more concise selection of the most relevant positive pairs leads to further improvements in training speed. The steps of the SimCLR-UAL framework are as follows:

1. Data augmentation is applied the similar way to the SimCLR method.
2. Pre-training was completed in the same way as the SimCLR method. The details for the pre-training parameters are provided in Table 1.
3. After pre-training, the “Unsupervised Active Learning” component was applied:
    a. The fully connected dense layer of each backbone had 10 features for all training samples in the CIFAR-10 data. The t-Stochastic Neighbor Embedding (t-SNE) map was used as a selection tool to extract two features for each training sample while simultaneously visualizing each sample as a coordinate in a two-dimensional space.
    b. The **unsupervised** k-means clustering algorithm was applied to the t-SNE map to cluster the samples in the embedding space into 10 classes. The centroids of each cluster for each class was also calculated.

c. The margin sampling principle belonging to **traditional active learning** techniques was applied via the following:
    i. For every cluster (class), the Euclidean distances between the centroid and each individual coordinate belonging to the same class was computed.
    ii. The coordinates were ordered from the smallest to the largest distance from the centroid.
    iii. The closest 256 coordinates were chosen to represent the most important 256 samples that were most informative for each class label. 256 was chosen as it is equal to the pre-training batch size.

4. These samples were chosen as the fine-tuned training set for the downstream image classification task unlike SimCLR where the training sets are random, and the samples may not always represent multiple classes.
5. Lastly, the classification accuracy is evaluated with the CIFAR-10 testing data. Figure 4 visually encapsulates the proposed SimCLR-UAL framework.

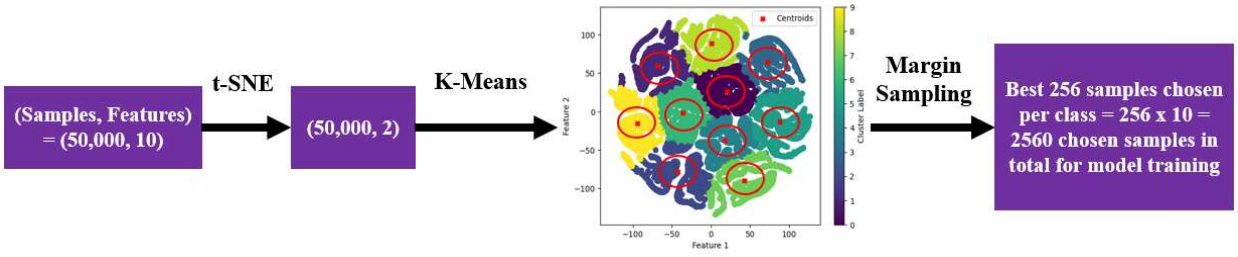

Figure 4. Overview of the SimCLR-UAL methodology

4. Implementation

All experiments were run on a Windows computer with a 12th Generation Intel (R) Core (TM) i9-12900 CPU @ 2.40 GHz and a single NVIDIA GeForce RTX 3070 GPU. Different subsets of the CIFAR-10 data were used to demonstrate that the methods are feasible even for limited and imbalanced training data splits. While training parameters were kept similar for both models, the pre-training parameters were manually tuned to account for model complexity. The LARS optimizer was not used due to the smaller batch sizes as observed in [25]. Table 1 describes the parameters used for the pre-training phase of each model. All implementations were conducted in Python. The GitHub is available at: https://github.com/ezy-8/SimCLR-UAL

Table 1. Pre-training parameters

Pre-training epochs	10
Batch size	256
Optimizer	Stochastic Gradient Descent
Optimizer Learning Rate	0.01
Loss Function	NT-Xent
Temperature value for loss function	0.2
Fully dense layer projection head	Yes; 256 Dimensions

5. Results and Discussion

Table 2. CIFAR-10 performance across methods using the three-layer DCNN backbone

% of data used	Three-layer DCNN backbone	Performance metrics			
		Pre-training time (seconds)	Training time (seconds)	Training accuracy (%)	Testing accuracy (%)
30	Supervised	N/A	37.14	100.00	65.23
	SimCLR	46.72	18.97	52.52	47.17
	SimCLR-UAL	**45.92**	**5.29**	**92.33**	**50.27**
60	Supervised	N/A	70.71	100.00	70.38
	SimCLR	**92.26**	35.61	58.32	49.15
	SimCLR-UAL	92.44	**5.3**	**98.97**	**52.75**
100	Supervised	N/A	115.83	100.00	73.80
	SimCLR	**152.42**	58.23	71.91	**57.93**
	SimCLR-UAL	152.85	**5.64**	**99.32**	52.52

In tables 2 and 3, the **bold** indicates the better result when comparing the existing SimCLR method versus the proposed SimCLR-UAL method. Because SimCLR-UAL only selects the best 256 samples to train the model, it drastically reduces the training time across all the subsets of the CIFAR-10 training data used and leads to superior training accuracies. There are improvements in the testing accuracy for 30% and 60% data used but a decline in testing accuracy for all data used.

Table 3. CIFAR-10 performance across methods using the three-layer DCNN-SVM backbone

% of data used	Three-layer CNN-SVM model	Best 5-Fold Cross Validation hyperparameters (C, gamma, kernel)	Performance metrics			
			Pre-training time (seconds)	Training time (seconds)	Training accuracy (%)	Testing accuracy (%)
30	Supervised	0.1, 0.1, RBF	N/A	37.50	92.99	65.90
	SimCLR	10, 10, Linear	**46.14**	19.19	**73.00**	**55.07**
	SimCLR-UAL	10, 10, Linear	46.64	**5.40**	58.05	49.47
60	Supervised	0.1, 1, RBF	N/A	70.88	93.92	69.78
	SimCLR	1000, 10, Linear	93.56	36.04	**71.79**	**53.82**
	SimCLR-UAL	1000, 10, Linear	**93.16**	**5.46**	70.11	50.95
100	Supervised	0.1, 0.1, RBF	N/A	116.29	95.03	74.43
	SimCLR	10, 1, RBF	**151.38**	59.71	**74.76**	**54.88**
	SimCLR-UAL	1000, 10, Linear	152.94	**6.20**	70.41	52.74

For this backbone, there were no improvements using SimCLR-UAL over SimCLR. Instead, there was a decline in training and testing accuracy values. There were still the same significant improvements in training times across all subsets of the CIFAR-10 training data used.

Figure 5 shows the graphical representation of the results for the full CIFAR-10 dataset used that highlight the trade-offs between training time and testing accuracy between the two different backbones for the Supervised, SimCLR, and SimCLR-UAL methodologies.

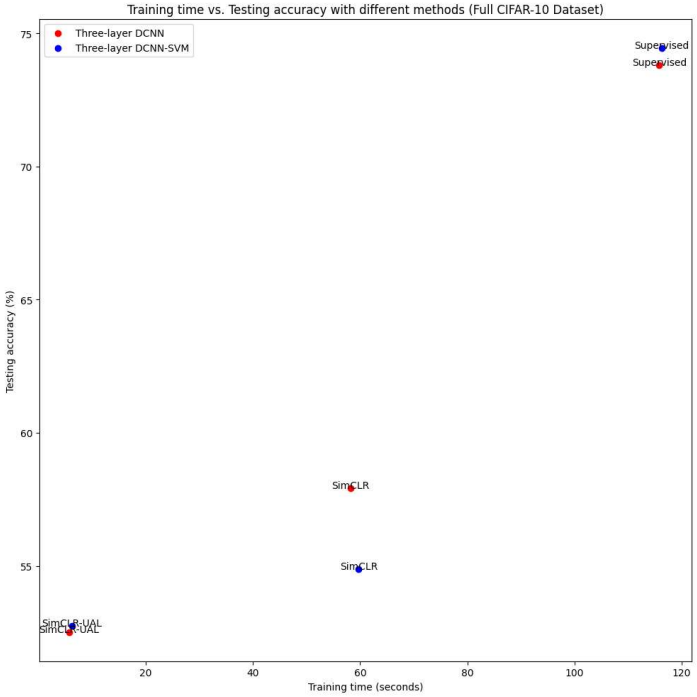

Figure 5. Training time vs. testing accuracy across backbones from all samples in CIFAR-10 data

The three-layer DCNN served as a more feasible backbone for the proposed SimCLR-UAL framework. Both backbones had comparable SimCLR and SimCLR-UAL pre-training and training times due to the same pre-training parameters. **These results prove that the best pre-training setups are capable of manually labeling big datasets that accelerate model training convergence and in the best cases, improve image classification accuracies for testing data while using DCNNs that are not as resource-intensive as ResNet-18 backbones.** Figure 6 shows the t-SNE maps of all the data used created by both backbone models.

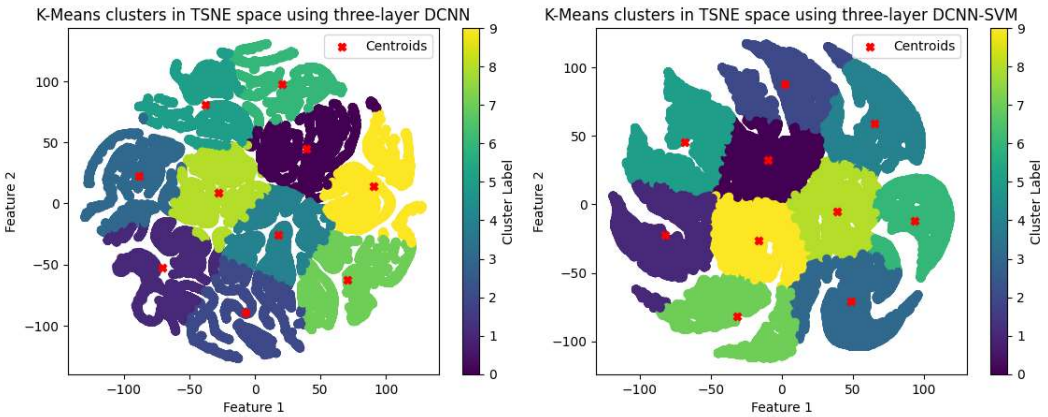

Figure 6. K-means generated clusters in the TSNE space from all samples in CIFAR-10 data

While the proposed method itself is a combination of SimCLR and UAL, the UAL component of the method was also analyzed separately as shown in Table 4. Instead of selecting the best 256 samples derived from the closest Euclidean distance of the centers, the best 1000 to 4000 samples with the closest distance to the centers of each class are now selected. For the three-

layer DCNN, the best testing accuracy came from selecting 3000 samples per class that are best representative of the data as defined by the model. For the three-layer DCNN-SVM, the best testing accuracy came from selecting 4000 samples per class.

Table 4. Unsupervised Active Learning ablation study for varying sample sizes

Model	Performance metrics	Number of selected samples per class				
		1000	2000	3000	4000	Supervised
Three-layer DCNN	Training time (seconds)	**24.47**	47.06	68.02	90.06	115.83
	Training accuracy (%)	100.00	100.00	**100.00**	100.00	100.00
	Testing accuracy (%)	74.07	74.67	**76.26**	75.09	73.80
Three-layer CNN-SVM	Best (C, gamma, kernels)	100, 10, Linear	10, 10, Linear	1000, 10, Linear	10, 10, Linear	0.1, 0.1, RBF
	Training time (seconds)	**24.60**	47.58	71.26	92.92	116.29
	Training accuracy (%)	99.90	99.88	**99.92**	99.89	95.03
	Testing accuracy (%)	75.34	74.75	75.36	**75.96**	74.43

6. Conclusion

This work aimed to optimize pre-training speed, training accuracy, and testing accuracies with less powerful computational resources to highlight that simpler DCNN backbones can be used with SimCLR for users that may not have access to multiple GPUs. The results indicate that the proposed SimCLR-UAL method can lead to hours and even days' worth of compute power and energy saved while reaching comparable classification accuracies with the original SimCLR framework. The UAL implementation of this work can be integrated with SimCLR or used separately to reach classification accuracies close to supervised learning techniques for the CIFAR-10 dataset at reduced training times and with better sample selections.

There are still many limitations of the proposed SimCLR-UAL that could be addressed for future work. The SimCLR based data augmentation is still stochastic in nature; however, optimization strategies can be integrated to automate the search for augmentation pairs that minimize pre-training loss, which can lead to better generalization of the training data. For the CIFAR-10 dataset, interpolating the image to a larger size can be an augmentation feature, as it is difficult to distinguish between classes due to a small image size. Progress can still be made to ensure no two positive pairs are in the same batch, as this limits the models' abilities to learn from diverse samples and thus, classification accuracies. The Euclidean distance metric was used to calculate the similarity of embeddings within the t-SNE space, but different distance metrics can be considered, which may group together more meaningful images of the same class. Lastly, this work used a fixed number of pre-training parameters, but this can vary based on the complexity of the model backbone. This way, an optimal number of pre-training epochs and other parameters can be determined to maximize the benefits of pre-training.

The implications of using both the benchmark and proposed methods towards real-world datasets besides CIFAR-10 are yet to be explored. While this work utilized pre-training to obtain labels that reduced training times, it is not certain that the automatically assigned labels will almost always match with the ground truth labels. This is because real-world datasets have stronger data imbalances, which may need to be offset with weighted categorical cross entropy loss along with other hyperparameter tuning methods for vanilla and hybrid DCNNs.

References

[1] X. Zhao *et al.*, "Contrastive Learning for Label-Efficient Semantic Segmentation," Aug. 18, 2021, *arXiv*: arXiv:2012.06985. doi: 10.48550/arXiv.2012.06985.

[2] J. Gui *et al.*, "A Survey on Self-supervised Learning: Algorithms, Applications, and Future Trends," Jul. 14, 2024, *arXiv*: arXiv:2301.05712. doi: 10.48550/arXiv.2301.05712.

[3] X. Liu *et al.*, "Self-supervised Learning: Generative or Contrastive," *IEEE Trans. Knowl. Data Eng.*, pp. 1–1, 2021, doi: 10.1109/TKDE.2021.3090866.

[4] J.-B. Grill *et al.*, "Bootstrap your own latent: A new approach to self-supervised Learning," Sep. 10, 2020, *arXiv*: arXiv:2006.07733. doi: 10.48550/arXiv.2006.07733.

[5] X. Chen and K. He, "Exploring Simple Siamese Representation Learning," Nov. 20, 2020, *arXiv*: arXiv:2011.10566. doi: 10.48550/arXiv.2011.10566.

[6] M. Caron *et al.*, "Emerging Properties in Self-Supervised Vision Transformers," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada: IEEE, Oct. 2021, pp. 9630–9640. doi: 10.1109/ICCV48922.2021.00951.

[7] M. Oquab *et al.*, "DINOv2: Learning Robust Visual Features without Supervision," Feb. 02, 2024, *arXiv*: arXiv:2304.07193. doi: 10.48550/arXiv.2304.07193.

[8] E. Xie *et al.*, "DetCo: Unsupervised Contrastive Learning for Object Detection," Jul. 23, 2021, *arXiv*: arXiv:2102.04803. doi: 10.48550/arXiv.2102.04803.

[9] J. N. Böhm, P. Berens, and D. Kobak, "Unsupervised visualization of image datasets using contrastive learning," Feb. 28, 2023, *arXiv*: arXiv:2210.09879. doi: 10.48550/arXiv.2210.09879.

[10] M. H. Sepanj and P. Fiegth, "SinSim: Sinkhorn-Regularized SimCLR," Feb. 13, 2025, *arXiv*: arXiv:2502.10478. doi: 10.48550/arXiv.2502.10478.

[11] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A Simple Framework for Contrastive Learning of Visual Representations".

[12] A. W. Peng, J. He, and F. Zhu, "Self-supervised visual representation learning on food images," *Electron. Imaging*, vol. 35, no. 7, pp. 269-1-269–6, Jan. 2023, doi: 10.2352/EI.2023.35.7.IMAGE-269.

[13] S. Albelwi, "Survey on Self-Supervised Learning: Auxiliary Pretext Tasks and Contrastive Learning Methods in Imaging," *Entropy*, vol. 24, no. 4, p. 551, Apr. 2022, doi: 10.3390/e24040551.

[14] A. Jaiswal, A. R. Babu, M. Z. Zadeh, D. Banerjee, and F. Makedon, "A Survey on Contrastive Self-Supervised Learning," *Technologies*, vol. 9, no. 1, p. 2, Dec. 2020, doi: 10.3390/technologies9010002.

[15] J. Mitrovic, B. McWilliams, and M. Rey, "Less can be more in contrastive learning".

[16] M. T. Kocyigit, T. M. Hospedales, and H. Bilen, "Accelerating Self-Supervised Learning via Efficient Training Strategies," in *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Waikoloa, HI, USA: IEEE, Jan. 2023, pp. 5643–5653. doi: 10.1109/WACV56688.2023.00561.

[17] M. Tan and Q. V. Le, "EfficientNetV2: Smaller Models and Faster Training," Jun. 23, 2021, *arXiv*: arXiv:2104.00298. doi: 10.48550/arXiv.2104.00298.

[18] N. Alon, D. Avdiukhin, D. Elboim, O. Fischer, and G. Yaroslavtsev, "Optimal Sample Complexity of Contrastive Learning," Dec. 01, 2023, *arXiv*: arXiv:2312.00379. doi: 10.48550/arXiv.2312.00379.

[19] L. Yang, Y. Zhang, J. Chen, S. Zhang, and D. Z. Chen, "Suggestive Annotation: A Deep Active Learning Framework for Biomedical Image Segmentation," Jun. 15, 2017, *arXiv*: arXiv:1706.04737. doi: 10.48550/arXiv.1706.04737.

[20] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," Dec. 10, 2015, *arXiv*: arXiv:1512.03385. Accessed: Aug. 14, 2024. [Online]. Available: http://arxiv.org/abs/1512.03385

[21] Y. You, I. Gitman, and B. Ginsburg, "Large Batch Training of Convolutional Networks," Sep. 13, 2017, *arXiv*: arXiv:1708.03888. doi: 10.48550/arXiv.1708.03888.

[22] S. Chakraborty, A. R. Gosthipaty, and S. Paul, "G-SimCLR : Self-Supervised Contrastive Learning with Guided Projection via Pseudo Labelling," Sep. 25, 2020, *arXiv*: arXiv:2009.12007. doi: 10.48550/arXiv.2009.12007.

[23] A. Ghosh and A. Lan, "Contrastive Learning Improves Model Robustness Under Label Noise," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Nashville, TN, USA: IEEE, Jun. 2021, pp. 2697–2702. doi: 10.1109/CVPRW53098.2021.00304.

[24] P. Khosla *et al.*, "Supervised Contrastive Learning," Mar. 10, 2021, *arXiv*: arXiv:2004.11362. doi: 10.48550/arXiv.2004.11362.

[25] J. Kim, D. Hwang, E. Lee, J. Suh, J. Kim, and W. Rhee, "Enhancing Contrastive Learning with Efficient Combinatorial Positive Pairing," Jan. 11, 2024, *arXiv*: arXiv:2401.05730. doi: 10.48550/arXiv.2401.05730.

[26] J. Wu, S. Mo, Z. Feng, S. Atito, J. Kitler, and M. Awais, "Rethinking Positive Pairs in Contrastive Learning," Oct. 23, 2024, *arXiv*: arXiv:2410.18200. doi: 10.48550/arXiv.2410.18200.

[27] Z. Yuan *et al.*, "Provable Stochastic Optimization for Global Contrastive Learning: Small Batch Does Not Harm Performance," Sep. 20, 2022, *arXiv*: arXiv:2202.12387. doi: 10.48550/arXiv.2202.12387.

[28] H. Ranganathan, H. Venkateswara, S. Chakraborty, and S. Panchanathan, "Deep active learning for image classification," in *2017 IEEE International Conference on Image Processing (ICIP)*, Beijing: IEEE, Sep. 2017, pp. 3934–3938. doi: 10.1109/ICIP.2017.8297020.

[29] R. Takezoe *et al.*, "Deep Active Learning for Computer Vision: Past and Future," *APSIPA Trans. Signal Inf. Process.*, vol. 12, no. 1, 2023, doi: 10.1561/116.00000057.

[30] G. J. J. Van Houtum and M. L. Vlasea, "Active learning via adaptive weighted uncertainty sampling applied to additive manufacturing," *Addit. Manuf.*, vol. 48, p. 102411, Dec. 2021, doi: 10.1016/j.addma.2021.102411.

[31] P. Ren *et al.*, "A Survey of Deep Active Learning," *ACM Comput. Surv.*, vol. 54, no. 9, pp. 1–40, Dec. 2022, doi: 10.1145/3472291.

[32] N. Beck, D. Sivasubramanian, A. Dani, G. Ramakrishnan, and R. Iyer, "Effective Evaluation of Deep Active Learning on Image Classification Tasks," Nov. 02, 2021, *arXiv*: arXiv:2106.15324. doi: 10.48550/arXiv.2106.15324.

[33] S. Budd, E. C. Robinson, and B. Kainz, "A survey on active learning and human-in-the-loop deep learning for medical image analysis," *Med. Image Anal.*, vol. 71, p. 102062, Jul. 2021, doi: 10.1016/j.media.2021.102062.

[34] K. Wang, D. Zhang, Y. Li, R. Zhang, and L. Lin, "Cost-Effective Active Learning for Deep Image Classification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 12, pp. 2591–2600, Dec. 2017, doi: 10.1109/TCSVT.2016.2589879.

[35] Q. Jin, M. Yuan, H. Wang, M. Wang, and Z. Song, "Deep active learning models for imbalanced image classification," *Knowl.-Based Syst.*, vol. 257, p. 109817, Dec. 2022, doi: 10.1016/j.knosys.2022.109817.

[36] M. Wu, C. Li, and Z. Yao, "Deep Active Learning for Computer Vision Tasks: Methodologies, Applications, and Challenges," *Appl. Sci.*, vol. 12, no. 16, p. 8103, Aug. 2022, doi: 10.3390/app12168103.

[37] Y. Zhao, Y. Li, C. Liu, and Y. Wang, "ADs: Active Data-sharing for Data Quality Assurance in Advanced Manufacturing Systems," Mar. 31, 2024, *arXiv*: arXiv:2404.00572. doi: 10.48550/arXiv.2404.00572.

[38] J. Z. Bengar, J. Van De Weijer, B. Twardowski, and B. Raducanu, "Reducing Label Effort: Self-Supervised meets Active Learning," in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, Montreal, BC, Canada: IEEE, Oct. 2021, pp. 1631–1639. doi: 10.1109/ICCVW54120.2021.00188.

[39] X. Jiang, L. Scheibenreif, and D. Borth, "Less is More: Active Self-Supervised Learning in Remote Sensing," in *IGARSS 2024 - 2024 IEEE International Geoscience and Remote Sensing Symposium*, Athens, Greece: IEEE, Jul. 2024, pp. 7025–7030. doi: 10.1109/IGARSS53475.2024.10640981.

[40] Z. Lai, C. Wang, L. C. Oliveira, B. N. Dugger, S.-C. Cheung, and C.-N. Chuah, "Joint Semi-supervised and Active Learning for Segmentation of Gigapixel Pathology Images with Cost-Effective Labeling," in *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, Montreal, BC, Canada: IEEE, Oct. 2021, pp. 591–600. doi: 10.1109/ICCVW54120.2021.00072.

[41] Y.-C. Chan, M. Li, and S. Oymak, "On the Marginal Benefit of Active Learning: Does Self-Supervision Eat Its Cake?," Nov. 16, 2020, *arXiv*: arXiv:2011.08121. doi: 10.48550/arXiv.2011.08121.

[42] P. Du, S. Zhao, H. Chen, S. Chai, H. Chen, and C. Li, "Contrastive Coding for Active Learning under Class Distribution Mismatch," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Montreal, QC, Canada: IEEE, Oct. 2021, pp. 8907–8916. doi: 10.1109/ICCV48922.2021.00880.

[43] P. Du, H. Chen, S. Zhao, S. Chai, H. Chen, and C. Li, "Contrastive Active Learning Under Class Distribution Mismatch," *IEEE Trans. Pattern Anal. Mach. Intell.*, pp. 1–13, 2022, doi: 10.1109/TPAMI.2022.3188807.

[44] E. Vaaras, M. Airaksinen, and O. Räsänen, "Analysis of Self-Supervised Learning and Dimensionality Reduction Methods in Clustering-Based Active Learning for Speech Emotion Recognition," Jun. 21, 2022, *arXiv*: arXiv:2206.10188. doi: 10.48550/arXiv.2206.10188.

[45] Z. Yan *et al.*, "Contrastive Open-Set Active Learning-Based Sample Selection for Image Classification," *IEEE Trans. Image Process.*, vol. 33, pp. 5525–5537, 2024, doi: 10.1109/TIP.2024.3451928.

[46] S. Chowdhury, G. Hamerly, and M. McGarrity, "Active Learning Strategy Using Contrastive Learning and K-means for Aquatic Invasive Species Recognition," in *2024 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW)*, Waikoloa, HI, USA: IEEE, Jan. 2024, pp. 848–858. doi: 10.1109/WACVW60836.2024.00097.

[47] F. Bal and F. Kayaalp, "A Novel Deep Learning-Based Hybrid Method for the Determination of Productivity of Agricultural Products: Apple Case Study," *IEEE Access*, vol. 11, pp. 7808–7821, 2023, doi: 10.1109/ACCESS.2023.3238570.