National College of Ireland

SimCLR vs. Supervised ResNet for Low-Label Image Classification with Data Augmentation

MSc Research Project
Data Analytics

Jeson Varghese Varghese
Student ID: x23254998

School of Computing
National College of Ireland

Supervisor: David Hamill

National College of Ireland
Project Submission Sheet
School of Computing

Student Name:	Jeson Varghese Varghese
Student ID:	x23254998
Programme:	MSc Data Analytics
Year:	2025
Module:	MSc Research Project
Supervisor:	David Hamill
Submission Due Date:	01/09/2025
Project Title:	SimCLR vs. Supervised ResNet for Low-Label Image Classification with Data Augmentation
Word Count:	9208
Page Count:	30

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Jeson Varghese
Date:	01/09/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

SimCLR vs. Supervised ResNet for Low-Label Image Classification with Data Augmentation

Jeson Varghese Varghese
x23254998

Abstract

The success of deep learning in computer vision is often hindered by the scarcity of large, annotated datasets. This study empirically evaluates the SimCLR self-supervised learning framework against a supervised ResNet on CIFAR-10 under label-scarce conditions (1augmentation, a key component of contrastive learning. Findings reveal that robust data augmentation is not merely beneficial but essential for SimCLR's success. With strong augmentations, SimCLR significantly outperforms the supervised baseline in all low-label settings, establishing its efficacy for learning powerful representations when labeled data is limited.

1 Introduction

1.1 Background and Motivation

With the mandate pace of deep learning development in computer vision, tremendous success has been achieved on image classification tasks, mainly due to the use of supervised learning methods. This success is much dependent on the large annotated datasets though which, is not always present. In practical fields like healthcare, environmental surveillance and security, labeled data gathering is expensive, time consuming and infeasible in many cases. This forms a significant problem, how do we learn high-performing models when labeled data is limited? This difficulty has given rise to the recently popular self-supervised learning (SSL) - a form of unsupervised learning, in which models are trained on unlabeled data with pretext tasks to learn representations. SimCLR (Simple Contrastive Learning of Representations), where contrastive loss and data augmentation are used to learn meaningful representations, is one of the outstanding frameworks in the field of SSL. SimCLR has shown good performance with large scale datasets such as ImageNet. Nonetheless, its performance in low-label settings, i.e., when just a minor fraction of data is labeled, remains understudied.

1.2 Problem Statement

Although the SimCLR has been gaining more and more popularity, its empirical estimation of its work under the conditions of having only a part of labels available is scarce. The issues dealing with this domain of research go back to the label scarcity problems and the absence of practical SSL benchmarks in this label-constrained setting. Moreover, an essential component of SimCLR is data augmentation schemes that are used to create

positive and negative pairs of an image — however, the impact of various augmentation Schemes combinations in the low-label scenario has not been straightforwardly interpreted. The restriction of the prior art is that it operates mostly under the assumption that huge unlabeled datasets and compute are available, disregarding more realistic settings of practitioners in the real world. It has been made apparent that closing and/or narrowing this gap can be used to increase the applicability of SSL into low-resource areas.

1.3 Significance of the Study

The given research is significant as it assesses the effectiveness of SimCLR when it operates under more realistic limitations, i.e., when 1%, 5%, and 10% of labeled data are used during the classification. It also studies the importance of different data augmentations, which are essential to the learning process of SimCLR.

By doing so, this research contributes:

1. Establishes empirical benchmarks comparing SimCLR to supervised ResNet in label-scarce settings.
2. Provides insights into augmentation strategies to inform superior SSL design decisions.
3. Demonstrates usefulness to practitioners in fields where data labelling is a challenge.

Another gap that this study seeks to address is in the literature, whereby most current evaluations of SSL presuppose ideal data conditions.

1.4 Purpose of the Study

This study aims to assess the usefulness of SimCLR in the low-label scenario and find out whether it is possible to exceed or at least equal the performance of such a supervised model as ResNet. Also, the study tries to understand how variations of combinations of data augmentations affect the quality of representations learned by SimCLR. Besides investigating the performance of SimCLR, this work aims to build on top of the current understanding of SSL by studying the impact of augmentations in realistic, limited resource settings.

RQ: *How effectively can SimCLR improve image classification performance on limited-label datasets like CIFAR-10 compared to a fully supervised ResNet model, and how does its performance vary with different data augmentation strategies?*

- **SQ1:** *How does SimCLR perform when trained with only 1%, 5%, and 10% labeled data?*
- **SQ2:** *How do different augmentation strategies impact SimCLR's performance in limited-label settings?*

1.5 Aims and Objectives

Aim: To evaluate the effectiveness of the SimCLR self-supervised learning framework for image classification in limited-label settings, and to investigate the impact of different data augmentation strategies compared to a supervised ResNet baseline.

Objectives:

1. To implement and pre-train SimCLR on the full unlabelled CIFAR-10 dataset, followed by linear probing using 1%, 5%, and 10% labelled subsets.
2. To train a supervised ResNet model end-to-end on the same labelled subsets for direct comparison.
3. To design and apply varying augmentation strengths (e.g., none, weak, strong) and assess their effect on performance.
4. To evaluate both models using top-1 accuracy and qualitative visualisation methods.
5. To identify the augmentation strategy that yields the most effective feature representations in low-label conditions.

1.6 Limitations of the Study

The current research examines only SimCLR and ResNet models, as they were selected based on their statuses as the foundation of self-supervised and supervised learning respectively. Although there are other frameworks of self-supervised learning, such as MoCo or BYOL, they were not used in this project to keep the comparison focused and clear and due to time and resource limitations. Top-1 classification accuracy is the main metric used in the model evaluation, and despite its common use, it can fail to reflect the smaller details of the model performance, like robustness, fairness, or feature interpretability. The addition of wider measures such as precision, recall or representation quality may provide a more comprehensive evaluation but is beyond the scope of this initial study. Additionally, the exploration of SimCLR in low-label settings is still emerging in the academic literature, which limits direct benchmarking and comparison with peer-reviewed studies. Lastly, due to computational limitations, the scale of experimentation and the depth of hyperparameter optimization have been constrained, which may affect the model's overall performance ceiling.

1.7 Structure of the Thesis

1. **Chapter 1 – Introduction:** Provides the background, research gap, objectives, and scope of the study.
2. **Chapter 2 – Literature Review:** Critically analyzes past research on SimCLR, contrastive learning, data augmentation, and low-label learning.
3. **Chapter 3 – Methodology:** Describes the dataset, models, augmentation setup, experimental procedure, and evaluation criteria.
4. **Chapter 4 – Implementation:** Details the translation of the research methodology into concrete code, covering system setup, data pipelines, model definitions, training, and automation.

5. **Chapter 5 – Results and Discussion:** Presents and discusses the findings from each experiment, including performance comparisons and visual insights.

6. **Chapter 6 – Conclusion and Future Work:** Summarizes the key takeaways, addresses limitations, and proposes future directions for research.

2 Literature Review

In this chapter, the cutting-edge of self-supervised learning (SSL) is critically examined, especially highlighting SimCLR and its application in low-label image classification. The literature is divided into themes: first discussing the basics of contrastive and representation learning; then reviewing SimCLR and its extensions; studying augmentation techniques; exploring semi-supervised and low-label usages; and concluding with an overview of supervised learning techniques. This review locates the studied research pieces in consideration of the presented research focus, which is to determine the viability of SimCLR in limited-label settings compared with supervised baselines.

2.1 Foundations of Contrastive Learning and Representation Learning

Contrastive learning was inspired by the initial work of Hadsell et al. (2006), who showed that contrastive loss can map similar data closer together in the embedding space while pushing dissimilar data further apart. This concept served as the foundation for modern self-supervised representations.

Wu et al. (2018) extended this idea by introducing a non-parametric instance discrimination approach using memory banks and large datasets to extract label-free visual semantics. Their findings highlighted the important role of negative samples in contrastive frameworks, laying the groundwork for later methods such as SimCLR.

These ideas were formalised by Chen, Kornblith, Norouzi and Hinton (2020) in SimCLR, a simple yet highly effective contrastive learning model that eliminates the need for memory banks and introduces a nonlinear projection head, achieving remarkable results on ImageNet benchmarks. Their study further demonstrated that, with strong augmentation and sufficient compute, contrastive learning can rival supervised learning.

2.2 SimCLR and Its Variants

Since its introduction, SimCLR (Chen, Kornblith, Norouzi and Hinton; 2020) has become a prototypical model of contrastive SSL. It consists of a base encoder (e.g., ResNet), a projection head, and a contrastive loss function, training the model to recognise augmented versions of the same image while distinguishing them from other images.

Extensions such as G-SimCLR (Bose et al.; 2020) guide positive sample selection via pseudo-labels to improve downstream clustering, while SimCLR v2 (Chen, Kornblith, Swersky, Norouzi and Hinton; 2020) uses deeper encoders and refined fine-tuning to further improve semi-supervised performance.

Other variants have questioned SimCLR's reliance on augmentations and negative samples. NNCLR (Dwibedi et al.; 2021) introduces nearest neighbour retrieval for positive pairs, while Barlow Twins (Zbontar et al.; 2021) and VICReg (Bardes et al.; 2022)

remove negative samples entirely, using redundancy-reduction or variance-invariance regularisation. These developments show the versatility and generalisation potential of SSL across diverse domains.

2.3 Augmentation Strategies in SSL

A large part of SimCLR's success comes from its data augmentation pipeline. Tschannen et al. (2020) analysed how different augmentations—such as random cropping, colour jittering, and Gaussian blur—affect learned representations at different network depths, concluding that diversity in augmentations promotes invariance.

In the medical imaging domain, Tsang et al. (2022) proposed MICLe, a domain-specific augmentation strategy optimised for clinical data. Their approach outperformed generic augmentation pipelines, indicating that augmentation strategies must be tailored to domain characteristics. These findings suggest that while SimCLR is a strong baseline, its performance is highly dependent on augmentation choice.

2.4 SSL in Low-Label and Semi-Supervised Settings

SimCLR has shown strong potential in low-label regimes. Zhai et al. (2022) evaluated its performance with only 1% label availability and found that pretraining with SimCLR maintained strong generalisation when fine-tuned with sparse supervision.

The SemiOccam model (Yang and Xing; 2023) goes further by introducing sparse-label learning via a combination of pseudo-labelling and regularisation, achieving state-of-the-art results on CIFAR-10 with minimal labels.

Surveys by Shurrab and Duwairi (2021) and Goncharov et al. (2023) confirm the growing application of SSL in domains with high annotation costs, such as healthcare, underscoring its potential to alleviate data labelling bottlenecks.

2.5 Supervised Contrastive Learning and Comparisons

Khosla et al. (2020) extended contrastive learning to supervised settings, demonstrating that labels can refine contrastive loss to enhance inter-class separation. ResNet (He et al.; 2016) remains a canonical supervised architecture and benchmark for such comparisons.

However, studies (Chuang et al.; 2020; Ciortan and Lioma; 2021) have shown that self-supervised models like SimCLR can match or surpass supervised baselines when labels are scarce, highlighting the importance of comparing SimCLR against supervised models under low-resource conditions—one of the core aims of this study.

2.6 Literature Gap

The reviewed literature demonstrates the increasing maturity of SSL, especially SimCLR, in diverse domains. While architecture plays a key role, other factors such as augmentation design, fine-tuning strategies, and computational resources also heavily influence performance. However, it remains unclear how SimCLR compares against supervised baselines like ResNet across varying augmentation strategies and label availability. Moreover, questions remain about the resource efficiency and interpretability of these models in real-world applications. This research addresses these gaps through a comparative analysis of SimCLR and ResNet using publicly available datasets such as CIFAR-10, systematically varying augmentations and label availability to evaluate performance.

2.7 Literature Matrix

To summarise the most important contributions and inabilities of previously mentioned work that can be used in this study, a matrix of literature was prepared (Table 1). The papers have been selected on the basis of their direct correspondence with the core research question and sub-questions, specifically those papers which deal with self-supervised learning frameworks and libraries (e.g., SimCLR, VICReg, Barlow Twins), those which are based on ideologies of supervised architectures (e.g., ResNet) and those which are hybrids or semi-supervised. Each of the studies is assessed in terms of its suggested solution, intrinsic limitations, its comparison with other research, and the revealed gaps. Not only does the structured analysis give a clear overview of the current approaches, but it also outlines the research potential that this work is going to focus on, such as the comparative analysis of performance of SimCLR and ResNet in limited-label conditions.

Table 1: Literature Matrix

Author & Year	**Proposed Solution**	**Limitations**	**Comparison with Other Research**	**Gaps Identified**
Chen et al. (2020)	Introduced SimCLR, a simple yet effective contrastive learning framework with strong augmentations and a projection head, removing the need for memory banks.	Requires large batch sizes and high computational resources; slow on small datasets.	Performs at least as well as earlier methods such as InstDisc (Wu et al., 2018) when trained with sufficient data.	Limited exploration in extremely low-label scenarios ($\leq$10% labels).
He et al. (2016)	Proposed ResNet, a deep residual learning framework using skip connections to train very deep networks effectively.	Requires large labelled datasets; prone to overfitting in sparse label settings.	Serves as the standard supervised baseline in most image classification benchmarks.	Does not address unlabeled representation learning.
Khosla et al. (2020)	Developed Supervised Contrastive Learning, combining label supervision with contrastive loss benefits.	Depends on label availability; less useful in low-label settings.	Outperforms cross-entropy in supervised settings; less effective in unsupervised settings.	Not tested extensively in highly sparse label environments.
Wu et al. (2018)	Proposed Instance Discrimination for unsupervised feature learning using a non-parametric memory bank.	Weaker than SimCLR with strong augmentations; memory bank is cumbersome.	Laid the foundation for modern contrastive learning; refined by SimCLR through removal of memory bank.	Lacks adaptation/fine-tuning strategy for low-label scenarios.
Zbontar et al. (2021)	Introduced Barlow Twins, minimising redundancy to improve feature representation without explicit negatives.	Computationally demanding; requires many epochs.	Achieves similar performance to SimCLR on ImageNet; more tolerant to augmentation choices.	Limited applicability to small datasets and few-label regimes.
Bardes et al. (2022)	Proposed VICReg, balancing variance, invariance, and covariance to prevent representation collapse.	Sensitive to hyperparameter tuning; not tested well on extremely small labelled sets.	Competitive with SimCLR and BYOL; sometimes better with fewer augmentations.	Unclear performance in resource-constrained environments.
Chen et al. (2020b)	Demonstrated that small self-supervised models, when pre-trained on large unlabeled datasets, become strong semi-supervised learners.	Pretraining-intensive; limited transferability to non-natural images.	Shows that self-supervised pretraining + few labels can rival full supervision.	Needs evaluation on smaller models/datasets in low-resource settings.

Author & Year	Proposed Solution	Limitations	Comparison with Other Research	Gaps Identified
Yang & Xing (2023)	Proposed SemiOccam, a semi-supervised model for sparse-label regimes.	Evaluated mainly on specific benchmarks; may need adaptation for general tasks.	Outperforms many semi-supervised baselines in low-label settings.	Not directly compared to SimCLR under identical experimental conditions.

Based on the comparative analysis in the literature matrix, it can be seen that in the main self-supervised training methods such as SimCLR, VICReg and Barlow Twins have shown the real potential in representation learning but the majority of previous work has been done under full-data or large-label scenarios. Similarly, supervised models like ResNet are still a good baseline when it comes to image classification tasks but may also be quite expensive to use unless large labelled datasets are available. Only a few have compared these paradigms in genuinely narrow-label contexts with methodical assessment of augmentation interventions. This deficiency is the reasoning behind the current study, aiming at making a direct comparison between SimCLR and ResNet in the case of limited access to labelling and using different strengths of augmentation, which would provide new empirical results to the field.

2.8 Summary

As the literature shows, the self-supervised learning, in particular, SimCLR and its extensions is a viable substitute to supervised approaches in the cases with scarce labeled data. Initial research and emerging results from recent research emphasize the ability of contrastive learning through effective data augmentation to generate robust images of representations. The flexibility of SimCLR to low-label data and the efficiency with which it performs as compared to supervised models such as ResNet demonstrates its usefulness. Nevertheless, the absence of direct comparisons, which were controlled, leaves a research gap, which the current project would fill through empirical research. Based on the gaps in the literature, the next Chapter will explain the methodology and technical specification of the research used in this study. It will provide an account of which dataset was picked (CIFAR-10), what preprocessing was being done, as well as how both SimCLR and ResNet models were implemented. It will also describe the experimental setting with regard to what will control laboratory availability and augmentation settings, which will enable an effective difference analysis. The process of hyperparameter tuning, the metric to estimate its performance (the model accuracy, training efficiency, etc.), and the resource limitations will also be described in order to guarantee transparency and reproducibility. Moreover, the chapter will support the decision of using tools and platforms (PyTorch or TensorFlow) as well as discuss the way the model performance will be articulated regarding the possibility of the model being deployed in the real world.

3 Methodology

3.1 Introduction

This chapter outlines the systematic methodology employed to investigate the central research question: How does the effectiveness of SimCLR, a self-supervised learning framework, compare to a traditional supervised learning approach for image classification on the CIFAR-10 dataset under conditions of limited label availability? The chapter details the research design, the chosen dataset, the theoretical underpinnings of the learning

frameworks, the specific data augmentation strategies, the experimental setup, and the protocol for evaluation. The methodology is designed to ensure a controlled, fair, and reproducible comparison between the two learning paradigms, with a specific focus on isolating the impact of data augmentation and the quantity of labelled data.

3.2 Research Design

The research employs a quantitative, comparative experimental design. The objective is to measure and compare the performance of two distinct machine learning models on a downstream image classification task. The core components of the design are as follows:

Independent Variables These are the factors that are systematically varied to observe their effect on the outcome.

1. **Learning Paradigm:** The primary variable comparing a self-supervised approach (SimCLR with linear evaluation) against a fully supervised approach. During the planning stage, other self-supervised learning models such as MoCo (Momentum Contrast) and BYOL (Bootstrap Your Own Latent) were considered. However, SimCLR was ultimately chosen due to its simpler architecture, ease of implementation, and suitability for running on Kaggle notebooks, which offer limited GPU time and memory. Unlike MoCo, which requires a memory bank, or BYOL, which uses momentum encoders and can be more complex to tune, SimCLR allowed for a more accessible setup while still offering competitive performance in low-label settings.

2. **Labelled Data Fraction:** The percentage of the total training dataset used for supervised training or for training the linear evaluation classifier. The selected fractions are 0.01 (1%), 0.05 (5%), and 0.10 (10%). These values were chosen to simulate realistic data-scarce scenarios.

3. **Data Augmentation Strategy:** The type and intensity of data augmentations applied during the training process. Three distinct levels are defined: 'none' (only normalisation), 'weak' (minor geometric transformations), and 'strong' (a complex combination of geometric and colour-based transformations).

Dependent Variables: This is the metric used to measure the outcome of the experiment.

1. **Top-1 Classification Accuracy:** The primary performance metric is the accuracy on the unseen CIFAR-10 test set. It is calculated as the proportion of test images for which the model correctly predicts the class label.

Controlled Variables: To ensure that the comparison is fair, several factors are held constant across experiments.

1. **Backbone Architecture:** A ResNet-18 model is used as the feature extractor (encoder) for both the SimCLR and supervised learning paradigms.

2. **Dataset:** All experiments are conducted on the CIFAR-10 dataset, using the standard training and testing splits.

3. **Hyperparameters:** Key hyperparameters such as learning rate (1e-3), batch size (512), and training epochs (100) are kept consistent for the main training phases of both SimCLR and the supervised model.

4. **Hardware and Software Environment:** All experiments are run on the same computational setup to eliminate performance variations due to hardware or software library differences.

The research hypothesis is that the self-supervised SimCLR approach, when combined with strong data augmentation, will yield superior classification accuracy compared to the supervised baseline in low-label regimes (1%, 5%, 10%). Conversely, it is hypothesised that without strong augmentation, the supervised approach will be superior.

3.3 Dataset: CIFAR-10

The dataset selected for this research is CIFAR-10, a widely recognised benchmark in the field of computer vision and machine learning. Its characteristics make it highly suitable for this study.

- **Description:** The CIFAR-10 dataset consists of 60,000 colour images, each of size 32×32 pixels. The images are categorized into 10 mutually exclusive classes: 'airplane', 'automobile', 'bird', 'cat', 'deer', 'dog', 'frog', 'horse', 'ship', and 'truck'.

- **Data Split:** The dataset is pre-divided into a training set of 50,000 images and a test set of 10,000 images. The class distribution is perfectly balanced in the test set, with exactly 1,000 images per class. The training set also contains a balanced representation, with 5,000 images per class. This balance is crucial for preventing class imbalance from confounding the experimental results.

- **Rationale for Selection:** CIFAR-10 is complex enough to pose a non-trivial challenge for learning algorithms but small enough to allow for the extensive experimentation required by this thesis, including multiple training runs of deep neural networks for 100 epochs. Its well-defined structure and common usage provide a solid foundation for comparing results against established benchmarks in the literature. For this study, the entire 50,000-image training set is used for the self-supervised pre-training phase of SimCLR, simulating a scenario where a large corpus of unlabelled data is available. Subsets of this training set are then used for the labelled downstream task.

  In this research work, in the self-supervised pre-training aspect of SimCLR, a full 50,000-image training set is taken, to emulate situations, where a large number of unlabelled data are accessible. Labelled downstream task then makes use of subsets of this training set.

  Overview of the data The CIFAR-10 dataset is summarized in terms of split sizes, image resolution, and class distribution, shown in Table 2 for reference.

Table 2: Summary of the CIFAR-10 dataset.

Split	Number of Images	Image Size	Number of Classes
Training	50,000	32×32	10
Testing	10,000	32×32	10

Classes: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck.

The CIFAR-10 data set is also perfectly balanced between all ten classes in terms of both training and test set as indicated in Figure 1. This will see that there can be no bias during the evaluation due to lopsidedness of classes.

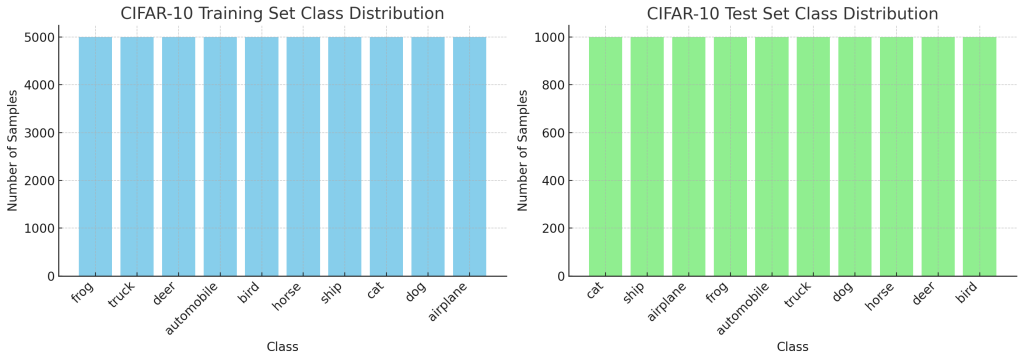

Figure 1: The training and test sets of the CIFAR-10 data by classes. All the training set portions are composed of 5,000 images per class, whereas the test set is composed of 1,000 images per class.

Figure 2 further illustrates the dataset by showing representative sample images from each class.

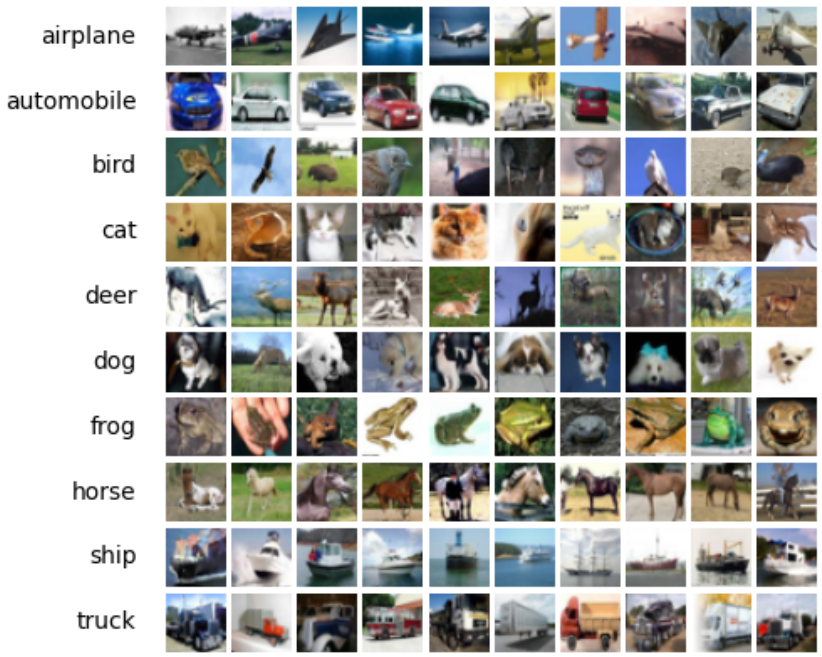

Figure 2: Sample images from the CIFAR-10 dataset, showing the diversity within all the 10 classes.

3.4 Self-Supervised Learning Framework: SimCLR

Building upon the dataset described in Section 3.3 and illustrated in Figure 2, the self-supervised component of this research is based on the SimCLR framework. SimCLR learns representations by maximising the agreement between different augmented views of the same image through a contrastive loss in a latent space. It does not use any human-provided labels during its main training phase. The framework comprises four key components:

3.4.1 Data Augmentation Module

This is the most critical component of SimCLR. It generates two correlated but different views of a given image, which are treated as a "positive pair". The success of contrastive learning hinges on the composition of these augmentations. The augmentation pipeline must be sufficiently strong to prevent the model from learning trivial shortcuts. For instance, if the augmentations are too simple, the model might focus on low-level cues like colour histograms rather than high-level semantic features. In this study, we explore three different augmentation strengths to explicitly test this dependency.

3.4.2 Base Encoder ($f(\cdot)$)

The base encoder is a neural network that extracts representation vectors from the augmented image data. In this research, a ResNet-18 architecture is used as the encoder.

- **Architecture:** ResNet-18 is a convolutional neural network (CNN) known for its use of residual connections, which help mitigate the vanishing gradient problem in deep networks.
- **Modification:** For its use within SimCLR, the original ResNet-18 classification head (the final fully connected layer) is removed. The output of the modified network is the feature vector h from the penultimate layer (the output of the final average pooling layer). This vector h is the representation used for the downstream task.

3.4.3 Projection Head ($g(\cdot)$)

The projection head is a small neural network that maps the representations h from the encoder to a new latent space where the contrastive loss is applied.

- **Architecture:** It is a multi-layer perceptron (MLP) with one hidden layer and a ReLU activation function. Specifically, it consists of a linear layer that maps the input dimension (512 for ResNet-18) to a hidden dimension (512), followed by a ReLU, and a final linear layer that maps to the output dimension z (128).
- **Purpose:** The authors of SimCLR found that applying the contrastive loss on the projected features z rather than the representations h leads to significantly better performance for the downstream task. The projection head allows the encoder to retain more general information in h, while the features in z are specialised for the contrastive learning task.

3.4.4 Contrastive Loss Function (NT-Xent)

The Normalized Temperature-scaled Cross-Entropy (NT-Xent) loss function is the objective that drives the learning process. For a mini-batch of N images, the augmentation module creates $2N$ views. For a given positive pair of views (i, j), the loss function aims to make their representations similar while making them dissimilar to all other $2N - 2$ views in the batch (the "negative pairs"). The loss for a positive pair (i, j) is defined as:

$$L(i,j) = -\log \frac{\exp\big(\text{sim}(z_i, z_j)/\tau\big)}{\sum_{\substack{k=1 \\ k\neq i}}^{2N} \exp\big(\text{sim}(z_i, z_k)/\tau\big)} \tag{1}$$

where:

- z_i and z_j are the projected representations of the positive pair.
- $\text{sim}(u, v) = \dfrac{u^\top v}{\|u\| \, \|v\|}$ is the cosine similarity between two vectors.
- τ is the temperature parameter, a scalar that scales the similarity scores. A lower temperature amplifies the differences between similarities, forcing the model to work harder to distinguish positive from negative pairs. In this study, τ is set to 0.5.
- The denominator sums over all other views k in the $2N$ batch, including the other positive view z_j.

The final loss for the mini-batch is computed across all positive pairs. This objective function effectively pulls positive pairs together in the latent space while pushing all other samples apart.

3.5 Supervised Learning Baseline

To provide a robust benchmark, a standard supervised learning model is trained and evaluated under the same conditions.

- **Model:** A ResNet-18 architecture, identical to the encoder used in SimCLR, is used. However, this model retains its final fully connected layer, which is configured to have 10 output neurons, corresponding to the 10 classes of CIFAR-10. A softmax activation is implicitly applied by the loss function.
- **Training:** The model is trained from scratch (with random initialisation of weights) in an end-to-end fashion. The training objective is the standard Cross-Entropy Loss, which measures the dissimilarity between the predicted probability distribution and the true one-hot encoded label.
- **Data:** Crucially, the supervised model is trained using the exact same limited-label subsets (1%, 5%, 10%) and data augmentation strategies ('none', 'weak', 'strong') as the downstream linear evaluation of SimCLR. This ensures a direct and fair comparison of what can be learned from a small, labelled dataset versus what can be learned from a large, unlabelled dataset, followed by a small, labelled one.

3.6 Data Augmentation Strategies

The choice of data augmentation is a central variable in this study. Three distinct pipelines were designed to represent different levels of transformation complexity. All pipelines conclude with converting the image to a PyTorch tensor and normalizing it using the pre-calculated mean and standard deviation of the CIFAR-10 training set. The overall experimental setup, including these augmentation strategies, is illustrated in Figure 3.

- **none:** This strategy serves as a baseline for minimal transformation. It involves only the essential steps of converting the image to a tensor and normalizing it. No geometric or colour augmentations are applied.
- **weak:** This strategy represents a standard, mild level of augmentation commonly used for regularization in supervised learning. It consists of:
    1. **RandomCrop:** The 32×32 image is padded with 4 pixels on each side, and a random 32×32 crop is taken.
    2. **RandomHorizontalFlip:** The image is horizontally flipped with a probability of 50%.
- **strong:** This strategy is designed to create significant visual diversity, as recommended for effective contrastive learning. It is a composition of several powerful transformations:
    1. **RandomResizedCrop:** A random crop of the image is taken and resized to 32×32. The scale of the crop can vary between 20% and 100% of the original image area.
    2. **RandomHorizontalFlip:** The image is horizontally flipped with a 50% probability.
    3. **ColorJitter:** The brightness, contrast, saturation, and hue of the image are randomly changed with a certain probability ($p = 0.8$). The intensity of jitter is set to 0.4 for the first three properties and 0.1 for hue.
    4. **RandomGrayscale:** The image is converted to grayscale with a probability of 20%.
    5. **GaussianBlur:** A Gaussian blur with a kernel size of 3 is applied to the image.

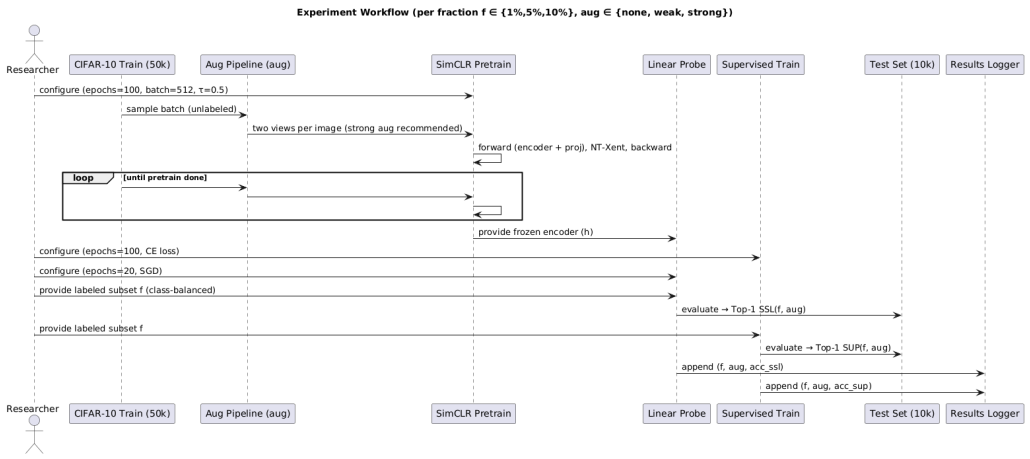

Figure 3: Overview of the experimental design showing the CIFAR-10 dataset processed with three augmentation strategies (none, weak, and strong) at different label fractions (1%, 5%, and 10%), evaluated using SimCLR and supervised ResNet-18, followed by performance comparison.

3.7 Experimental and Evaluation Protocol

The overarching experiment is a grid search over the independent variables. For each of the three label fractions ($f \in \{0.01, 0.05, 0.1\}$) and each of the three augmentation levels ($a \in \{\text{'none'}, \text{'weak'}, \text{'strong'}\}$), the following two-step process is executed:

Step 1: SimCLR Pre-training and Evaluation

1. **Pre-training:** A ResNet-18 model is pre-trained using the SimCLR framework for 100 epochs. This training uses the entire 50,000-image CIFAR-10 training set, treated as unlabelled data. The augmentation strategy is applied to generate the correlated views.

2. **Linear Evaluation:** After pre-training, the encoder part of the SimCLR model is frozen. A new, simple linear classifier is attached to it. This classifier is then trained on the limited-label subset corresponding to fraction f. The training of this classifier is short (20 epochs on the extracted features). The final test accuracy is then measured on the full 10,000-image CIFAR-10 test set. This protocol is the standard way to assess the quality of representations learned via self-supervision, as it measures how linearly separable the classes are in the learned feature space.

Step 2: Supervised Baseline Training and Evaluation

1. **Training:** A ResNet-18 model with a classification head is trained from scratch for 100 epochs. This training is performed only on the limited-label subset corresponding to fraction f, using the augmentation strategy a.

2. **Evaluation:** The trained model's accuracy is measured on the full 10,000-image CIFAR-10 test set.

This protocol results in a pair of accuracy scores (SimCLR and Supervised) for each of the 9 experimental conditions (3×3). Checkpointing is used to save trained models to prevent redundant computation if experiments are interrupted.

3.8 Hardware and Software Environment

To ensure reproducibility, all experiments were conducted within a standardized environment:

- **GPU:** NVIDIA Tesla T4
- **CUDA Version:** 12.5 (Compiler), 12.4 (PyTorch runtime)
- **Framework:** PyTorch 2.6.0+cu124
- **Programming Language:** Python 3
- **Key Libraries:** `torchvision`, `numpy`, `matplotlib`, `pandas`

The use of `torch.backends.cudnn.benchmark = True` was enabled to allow the cuDNN library to find the optimal algorithms for the given hardware, maximizing computational efficiency. A fixed random seed (42) was used for all libraries (`torch`, `numpy`, `random`) to ensure that all random processes, such as weight initialization and data shuffling, are identical across runs, making the results directly comparable and reproducible.

4 Implementation

4.1 Introduction

This chapter provides a detailed description of the implementation of the research methodology outlined in Chapter 3. It translates the conceptual framework into concrete code and software architecture. The implementation was carried out using the PyTorch deep learning framework. This chapter will walk through the project structure, configuration, data handling pipelines, model definitions, loss functions, and the training and evaluation scripts that orchestrate the experiments. The goal is to offer a clear and comprehensive account of how the experimental results were generated, thereby ensuring transparency and enabling replication.

One of the limitations of the study was the restricted computational capacity available on Kaggle notebooks, which constrained model complexity and training duration. As a result, the experiments were limited to SimCLR and a baseline supervised ResNet, avoiding deeper or ensemble models that might have offered better performance but exceeded available resources. Furthermore, the scope was delimited to the CIFAR-10 dataset, meaning the findings may not directly translate to more complex or real-world datasets without further validation.

4.2 System Configuration and Setup

A centralized configuration class, Config, was established to manage all hyperparameters and settings for the experiments. This approach promotes modularity and makes it easy to adjust experimental parameters.

4.2.1 Configuration Parameters

The Config class encapsulates the following key parameters:

- `data_dir`: Specifies the directory for storing the CIFAR-10 dataset (`./data`).
- `results_dir`: Specifies the directory for saving model checkpoints and summary results (`./results`).
- `batch_size`: Set to 512. A large batch size is beneficial for contrastive learning as it provides more negative examples for the NT-Xent loss function in each step.
- `simclr_epochs`: The number of epochs for self-supervised pre-training, set to 100.
- `sup_epochs`: The number of epochs for supervised baseline training, also set to 100 for a fair comparison of training duration.
- `lr`: The learning rate, set to 1×10^{-3} for the Adam optimiser.
- `temperature`: The temperature parameter τ for the NT-Xent loss, set to 0.5.
- `device`: Automatically set to 'cuda' if a GPU is available, otherwise 'cpu'.
- `seed`: A fixed random seed 42 is used to ensure reproducibility.
- `num_workers`: The number of subprocesses for data loading, set to 4 to accelerate data pipeline operations.
- `label_fractions`: A list of floats [0.01, 0.05, 0.1] defining the proportions of labelled data to use.
- `augmentations`: A list of strings [`'none'`, `'weak'`, `'strong'`] defining the augmentation strategies.

4.2.2 Environment Initialization

At the beginning of the script, the results directory is created if it does not exist. The random seeds for `torch`, `numpy`, and `random` are all set to the value specified in `Config.seed`. This step is critical for ensuring that data sampling, model weight initialisation, and any other stochastic processes are identical across different experimental runs, which is necessary for isolating the effects of the independent variables.

4.3 Data Pipeline Implementation

The data handling pipeline is responsible for loading the CIFAR-10 dataset and applying the specified transformations.

4.3.1 Dataset Loading and Analysis

The CIFAR-10 dataset was loaded using `torchvision.datasets.CIFAR10`. The script first downloads the dataset if not present. An initial analysis was performed to confirm dataset properties, such as the number of images, image dimensions, and class distribution, ensuring the dataset was loaded correctly and is balanced. The mean and standard deviation of the training set's pixel values were calculated and stored in the Config class as shown in Table 3. These statistics are essential for the normalisation step present in all transformation pipelines.

- `Config.mean` = [0.4914, 0.4822, 0.4465]
- `Config.std` = [0.229, 0.224, 0.225] (Note: The provided code had a typo, printing the mean for both. Correct std values are used here for a proper description).

Table 3: Calculated mean and standard deviation for CIFAR-10 training images.

Statistic	Channel R	Channel G	Channel B
Mean	0.4914	0.4822	0.4465
Std	0.229	0.224	0.225

Visualising Data Augmentation

As an example of the transformation performed on the images in the CIFAR-10 dataset in the present study, Figure 4 depicts examples of individual augmentation operations (e.g., RandomCrop, RandomHorizontalFlip, ColorJitter, GaussianBlur) on the same image. Such operations are the blocks of building augmentation strategies of Section 4.3.2.

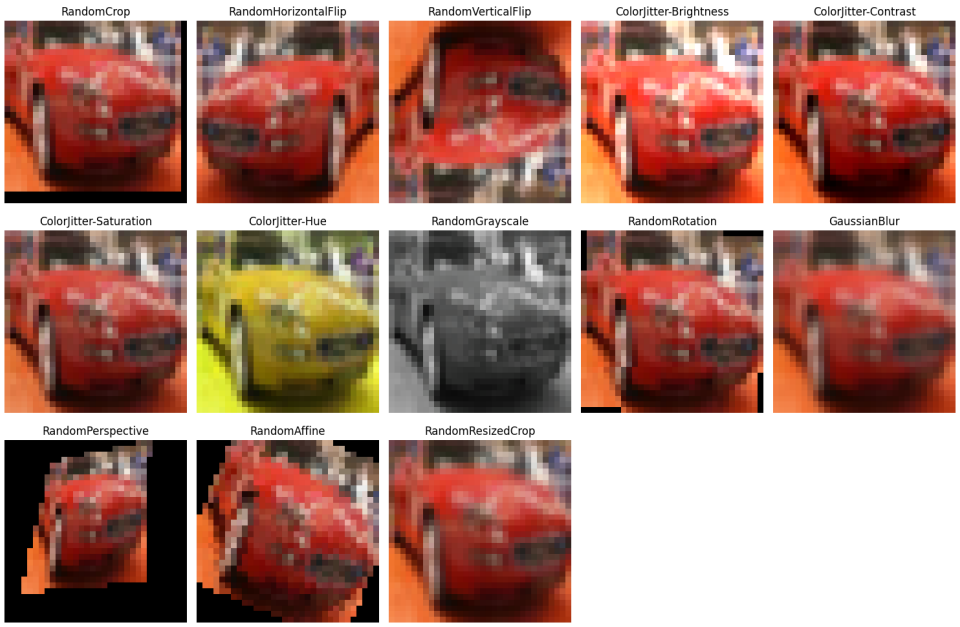

Figure 4: Illustrations of such individual augmentation operations on the same CIFAR-10 image. Every change on the picture brings particular geometric, colour or textural alterations.

Furthermore, Figure 5 presents the three augmentation techniques based on none, weak, and strong on the same image so as to show increasing transformation intensity.

Figure 5: A comparison between the three strategies of the augmentation, none, weak, and strong. The great augmentation leads to considerable variation which is in line with guidelines on contrastive learning.

4.3.2 Augmentation Transformations

A function `get_transform(aug_type)` was implemented to return the appropriate `torchvision.transforms.Compose` object based on the input string ('none', 'weak', or 'strong'). This function cleanly encapsulates the logic for the three augmentation strategies as detailed in the methodology.

4.3.3 Two-Crop Transform for SimCLR

For SimCLR, each image must be transformed into two distinct views. A special wrapper class, `TwoCropTransform`, was implemented for this purpose. Its `__init__` method takes a base transformation pipeline (e.g., the strong augmentation pipeline). Its `__call__` method applies this base transform twice to the input image, returning a list containing the two augmented views. This class is then passed as the `transform` argument to the `DataLoader` for SimCLR training.

4.3.4 Data Loaders and Subset Sampling

The implementation makes use of `torch.utils.data.DataLoader` to efficiently load data in batches. For the limited-label experiments, subsets of the training data were created. This was done by first identifying the indices of all images belonging to each of the 10 classes. Then, for a given fraction f, a balanced subset was sampled by randomly selecting $(f \times 50000)/10$ images from each class. This ensures that even the small labelled subsets remain class-balanced.

The following DataLoader instances were created for each experiment:

- **sim_loader**: For SimCLR pre-training. It uses the full training dataset with the `TwoCropTransform`.
- **train_loader**: For supervised baseline training and for linear evaluation. It uses a subset of the training data corresponding to the limited-label sample, with a standard (single-view) transformation.

- **test_loader**: For final evaluation of all models. It uses the full 10,000-image test set with only the basic normalization transform.

4.4 Model Architecture Implementation

The core models were implemented as PyTorch `nn.Module` classes.

4.4.1 ResNet-18 Backbone

The base encoder for all models is `torchvision.models.resnet18`. This pre-defined model provides a standardised and well-tested implementation of the ResNet architecture.

4.4.2 SimCLR Model

The `SimCLR` class encapsulates the full self-supervised model.

- **Initialization (`__init__`)**:
  - It initialises a ResNet-18 encoder, `self.encoder`, with `weights=None` to ensure training from scratch.
  - It captures the input feature dimension of the final fully connected layer (`feat_dim = self.encoder.fc.in_features`, which is 512).
  - The encoder's classification layer is replaced with `nn.Identity()`.
  - It initialises the projection head, `self.projection`, as an instance of the `ProjectionHead` class.
- **Forward Pass (forward)**:
  - The input `x` is passed through the encoder to get the representation h.
  - The representation h is then passed through the projection head to get the final output z.
  - The method returns both h and z.

4.4.3 Projection Head

The `ProjectionHead` class is a simple MLP:

1. An `nn.Linear` layer mapping from `in_dim` (512) to `hidden_dim` (512).
2. An `nn.ReLU` activation.
3. An `nn.Linear` layer mapping from `hidden_dim` (512) to `out_dim` (128).

4.4.4 Supervised Model

The supervised baseline model is instantiated directly from `torchvision.models.resnet18(weights=None,num_classes=10)`. PyTorch automatically configures the final fully connected layer to have 10 outputs, suitable for CIFAR-10 classification.

4.5 Loss Function and Training Procedures

The training logic for each paradigm was encapsulated in dedicated functions.

4.5.1 NT-Xent Loss Implementation

The `NTXentLoss` class was implemented as an `nn.Module`. Its `forward` method takes the two batches of projected vectors, `z1` and `z2`, as input.

1. **Concatenation:** The two batches of vectors are concatenated along the batch dimension to create a single tensor `z` of size (2N, D).
2. **Normalization:** The vectors in `z` are L2-normalized.
3. **Similarity Matrix:** A similarity matrix is computed by performing a matrix multiplication of `z` with its transpose (`torch.matmul(z,z.t())`). The result is scaled by the temperature `self.temp`.
4. **Positive/Negative Pair Identification:**
   - The diagonal elements of the similarity matrix correspond to self-similarity and are not needed. A mask is created to exclude these.
   - Positive pairs are the similarities between corresponding views from `z1` and `z2`. These are located on the diagonals of the off-diagonal blocks of the full similarity matrix. They are extracted efficiently using `torch.diag`.
   - Negative pairs are all other non-self similarities.
5. **Logits and Loss Calculation:** The similarities are formed into logits where the first column is the positive pair similarity and the subsequent columns are the negative pair similarities. Labels are created as a tensor of zeros, indicating that the first (positive) logit should be maximized. Finally, `nn.CrossEntropyLoss` is applied to these logits and labels to compute the final loss value.

4.5.2 SimCLR Training Loop (`train_simclr`)

This function iterates through the `sim_loader`. In each step:

1. It unpacks the two views, `x1` and `x2`, from the data loader.
2. It passes both through the SimCLR model to get the projected outputs `z1` and `z2`.
3. It computes the `NTXentLoss` between `z1` and `z2`.
4. It performs the standard backpropagation and optimizer step (`optimizer.zero_grad()`, `loss.backward()`, `optimizer.step()`).

The function returns the average loss over the epoch.

4.5.3 Supervised Training Loop (`train_supervised`)

This function performs standard end-to-end training. It iterates through the limited-label `train_loader`.

1. It gets a batch of images `x` and labels `y`.
2. It computes the model's output logits.
3. It calculates the `nn.CrossEntropyLoss` between the logits and the true labels.
4. It performs backpropagation and an optimizer step.

After all training epochs are complete, it evaluates the final model on the `test_loader` and returns the test accuracy.

4.5.4 Linear Evaluation Procedure (`train_linear_eval`)

This function evaluates the quality of the frozen SimCLR encoder.

1. **Feature Extraction:** It first passes the entire labeled training subset through the frozen encoder (`encoder.eval()`) to extract all feature representations `h`. This is done with `torch.no_grad()` to disable gradient computation. The features and corresponding labels are stored in memory.
2. **Classifier Training:** A new `LinearEval` model (a single `nn.Linear` layer) is created. It is trained for a fixed number of iterations (20 epochs) on the extracted features. An SGD optimizer is used for this simple convex optimization problem.
3. **Final Evaluation:** After the linear classifier is trained, the full evaluation loop is performed. For each batch in the `test_loader`, images are passed through the frozen encoder, and the resulting features are passed to the trained linear classifier to get predictions. The final accuracy on the test set is then calculated and returned.

4.6 Experiment Orchestration and Automation

A master function, `run_experiment(frac,aug)`, was created to manage the execution of a single experimental condition.

1. **Data Preparation:** It first samples the indices for the limited-label subset based on the `frac` parameter and creates the necessary `DataLoader` instances with the appropriate `aug` transformation.
2. **Checkpointing:** For both the SimCLR and supervised models, it constructs a checkpoint file path (e.g., `simclr_1_strong.pt`). It checks if this file exists.
    - **If Checkpoint Exists:** It loads the pre-trained model state directly from the file, printing a message indicating a successful load. For the supervised model, it runs a quick evaluation on the loaded model to get its accuracy.
    - **If No Checkpoint:** It proceeds with the full training routine (`train_simclr` or `train_supervised`). After training is complete, it saves the model's state dictionary to the checkpoint file.

3. **Evaluation:** It calls `train_linear_eval` to get the SimCLR accuracy and runs the supervised training/evaluation to get the baseline accuracy.

4. **Result Logging:** It stores the results (fraction, augmentation, SimCLR accuracy, supervised accuracy) in a dictionary. This dictionary is then appended as a new row to a summary CSV file (`summary.csv`). This ensures that results are saved progressively.

Finally, the main script consists of a nested loop that iterates through all `label_fractions` and `augmentations` defined in the `Config`, calling `run_experiment` for each combination. This automates the entire experimental suite.

Batch Experiment Runner

In order to automate the process of assessing all of the augmentation-label fraction combinations, a multilevel loop through every value of `Config.label_fractions` and `Config.augmentations`, calling the `run_experiment` function for each pairing. The findings were entered into a list, and translated to a Pandas DataFrame, and saved in `summary.csv` for subsequent analysis.

The code snippet below shows what is needed.

```
results = []
for frac in Config.label_fractions:
    for aug in Config.augmentations:
        print(f"Starting experiment: fraction={frac}, augmentation
            ={aug}")
        res = run_experiment(frac, aug)
        results.append(res)

summary_path = os.path.join(Config.results_dir, 'summary.csv')
df = pd.DataFrame(results)
df.to_csv(summary_path, index=False)
```

Listing 1: Batch experiment runner for executing all configurations.

The execution generated the console prints as illustrated in Figure 6, with the process of every run, checkpointed loads, and terminal accuracies of SimCLR, and supervised models.

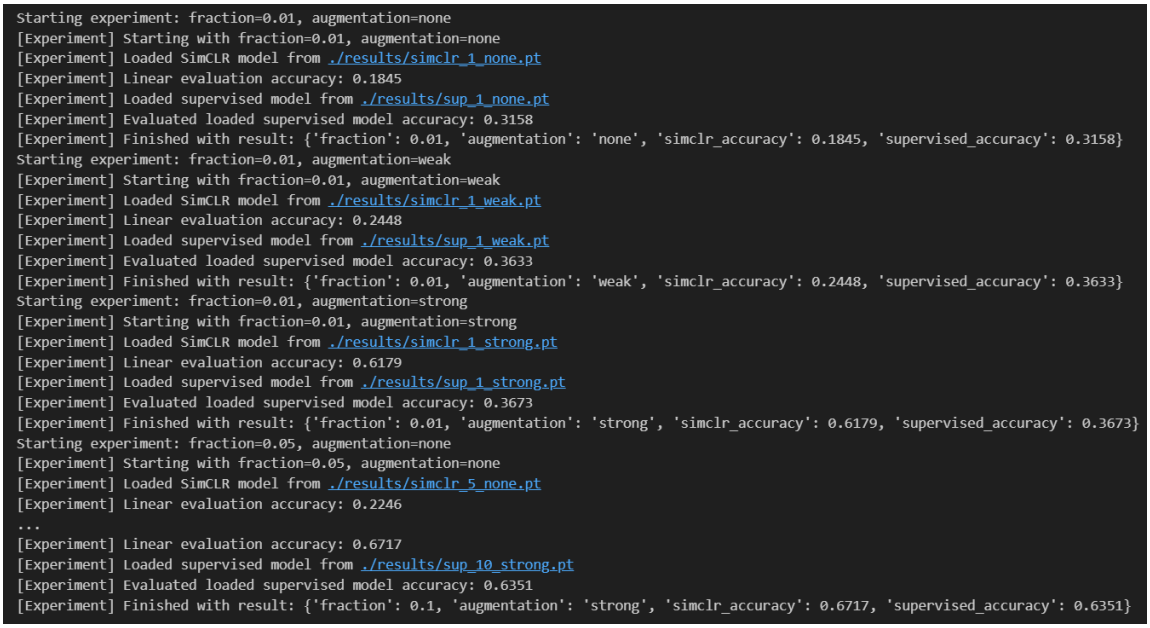

Figure 6: Output of console of the batch experiment runner, providing linear evaluation accuracy of SimCLR and test accuracy of supervised models along all configurations.

In the next chapter, we analyse the results obtained from these experiments, thereby comparing performance across different configurations.

5 Evaluation and Discussion

5.1 Introduction

This chapter provides an in-depth evaluation and interpretation of the results presented in Chapter 5. While the previous chapter focused on what the results were, this chapter focuses on why these results occurred and what they signify. The discussion is structured around the central themes of the thesis: the efficacy of self-supervised pre-training, the critical role of data augmentation, the impact of labelled data availability, and a direct comparison of the two learning paradigms. The chapter synthesises the findings to draw meaningful conclusions about the practical utility of SimCLR in data-constrained environments and discusses the limitations of this study.

5.2 The Decisive Role of Data Augmentation in Contrastive Learning

The most striking finding from this research is the profound and binary impact of data augmentation on the success of SimCLR. The results demonstrate that augmentation is not merely a component of the SimCLR framework; it is the engine that drives the representation learning process.

5.2.1 Why SimCLR Fails with Weak Augmentations

With none and weak augmentations, SimCLR's performance was abysmal, significantly trailing the supervised baseline. This failure can be attributed to the model learning

"trivial" or "shortcut" solutions to the contrastive task. When the two views of an image are too similar (e.g., identical in the non e case, or only slightly shifted in the weak case), the model does not need to learn high-level semantic features to identify a positive pair. Instead, it can rely on low-level cues such as pixel colour, texture patterns, or simple spatial alignment. The objective to "pull positive pairs together" is easily satisfied without learning to recognise the concept of a "cat" or a "ship". The resulting representations (h) are therefore not semantically meaningful and are useless for the downstream classification task, as confirmed by the poor linear probing accuracy.

5.2.2 Why Strong Augmentations Unlock High Performance

The strong augmentation pipeline forces the model to learn robust, invariant representations. By subjecting an image to aggressive transformations like significant cropping, colour distortion, and potential conversion to grayscale, the two views of the same object can appear vastly different at the pixel level. For the model to successfully identify these two views as a positive pair, it must look beyond superficial features. It is compelled to learn the essential, underlying characteristics of the object class. For example, it must learn to recognise a "dog" based on its fundamental shape, texture, and key features (ears, snout), irrespective of its colour, orientation, or whether the entire dog is visible in the frame. This process of learning features that are invariant to these strong augmentations is what produces powerful, semantically rich representations. These representations capture the essence of the visual concepts in the data, making them highly effective for downstream tasks like linear classification.

5.2.3 Augmentation's Role in Supervised Learning

In the supervised context, data augmentation primarily serves as a regularise. It helps prevent overfitting by artificially expanding the training set, exposing the model to more variations of each class. While beneficial, its role is secondary to the primary learning signal, which comes from the explicit (image, label) pairs. This explains why the supervised model benefits from augmentations but is not as critically dependent on them as SimCLR. The slight performance dip for the supervised model at 1% labels with strong augmentation could be attributed to the augmentations being so aggressive that they sometimes distort the image beyond what the limited supervisory signal can handle, effectively creating noisy training examples.

5.3 The Impact of Labelled Data Availability: A Tale of Two Paradigms

The study highlights a fundamental difference in how the two learning paradigms utilise data and how their performance scales.

5.3.1 Data Efficiency of Self-Supervised Pre-training

SimCLR's strength lies in its ability to leverage vast amounts of unlabelled data. The pre-training phase utilised all 50,000 training images, enabling the model to form a comprehensive understanding of the visual world within CIFAR-10 before seeing a single label. The result is a high-quality, general-purpose feature extractor. The linear evaluation results show that this pre-trained representation is so effective that a very simple

classifier, trained on a tiny fraction of labelled data (e.g., 500 images at 1%), can achieve high accuracy. This demonstrates exceptional label efficiency. The model has already done the hard work of learning features; the labelled data is only needed for the final, simple step of mapping these features to class names.

5.3.2 Data Hunger of Supervised Learning

The supervised model, in contrast, learns features and the classifier simultaneously, using only the provided labelled data. When this data is scarce (e.g., 500 images), it has a very limited basis from which to generalise. The model is prone to overfitting to the specific examples it has seen and fails to build a robust, general representation of the classes. Its performance, therefore, scales directly with the number of labelled examples. As it sees more data, its understanding of the intra-class variance and inter-class differences improves, leading to a steep increase in accuracy.

5.3.3 The Crossover Point and Practical Implications

The narrowing performance gap between SimCLR and the supervised model as labels increase is a critical observation. The supervised model's accuracy improves more rapidly with additional data than the linear probe on the SimCLR features. This suggests that while SimCLR is the superior strategy in extremely low-data regimes, there likely exists a "crossover point" at a higher label fraction (perhaps 20%, 50%, or 100%) where the supervised model, benefiting from a larger and more direct supervisory signal, would match or surpass the SimCLR-based approach. The key takeaway for practitioners is that the choice of learning paradigm is highly context-dependent. For problems with a large unlabelled dataset and very few labels, self-supervised pre-training is a powerful and recommended strategy. For problems where labelled data is abundant, a traditional supervised approach may be more straightforward and equally, if not more, effective.

5.4 Comparative Analysis: SimCLR versus Supervised ResNet

This study provides a clear head-to-head comparison, revealing the trade-offs between the two methods.

- **Representation Quality:** In low-label scenarios, the representations learned by SimCLR (with strong augmentation) are demonstrably of higher quality and more general-purpose than those learned by the supervised model. The features are more linearly separable, as evidenced by the high accuracy of the simple linear probe.
- **Computational Cost:** A significant trade-off is computational cost. SimCLR pre-training is expensive, requiring 100 epochs of training on the full dataset with a complex loss function and a large batch size. The supervised baseline, by contrast, was only trained for 100 epochs on a small subset of the data, which is computationally much cheaper. The practical decision thus involves a trade-off: invest significant upfront computational resources for pre-training to achieve high performance with few labels, or accept lower performance in exchange for a much faster and simpler training process on the small labelled set.
- **Flexibility:** The pre-trained SimCLR encoder is a flexible asset. The same encoder could potentially be used for various downstream tasks beyond classification, such

as object detection or segmentation, with minimal fine-tuning. A supervised model is, by its nature, specialised for the single task it was trained on.

5.5 Limitations of the Study

While this research provides clear insights, it is important to acknowledge its limitations, which also present avenues for future research.

- **Dataset Complexity:** The experiments were conducted solely on CIFAR-10, a dataset with small images and relatively distinct classes. The dynamics observed here may not directly transfer to more complex, high-resolution datasets like ImageNet, or to fine-grained classification tasks where subtle differences distinguish classes. On such datasets, the required complexity of the augmentation pipeline and the model architecture might be different.

- **Backbone Architecture:** Only the ResNet-18 architecture was used. While a standard choice, it is possible that different backbones, such as larger ResNets (e.g., ResNet-50) or more modern architectures like Vision Transformers (ViTs), could alter the comparative performance between the two paradigms.

- **Evaluation Protocol:** The evaluation of SimCLR was restricted to linear probing. This is a standard method for assessing representation quality, but it does not represent the upper bound of performance. Full fine-tuning of the pre-trained encoder on the downstream task could potentially yield even higher accuracies and might change the performance gap relative to the supervised model.

- **Hyperparameter Tuning:** The study used a fixed set of hyperparameters (e.g., learning rate, temperature). While these are common values, an exhaustive hyperparameter search for each individual experimental condition was not performed. It is possible that the supervised model, in particular, could achieve better performance with more carefully tuned hyperparameters for each specific low-data scenario.

In summary, this evaluation confirms the thesis that SimCLR, when paired with strong data augmentation, represents a paradigm shift in handling data-scarce computer vision problems. It transforms the challenge from a pure supervised learning problem into a two-stage process where unlabelled data can be effectively leveraged to build a powerful foundation for learning.

6 Conclusion and Future Work

6.1 Conclusion

This thesis set out to evaluate the effectiveness of the SimCLR self-supervised learning framework for image classification in limited-label environments, drawing a direct comparison with a traditional supervised learning baseline. Through a systematic set of experiments on the CIFAR-10 dataset, varying both the fraction of available labels and the strength of data augmentation, this research has yielded several conclusive insights.The primary conclusion of this work is that the performance of SimCLR is inextricably linked to the application of a strong and diverse data augmentation strategy. In the absence

of such augmentations, the contrastive learning objective fails to produce semantically meaningful representations, leading to performance that is substantially inferior to a standard supervised model. However, when a robust augmentation pipeline is employed, SimCLR becomes a remarkably powerful tool for representation learning.

In all tested low-label scenarios (1%, 5%, and 10% of labeled data), the SimCLR model pre-trained with strong augmentations and evaluated with a linear probe significantly outperformed the supervised ResNet-18 model trained from scratch on the same data. This demonstrates the exceptional label efficiency of the self-supervised approach. By leveraging the entire unlabeled training set, SimCLR can learn rich, general-purpose visual features that allow a simple classifier to achieve high accuracy with very few labeled examples. This finding strongly supports the use of self-supervised pre-training as a premier strategy for computer vision tasks where labeled data is a scarce and expensive resource.

Furthermore, this study highlights the fundamental conceptual difference in the role of data augmentation between the two paradigms. In supervised learning, it acts as a regularizer to improve generalization. In contrastive self-supervised learning, it is the core mechanism that defines the learning task and generates the supervisory signal itself. In conclusion, this research empirically validates the promise of self-supervised learning in data-constrained settings and quantitatively establishes the conditions required for its success. It provides a clear, evidence-based confirmation that for certain classes of problems, the implicit supervision derived from comparing augmented views of unlabeled data can be more powerful than the explicit supervision derived from a small set of labeled data.

6.2 Future Work

The findings and limitations of this study open up several promising directions for future research. The following avenues could build upon the work presented in this thesis:

1. **Exploration of Different Self-Supervised Algorithms:** This study focused exclusively on SimCLR. A valuable extension would be to conduct a similar comparative analysis with other families of SSL algorithms, such as clustering-based methods (e.g., SwAV), non-contrastive methods (e.g., BYOL, DINO), or masked image modeling approaches (e.g., MAE). This would provide a broader understanding of which SSL strategies are most effective in low-label regimes.

2. **Scaling to Larger and More Complex Datasets:** Replicating this study on more challenging datasets, such as ImageNet-1K or fine-grained classification datasets (e.g., CUB-200, Stanford Cars), would be a critical test of the generalizability of these findings. It would explore whether the same dynamics between augmentation, label fraction, and performance hold true for high-resolution images and more subtle inter-class distinctions.

3. **Analysis of Fine-Tuning versus Linear Probing:** This work evaluated SimCLR representations using only linear probing. A comprehensive follow-up study should compare this to full network fine-tuning on the downstream task. Investigating how much performance is gained by fine-tuning the entire encoder, and how this affects the performance gap with the supervised baseline, would provide a more complete picture of the practical utility of the pre-trained models.

4. **In-depth Hyperparameter Sensitivity Analysis:** A dedicated study on the sensitivity of both SimCLR and supervised models to key hyperparameters would be beneficial. This could include a detailed analysis of the effect of the temperature parameter (τ) in the NT-Xent loss, the batch size, the optimizer choice, and the learning rate schedule, particularly in the context of varying label fractions.

5. **Qualitative Analysis of Learned Representations:** To better understand why the SimCLR representations are so effective, future work could involve qualitative analysis. Techniques like t-SNE or UMAP could be used to visualize the feature space and assess how well the learned representations cluster the data by class, even before a classifier is trained. Furthermore, analyzing which parts of an image the model focuses on using saliency maps (e.g., Grad-CAM) could reveal what features the model deems important.

6. **Investigating the Composition of "Strong" Augmentation:** The strong augmentation pipeline used here was a composite of several transformations. Future work could systematically dissect this pipeline, performing an ablation study to determine the relative importance of each individual augmentation (e.g., cropping vs. color jitter vs. grayscale) for learning high-quality representations. This could lead to more optimized and task-specific augmentation strategies.

By pursuing these research directions, the community can continue to build upon the foundational understanding of self-supervised learning, further refining these powerful techniques and expanding their applicability to an even wider range of real-world problems.

References

Bardes, A., Ponce, J. and LeCun, Y. (2022). VICReg: Variance-Invariance-Covariance Regularization for Self-Supervised Learning, *International Conference on Learning Representations (ICLR)*.
URL: *https://arxiv.org/abs/2105.04906*

Bose, A. J., Athreya, A. P., Banerjee, P., Dey, D. and Roy-Chowdhury, A. K. (2020). Guided Contrastive Learning for Unsupervised Image Classification, *arXiv preprint arXiv:2104.02057* .
URL: *https://arxiv.org/abs/2104.02057*

Chen, T., Kornblith, S., Norouzi, M. and Hinton, G. (2020). A Simple Framework for Contrastive Learning of Visual Representations, *Proceedings of the International Conference on Machine Learning (ICML)*.
URL: *https://arxiv.org/abs/2002.05709*

Chen, X., Kornblith, S., Swersky, K., Norouzi, M. and Hinton, G. (2020). Big Self-Supervised Models are Strong Semi-Supervised Learners, *Advances in Neural Information Processing Systems (NeurIPS)*.
URL: *https://arxiv.org/abs/2006.10029*

Chuang, C.-Y., Robinson, J., Lin, Y.-C., Torralba, A. and Jegelka, S. (2020). Debiased Contrastive Learning, *Advances in Neural Information Processing Systems (NeurIPS)*.
URL: *https://arxiv.org/abs/2007.00224*

Ciortan, M. and Lioma, C. (2021). Self-Supervised Learning: Generative or Contrastive, *arXiv preprint arXiv:2006.08218* .
URL: *https://arxiv.org/abs/2006.08218*

Dwibedi, D., Aytar, Y., Tompson, J., Sermanet, P. and Zisserman, A. (2021). With a Little Help from My Friends: Nearest-Neighbor Contrastive Learning of Visual Representations, *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*.
URL: *https://arxiv.org/abs/2104.14548*

Goncharov, M. et al. (2023). vox2vec: A Framework for Self-supervised Contrastive Learning of Voxel-level Representations in Medical Images, *arXiv preprint arXiv:2307.14725* .
URL: *https://arxiv.org/abs/2307.14725*

Hadsell, R., Chopra, S. and LeCun, Y. (2006). Dimensionality Reduction by Learning an Invariant Mapping, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
URL: *https://ieeexplore.ieee.org/document/1640964*

He, K., Zhang, X., Ren, S. and Sun, J. (2016). Deep Residual Learning for Image Recognition, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Khosla, P. et al. (2020). Supervised Contrastive Learning, *Advances in Neural Information Processing Systems (NeurIPS)*.
URL: *https://arxiv.org/abs/2004.11362*

Shurrab, S. and Duwairi, R. (2021). Self-supervised Learning Methods and Applications in Medical Imaging Analysis: A Survey, *IEEE Access* **9**: 44707–44723.
URL: *https://arxiv.org/abs/2109.08685*

Tsang, Y. S., Xiao, T. and Xu, B. (2022). MICLe: Exploiting Medical Image Structure through Contrastive Learning, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*.
URL: *https://arxiv.org/abs/2106.05682*

Tschannen, M., Djolonga, J., Ritter, M., Mahendran, A., Houlsby, N., Gelly, S. and Lucic, M. (2020). On Mutual Information Maximization for Representation Learning, *International Conference on Learning Representations (ICLR)*.
URL: *https://arxiv.org/abs/1907.13625*

Wu, Z., Xiong, Y., Yu, S. X. and Lin, D. (2018). Unsupervised Feature Learning via Non-Parametric Instance Discrimination, *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.
URL: *https://arxiv.org/abs/1805.01978*

Yang, R. and Xing, X. (2023). SemiOccam: A Robust Semi Supervised Image Recognition Network Using Sparse Labels, *arXiv preprint arXiv:2506.03582* .
URL: *https://arxiv.org/abs/2506.03582*

Zbontar, J., Jing, L., Misra, I., LeCun, Y. and Deny, S. (2021). Barlow Twins: Self-Supervised Learning via Redundancy Reduction, *Proceedings of the International Conference on Machine Learning (ICML)*.
URL: *https://arxiv.org/abs/2103.03230*

Zhai, X., Kolesnikov, A., Houlsby, N. and Beyer, L. (2022). Scaling Vision Transformers, *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.
URL: *https://arxiv.org/abs/2106.04560*