Exploring Self-Supervised Learning Through SimCLR: Reproducing and Evaluating Natural Image Classification and Transfer Learning Accuracy

Bachelor Thesis

Bachelor of Science in Applied Computer Science

Sascha Alexander Wolf

March 11, 2025

Supervisor:

1st: Prof. Dr. Christian Ledig
2nd: Jonas Alle M. Sc.

Chair of Explainable Machine Learning
Faculty of Information Systems and Applied Computer Sciences
Otto-Friedrich-University Bamberg

Abstract

Self-supervised learning (SSL) is a powerful technique for training deep neural networks without the need for large labeled datasets. However, the effectiveness of SSL heavily relies on data augmentation strategies to generate diverse and informative training samples. Existing augmentation techniques often apply global transformations indiscriminately, ignoring the semantic structure of images, which can lead to suboptimal feature learning.

This thesis introduces a Segmentation-Based Augmentation Pipeline (SegAug-SimCLR), which enhances the standard SimCLR framework by incorporating semantic image segmentation into the augmentation process. Instead of applying transformations uniformly, our method differentiates between foreground and background regions, allowing for targeted augmentations that preserve important object features while still increasing variability in the training data.

SegAug-CimCLR consists of four major components: (1) DeepLabV3-based segmentation to generate foreground-background masks, (2) region-specific augmentations tailored for each part of the image, (3) integration into the SimCLR contrastive learning pipeline, and (4) evaluation on standard benchmarks.

Quantitative experiments demonstrate that segmentation-based augmentations lead to minor improvements in contrastive learning performance. On ImageNet (10% subset), the Top-1 accuracy improves from 34.6% to 35.2% when segmentation-based augmentations are applied. In transfer learning experiments across multiple datasets, segmentation-based augmentations show dataset-dependent benefits, with notable gains in Flowers (+1.0%).

Despite its effectiveness, challenges remain in improving segmentation quality and ensuring robustness across diverse datasets. Nonetheless, SegAug-SimCLR presents an approach for structure-aware augmentations in contrastive learning, enabling better feature learning while maintaining compatibility with existing self-supervised learning frameworks.

Abstract

Selbstüberwachtes Lernen (SSL) hat sich als leistungsstarke Methode für das Training von tiefen neuronalen Netzwerken ohne umfangreiche gelabelte Datensätze etabliert. Die Effektivität von SSL hängt jedoch stark von geeigneten Augmentations-Strategien ab, um vielfältige und informative Trainingsproben zu erzeugen. Bestehende Augmentierungstechniken wenden globale Transformationen oft wahllos an, ohne die semantische Struktur von Bildern zu berücksichtigen, was zu einer suboptimalen Merkmalserfassung führen kann.

Diese Arbeit stellt eine segmentierungsbasierte Augmentierungspipeline (SegAug-SimCLR) vor, die das SimCLR-Framework durch semantische Bildsegmentierung erweitert. Anstatt Transformationen einheitlich auf das gesamte Bild anzuwenden, unterscheidet unsere Methode zwischen Vordergrund und Hintergrund, wodurch gezielte Augmentierungen möglich werden. So bleiben relevante Objektmerkmale erhalten, während gleichzeitig die Datenvariabilität erhöht wird.

SegAug-SimCLR besteht aus vier Hauptkomponenten: (1) Segmentierung mittels DeepLabV3 zur Generierung von Vordergrund-Hintergrund-Masken, (2) bereichsspezifische Augmentierungen, die gezielt für verschiedene Bildregionen angewendet werden, (3) Integration in SimCLR sowie (4) Evaluation auf gängigen Benchmark-Datensätzen.

Quantitative Experimente zeigen, dass segmentierungsbasierte Augmentierungen zu leicht verbesserten Leistungen im kontrastiven Lernen führen. Auf ImageNet (10%-Subset) verbessert sich die Top-1-Genauigkeit von 34,6% auf 35,2%, wenn segmentierungsbasierte Augmentierungen eingesetzt werden. Bei Transfer-Learning-Experimenten über mehrere Datensätze zeigen sich datensatzabhängige Verbesserungen, insbesondere bei Flowers (+1,0%).

Trotz dieser Fortschritte bestehen weiterhin Herausforderungen, insbesondere bei der Verbesserung der Segmentierungsqualität und der Robustheit über verschiedene Domänen hinweg. Dennoch stellt SegAug-SimCLR eine Erweiterung für strukturbewusste Augmentierungen im kontrastiven Lernen dar, die eine präzisere Merkmalsextraktion ermöglicht, während sie mit bestehenden selbstüberwachten Lernverfahren kompatibel bleibt.

Acknowledgements

First, I would like to express my gratitude to my family for their support throughout my studies. Their encouragement and belief in me have been invaluable.

I am also sincerely grateful to Professor Ledig for giving me the opportunity to write my thesis at his chair and for inspiring me in this field through his seminar.

Last but not least, I would like to extend my thanks to my supervisor, Jonas Alle, whose guidance has been instrumental in the successful completion of this thesis.

Contents

List of Figures

List of Tables

List of Acronyms

BYOL	Bootstrap Your Own Latent
CNN	Convolutional Neural Network
CPC	Contrastive Predictive Coding
DrLIM	Dimensionality Reduction by Learning an Invariant Mapping
MLP	Multi-Layer Perceptron
MoCo	Momentum Contrast
SAM	Segment Anything Model
SGD	Stochastic Gradient Descent
SimCLR	A Simple Framework for contrastive learning
SSL	Self-Supervised Learning
SwAV	Swapping Assignments between Views

1 Introduction

Machine learning has seen remarkable advancements in recent years, particularly in tasks such as image classification, object detection, and image segmentation. A key factor behind this success has been supervised learning, where models are trained using large labeled datasets. However, collecting and labeling these datasets is expensive, requiring significant time and resources. The introduction of ImageNet by Deng et al. (2009) greatly accelerated progress by providing a large, diverse labeled dataset.

Despite these advancements, acquiring labeled data remains a challenge in certain fields, particularly medical imaging, where expert knowledge is essential for reliable annotations (Esteva et al., 2019). Self-supervised learning (SSL) is a promising alternative by leveraging unlabeled data to learn meaningful representations. SSL models are trained using tasks that encourage them to discover patterns without explicit labels, enabling their use in various downstream applications such as classification and object detection.

Among different SSL approaches, contrastive learning has gained widespread adoption. This method encourages the model to learn representations by bringing similar images closer together in the feature space while pushing dissimilar images apart. A major milestone in contrastive learning was the introduction of SimCLR—A Simple Framework for Contrastive Learning of Visual Representations (Chen et al., 2020a). SimCLR demonstrated strong performance across multiple self-supervised benchmarks while remaining simpler than previous methods. Unlike earlier approaches, it does not require specialized architectures or memory banks (Chen et al., 2020a). Instead, it relies on random augmentations to generate positive pairs, which are then optimized using a contrastive loss function. By maximizing agreement between these augmented pairs, SimCLR learns meaningful representations that are invariant to various transformations. The success of SimCLR has inspired subsequent methods such as Bootstrap Your Own Latent (BYOL) (Grill et al., 2020) and Swapping Assignments between Views (SwAV) (Caron et al., 2021).

Despite its achievements, SimCLR has certain limitations. The framework applies augmentations uniformly across the entire image, disregarding its internal structure. This can lead to a loss of crucial features, particularly in tasks where fine details are critical. For example, aggressive cropping or blurring may obscure important patterns or remove key object features. Recent research, such as Local Augment (LA) (Kim et al., 2021), suggests that applying transformations selectively to different regions can better preserve essential details and improve the robustness of learned representations. LA leverages the local bias of convolutional neural networks (CNNs) by introducing region-specific transformations, such as noise, brightness changes, or localized cropping, to enhance feature learning.

Building on this idea, this thesis investigates segmentation-based augmentations as a means of improving SimCLR. Modern semantic segmentation models, such as Segment Anything (SAM) (Kirillov et al., 2023) and DeepLabv3 (Chen et al., 2017), can partition an image into semantically meaningful regions, distinguishing

objects from their background. Using these segmentation maps, augmentations can be selectively applied to different parts of an image. For instance, color distortions might be applied only to the background while preserving the main object, ensuring that key semantic information remains intact.

This segmentation-based approach introduces controlled variability in data while maintaining crucial structural details, potentially leading to better learned representations and improved performance on downstream tasks.

1.1 Motivation

SSL is a strong alternative to traditional supervised learning by reducing the dependency on labeled data, which is often costly and labor-intensive to obtain (Goodfellow et al., 2016). In many real-world applications, such as medical imaging and autonomous driving, acquiring high-quality annotations requires significant domain expertise, making large-scale labeled datasets impractical (Esteva et al., 2019; Geiger et al., 2012). Contrastive learning, a key paradigm within SSL, has demonstrated remarkable success in learning meaningful representations without manual supervision (Chen et al., 2020a).

However, despite these developments, existing contrastive learning methods such as SimCLR still depend on global data augmentations, which treat the entire image identically and might completely ignore semantic structure information about objects within a scene (Chen et al., 2020a). This limitation becomes particularly critical in cases where learning robust representations depends on preserving meaningful semantic structure (Tian et al., 2020) . Motivated by this gap, this work investigates the usage of segmentation-based augmentations, with a view to improving feature learning through spatially aware transformations.

By incorporating segmentation techniques into the contrastive learning pipeline, we will develop a method that retains object structures while maintaining the benefits of standard contrastive learning. The reproduction of key results from SimCLR, introducing segmentation-driven augmentations, and evaluation of the same on downstream tasks are covered in this thesis to contribute toward the larger field of self-supervised representation learning.

1.2 Contribution

This bachelor thesis makes following contributions in the field of contrastive learning:

1. **Reproducing SimCLR results:** The SimCLR framework is implemented in PyTorch and the code has been made open source. Additionally, key experiments from the original paper Chen et al. (2020a) are reproduced. This includes experiments from Table 6, 7, and 8, which evaluate the performance of the base encoder on downstream tasks like linear classification. These results serve as a baseline for evaluating the proposed segmentation-based augmentations.

2. **Developing segmentation-based augmentations:** A novel augmentation method is introduced using segmentation maps generated from DeepLabV3 (Chen et al., 2017). This approach allows for region-specific augmentations, differentiating between foreground and background transformations, unlike traditional global augmentations.

3. **Comparative analysis of segmentation-based augmentations:** The proposed segmentation-based augmentations are compared with standard SimCLR augmentations in linear classification tasks. The evaluation assesses whether foreground-background separation improves feature learning and how it impacts downstream performance.

These contributions aim to improve contrastive learning by integrating segmentation-based augmentations.

2 Related Work

The success of SSL has been driven by the development of powerful frameworks and techniques, including contrastive learning and advanced data augmentations. This section reviews the key concepts relevant for this thesis, focusing on the evolution of contrastive learning frameworks, the role of data augmentation in SSL and segmentation-based augmentations.

Contrastive learning is an important method in self-supervised learning, which enables models to learn meaningful representations without the need for labeled data. The central idea is to maximize the agreement between positive pairs while minimizing the agreement between negative pairs.

The foundation of contrastive loss was laid by Hadsell et al. (2006), where contrastive loss was introduced to learn mappings that are invariant to input transformations. They minimized the distance between similar samples while maximizing the distance between dissimilar ones. A key factor in Dimensionality Reduction by Learning an Invariant Mapping (DrLIM) was the use of different loss functions for similar and dissimilar pairs. Unlike conventional learning systems, where the loss is computed as a sum over individual samples, DrLIM applies its loss function over pairs of samples, categorizing them as similar or dissimilar. Over time, researchers introduced more advanced contrastive loss functions. Triplet Loss (Schroff et al., 2015), which was introduced for FaceNet, minimizes the distance between an anchor and a positive of the same identity and maximizes the distance between the anchor and a negative of a different identity. Lifted Structured Loss (Song et al., 2015) improved computational efficiency by considering all pairs within a batch. InfoNCE, used in Contrastive Predicitve Coding (CPC) (van den Oord et al., 2019), extends Noise Contrastive Estimation (NCE) (Gutmann and Hyvärinen, 2010) to maximize mutual information between representations.

While early methods were limited in scalability, modern frameworks for contrastive learning have arisen. One of the first was introduced by van den Oord et al. (2019), called CPC. CPC learns representations by predicting future observations in latent space by using powerful autoregressive models. It demonstrated the effectiveness of contrastive learning in extracting useful representations from high-dimensional data. Momementum Contrast (MoCo) (He et al., 2020) trains a visual representation encoder by matching an encoder query to a dictionary of encoded keys using contrastive loss. SimCLR (Chen et al., 2020a) introduced a simple yet powerful framework for contrastive learning. SimCLR emphasizes the importance of strong data augmentations in SSL and the use of a nonlinear transformation to improve the quality of learned representations. It demonstrated that with sufficient data augmentation and large batch sizes, contrastive learning could achieve performance comparable to supervised learning. MoCo v2 (Chen et al., 2020b) is an improved version of MoCo. By combining findings from the SimCLR paper, namely using a mulit-layer perceptron (MLP) projection head and more data augmentation, they could outperform SimCLR, while not requiring large batch sizes. BYOL (Grill et al., 2020) introduces a new approach to self-supervised image representation learning. BYOL eliminates the need for negative pairs, by using two neural networks, an online and a target

network, that interacts and learn from each other. For each augmented view of an image, the online network is trained to predict the target network's representation of the same image under a different augmentation. Simultaneously, the target network is updated using an exponential moving average of the online network's parameters, ensuring stable and consistent learning. SwAV (Caron et al., 2021) introduces a clustering-based approach to self-supervised learning, which eliminated the need for pairwise comparison. Instead of directly contrasting representations, SwAV simultaneously clusters data and enforces consistency between cluster assignments across different augmentations of the same image. This is achieved using a "swapped" prediction mechanism, where the model predicts the assignment of one view using the representation of another. Also multi-crop augmentation strategy is introduced, which generates two full-resolution crops along with multiple smaller, low-resolution crops. This increases the number of views of an image with no computational or memory overhead, while improving ImageNet (Deng et al., 2009) linear evaluation accuracy between 2% and 4%. Barlow Twins (Zbontar et al., 2021) learns useful representations by ensuring that different versions of the same image have similar embeddings while reducing redundancy between feature components. It does this by aligning embeddings along the diagonal of a cross-correlation matrix and minimizing off-diagonal values. Unlike other methods, Barlow Twins does not require negative samples, large batches, or special network designs.

Data augmentation is a fundamental part in contrastive learning by creating diverse yet semantically meaningful transformations of input images. Chen et al. (2020a) demonstrated that random cropping and color distortion are essential for learning high-quality representations in SimCLR. Tian et al. (2020) further explored the principles of effective augmentations and introduced the InfoMin principle, which states that the best views for contrastive learning should retain task-relevant information while minimizing redundant details. Their experiments showed that stronger augmentations lead to better accuracy on downstream tasks.

Beyond standard augmentations, learned augmentation strategies like AutoAugment (Cubuk et al., 2019a) and RandAugment (Cubuk et al., 2019b) have been proposed to optimize augmentation policies. However, these methods apply transformations globally and do not take the semantic structure of an image into account.

To address this limitation, segmentation-based augmentations introduce semantic awareness into SSL by selectively modifying images based on their content rather than applying global transformations. ClassMix (Olsson et al., 2020) introduces an approach that blends regions from different images using segmentation masks, enhancing sample diversity. CutMix (Yun et al., 2019) follows a similar approach by replacing patches of an image with regions from another, forcing models to learn robust representations. Kim et al. (2021) have introduced Local Augment (LA), a novel augmentation strategy that utilizes local bias properties of CNNs. Instead of applying global transformations LA selects patches within an image and applies different augmentations to each patch. This creates locally diversified examples, which helps the network learn in a more generalized way. LA outperforms previous methods on ImageNet and STL10 (Coates et al., 2011) and has competitive performance on CIFAR100 (Krizhevsky, 2009).

3 Background

3.1 Methods of Deep Learning

Deep learning, a subset of machine learning, has revolutionized numerous fields, including computer vision, natural language processing, and autonomous systems. It leverages artificial neural networks with multiple layers to learn hierarchical representations of data, enabling models to perform complex tasks such as image recognition, speech processing, and decision-making with remarkable accuracy (LeCun et al., 2015). The following subsections outline the primary learning paradigms in deep learning: supervised learning, unsupervised learning, semi-supervised learning, and self-supervised learning, with a particular focus on contrastive learning, a key component of this thesis.

3.1.1 Supervised Learning

Supervised learning (SL) is the most important methodology in machine learning (Cunningham et al., 2008). The objective is to approximate a function $f : X \to Y$ by minimizing a loss function $L(y, \hat{y})$, where y represents the ground truth labels and $\hat{y}$ represents the model's predicted output. Often, y may be difficult to collect automatically and must be provided by a human supervisor (Goodfellow et al., 2016). SL requires large-scale labeled data, which can be expensive and time-consuming to obtain.

Prominent architectures for SL include Convolutional Neural Networks (Krizhevsky et al., 2012) and Transformers (Vaswani et al., 2023), which are widely used in computer vision and natural language processing tasks. Despite its success, supervised learning is highly dependent on labeled data, limiting its applicability in domains where annotations are scarce.

3.1.2 Unsupervised Learning

Unsupervised learning enables models to identify patterns and structures in data without relying on labeled examples. Instead of learning explicit input-output mappings, it seeks to discover underlying relationships (Goodfellow et al., 2016). No human intervention is needed. This approach is fundamental for tasks such as clustering, dimensionality reduction and association rule learning. Despite its advantages, interpreting the learned representations and ensuring their relevance on downstream tasks remain challenging.

3.1.3 Semi-Supervised Learning

Semi-supervised learning bridges the gap between supervised learning and unsupervised learning by using both labeled and unlabeled data (Chapelle et al., 2009).

In the standard setting of semi-supervised learning, the dataset $X = (x_i)_{i\in[n]}$ is divided into two subsets: the labeled subset $X_l = (x_1, \ldots, x_l)$ with corresponding labels $Y_l = (y_1, \ldots, y_l)$, and the unlabeled subset $X_u = (x_{l+1}, \ldots, x_{l+u})$, where the labels are unknown.

3.1.4 Self-Supervised Learning

SSL is a a powerful alternative to traditional supervised learning, enabling models to learn from vast amounts of unlabeled data while learning useful feature representations without human annotation (Jing and Tian, 2019). Unlike fully unsupervised learning, SSL introduces structured pretext tasks to generate pseudo-labels, allowing models to learn representations that transfer effectively to multiple downstream tasks (Chen et al., 2020a). One of SSL's greatest advantages is its broad applicability across domains, like in medicine or biology, where labeled data is expensive and scarce, and the specific task is not always known in advance (Krishnan et al., 2022). By learning general-purpose feature representations, SSL enables models to adapt flexibly to new tasks without retraining from scratch, making it particularly useful in low-data scenarios.
Formally, SSL optimizes a loss function based on automatically generated pseudo-labels:

$$\mathcal{L}(D) = \min_{\theta} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(X_i, P_i), \tag{1}$$

where P_i represents pseudo-labels derived from self-supervised pretext tasks. This framework allows SSL models to surpass fully supervised methods in some tasks, particularly when fine-tuned on small amounts of labeled data (Chen et al., 2020a).

3.1.5 Contrastive Learning

Contrastive learning (CL) is a dominant approach in SSL. CL aims to train a model to generate similar representations for different augmented versions of the same input while distinguishing them from representations of other inputs (Balestriero et al., 2023). These augmented views are derived through data transformations, such as cropping, color distortion, or geometric manipulations. Contrastive learning aims to learn representations by maximizing the similarity between positive pairs while minimizing the similarity between negative pairs. Formally, given a reference sample x, a positive sample x^+ (a different augmented view of the same sample), and a negative sample x^- (a different instance), contrastive learning enforces the following constraint:

$$\text{score}(f(x), f(x^+)) \gg \text{score}(f(x), f(x^-)) \tag{2}$$

where:

- $f(x)$ represents the encoded feature representation of sample x.

The goal is to ensure high similarity between positive pairs (x, x^+) and low similarity between negative pairs (x, x^-) (Fei-Fei Li, 2022). This principle forms the foundation of self-supervised contrastive learning, where models learn to distinguish between augmented views of the same instance and other samples in the dataset.

3.2 Transfer Learning

Transfer learning is a machine learning paradigm where a model trained on one task is repurposed for a different but related task. This technique is particularly beneficial when labeled data for the target task is scarce, as it allows the reuse of knowledge from a source domain, improving generalization and reducing computational costs (Pan and Yang, 2010). Transfer learning can be mathematically formulated as follows:

Given a source domain D_S with a corresponding task T_S and a target domain D_T with a task T_T, transfer learning aims to improve the learning of the target predictive function f_T in D_T using knowledge from D_S and T_S, where $D_S \neq D_T$ or $T_S \neq T_T$.

This framework allows adaptation even when distributions $P(D_S)$ and $P(D_T)$ differ, which leads to the concept of domain adaptation (Pan and Yang, 2010).

Domain Adaptation Domain adaptation is a specialized case of transfer learning where the source and target domains are similar but have different distributions. Formally, given a source domain D_S with probability distribution $P(X_S)$ and a target domain D_T with $P(X_T)$, domain adaptation assumes that $P(X_S) \neq P(X_T)$ while still sharing commonalities in feature space.

3.3 Image Segmentation

Image segmentation is a fundamental task in computer vision that involves dividing an image into meaningful regions to facilitate object recognition, analysis, and processing (Minaee et al., 2020). The goal is to simplify the representation of an image into a form that is more meaningful and easier to analyze. Image segmentation serves as a critical pre-processing step for many computer vision tasks. By segmenting an image into its constituent parts, it becomes possible to focus on specific objects or regions of interest. It is widely used in various applications such as medical imaging (Ronneberger et al., 2015), autonomous driving (Cordts et al., 2016) or video surveillance (Reenu and Saleem Durai, 2019).

There are several types of image segmentation, each suited to different tasks and objectives:

Semantic Segmentation Assigns a label to each pixel in an image, grouping similar objects together. Unlike other segmentation methods, semantic segmentation does not distinguish between instances of the same class. Instead, it aims to understand the high-level structure of the scene and identify the objects present in the image.

Instance Segmentation Extends semantic segmentation by distinguishing individual objects of the same class. This is crucial in scenarios where multiple objects of the same type must be identified separately.

Panoptic Segmentation A comprehensive approach that combines semantic and instance segmentation. It provides a unified view where each pixel is classified into either a specific object instance or a background region. This method is beneficial for complex scene understanding in robotics and automated systems (Kirillov et al., 2019).

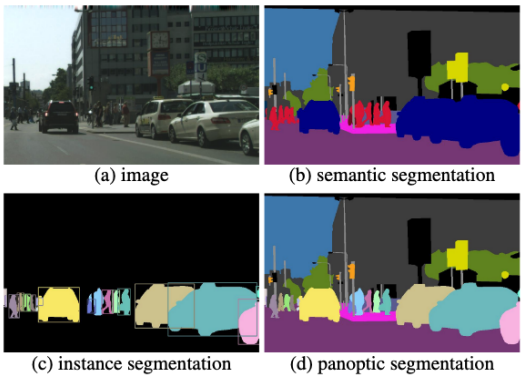

Figure 1: Visualization of different segmentation methods. (a) shows the ground truth image. Reproduced from Kirillov et al. (2019).

3.4 The SimCLR Framework

Self-supervised learning provides a method to train deep neural networks without requiring large amounts of labeled data. SimCLR simplifies recently proposed contrastive self-supervised learning algorithms and introduces a framework for contrastive learning of visual representations. Unlike traditional supervised learning, where models are trained with explicit labels, SimCLR constructs a pretext task that enables the model to learn meaningful representations directly from raw image data.

This is achieved through contrastive learning, where the model is trained to maximize the similarity between different augmented versions of the same image (positive pairs) while minimizing the similarity between representations of different images (negative pairs). By applying this contrastive objective, SimCLR learns robust and transferable representations without requiring labeled supervision.

The success of SimCLR does not stem from entirely novel components but rather from their specific combination, which significantly improves performance over previous unsupervised baselines (Chen et al., 2020a). The following section provides a detailed explanation of these components.

3.4.1 Data Augmentation

One of the key findings in SimCLR is that strong data augmentation plays a crucial role in the success of the framework (Chen et al., 2020a). Unlike supervised learning, where augmentations are often mild to avoid altering the semantic meaning of labeled data, SimCLR relies on strong augmentations to generate meaningful transformations of images. These augmentations create diverse views of the same image while preserving essential content, enabling the model to learn invariant representations that generalize well to unseen data.

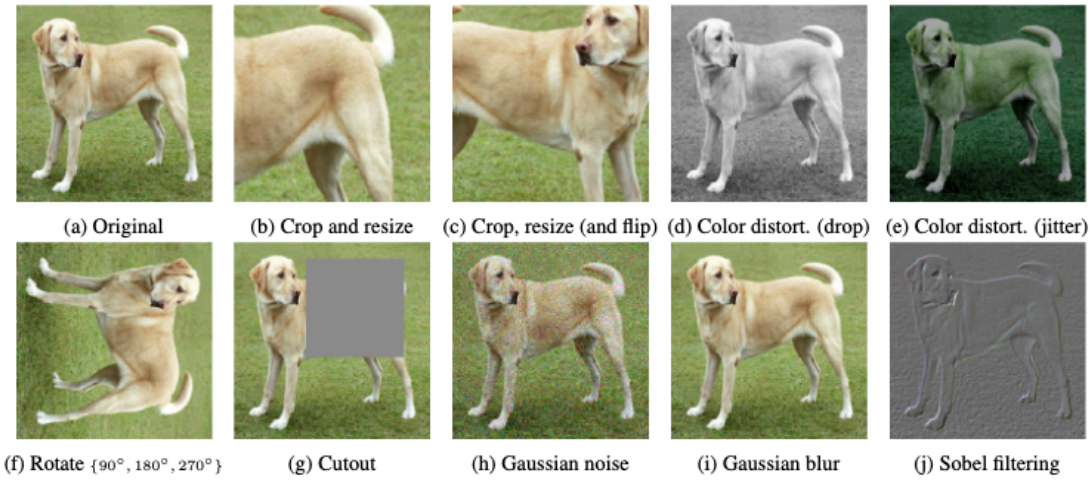

Figure 2: Illustrations of the studied data augmentations. Reproduced from Chen et al. (2020a).

As shown in Figure 2, the studied augmentations can be grouped into the following categories:

- **Spatial/Geometric Transformations:** Random cropping and resizing (with flipping), rotation, and cutout.
- **Appearance Transformations:** Color distortion (modifying brightness, contrast, and hue), Gaussian blur, and Sobel filtering.

Augmentation Pipeline In SimCLR, each input image undergoes two independent sets of augmentations, resulting in two transformed views of the same image. These views serve as positive pairs for contrastive learning, while views from different images in the batch act as negative pairs. The process is visualized in Figure 3. The objective is to maximize agreement between positive pairs while ensuring separability from negative pairs.

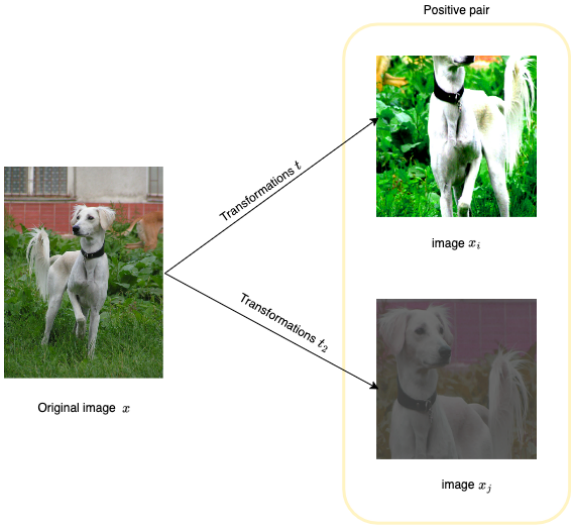

Figure 3: Visualization of a positive pair.

The table in Figure 4 presents the results of linear evaluation for Top-1 accuracy on the ImageNet dataset when various data augmentations are applied either individually or in combination. Notably, the combination of cropping and color jittering achieves the highest accuracy of 56.3%. Overall, stronger augmentations tend to yield better performance than lighter ones, demonstrating the importance of data transformations in contrastive learning.

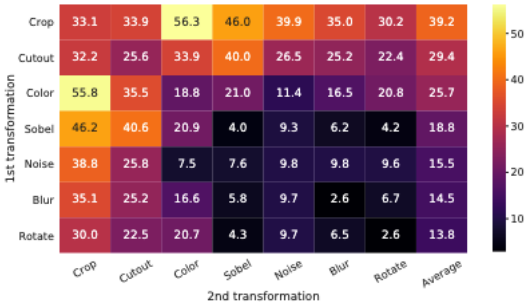

Figure 4: Linear evaluation (ImageNet Top-1 accuracy) under individual or combined augmentations. Diagonal entries correspond to single transformations, while off-diagonal entries represent compositions of two augmentations. Reproduced from Chen et al. (2020a).

3.4.2 Base Encoder

The base encoder is responsible for extracting feature representations from input images. In the original SimCLR study, a ResNet architecture was chosen as the base encoder due to its proven effectiveness in image representation tasks. Therefore, ResNet-50, a deep convolutional neural network with 50 layers, was employed. Given an input image x, the base encoder transforms it into a feature representation h in a high-dimensional space as follows:

$$h = f(x) \in R^d$$

where d represents the dimensionality of the feature space, and h is the output before the average pooling layer. For ResNet-50, this results in a 2048-dimensional feature vector. These features are not directly optimized for the contrastive learning task. Instead, they serve as intermediate representations that are further processed by the projection head.

Variants In addition to the standard ResNet-50 architecture, two extended variants were investigated as base encoders in the SimCLR study:

- ResNet-50: The original ResNet architecture, where channel sizes follow the default design.
- ResNet-50 2x: A variant of ResNet where the channel dimensions are doubled across the network.
- ResNet-50 4x: A further extension where channel dimensions are quadrupled.

Zagoruyko and Komodakis (2017) demonstrated the effectiveness of wide residual networks. This claim is supported by multiple experimental results presented in this study. The experiments revealed that ResNet-4x outperformed both the standard ResNet and ResNet-2x on downstream performance, as detailed in Section 5.

Backbone The base encoder serves as the backbone of the model for downstream tasks. After pretraining, the projection head is discarded, and a linear classifier is placed on top of the base encoder. This classifier is trained to evaluate the quality of the learned representations using linear evaluation. During the training of the linear classifier, the base encoder can be fine-tuned.

3.4.3 Projection Head

The projection head improves the performance of the contrastive learning task by mapping the high-dimensional feature representations extracted by the base encoder into a lower-dimensional latent space, which is more effective for computing the contrastive loss.

Architecture The projection head in SimCLR is implemented as a MLP with two linear layers. A batch normalization layer and a ReLU activation function are applied to ensure stable training and introduce non-linearity. The second linear layer reduces the hidden representation to a lower-dimensional space, which is then used for calculating the contrastive loss.

The final output z of the projection head is computed as follows:

$$z = g(h) = W * \text{ReLU}(W * h)$$

Performance As shown in Figure 5, models with a projection head outperform those without it in a linear evaluation benchmark. The x-axis represents the dimensionality of the projection head's output, while the y-axis shows the corresponding Top-1 accuracy. The non-linear projection head consistently outperforms the linear projection head across all dimensionalities.

Increasing the output dimension beyond 128 does not yield further performance improvements. When no projection head is used, Top-1 accuracy is only measured for an output dimension of 2048, and it performs significantly worse than with a projection head. This highlights the importance of the projection head in contrastive learning.

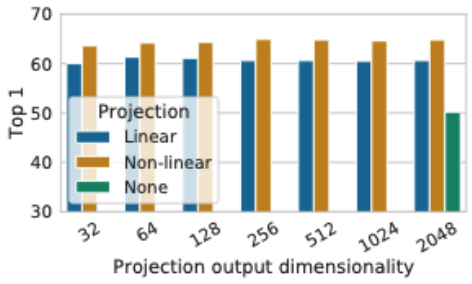

Figure 5: Linear evaluation of representations with different projection heads $g()$ and various dimensions of $z = g(h)$. Reproduced from Chen et al. (2020a).

3.4.4 Loss Function

Contrastive loss encourages the model to produce similar representations for augmented views of the same image while pushing apart representations of different images. This is achieved by maximizing the similarity of positive pairs while minimizing the similarity of negative pairs. SimCLR uses a specific variant of contrastive loss called NT-Xent:

Given a batch of N images, two augmented views are generated for each image, resulting in $2N$ total samples. For a pair of positive samples (i, j), the NT-Xent loss is defined as:

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

where:

- z_i and z_j: The output representations from the projection head for the two augmented views of the same image.
- $\text{sim}(z_i, z_j)$: Cosine similarity between z_i and z_j, defined as:

$$\text{sim}(z_i, z_j) = \frac{z_i \cdot z_j}{\|z_i\| \|z_j\|}$$

- τ: Temperature parameter.
- $\mathbf{1}_{[k \neq i]}$: An indicator function ensuring that z_i does not compare to itself.

4 Methods

This section outlines the methodologies employed in this thesis. We begin by detailing the standard SimCLR pipeline, followed by our proposed segmentation-based augmentation approach. The goal is to enhance contrastive learning by incorporating region-specific augmentations, leveraging segmentation techniques to improve feature extraction and representation learning.

4.1 Standard SimCLR Pipeline

Building upon the SimCLR framework introduced in 3.4, this section focuses on details.

Figure 6 illustrates the standard SimCLR training pipeline. Our implementation follows the original framework, incorporating the following components:

- Base Encoder: A ResNet-50 architecture is used as the feature extraction network.
- Projection Head: A two-layer MLP with ReLU activation maps the encoded features to a 128-dimensional latent space, where contrastive loss is applied.
- Contrastive Loss: NT-Xent loss with a temperature scaling factor of 0.1.

Data Augmentation Strategy We adopt the same augmentation strategy proposed in the SimCLR study. The strategy consists of:

- Random Resize Crop: Crops a random area of the image and resizes it to 224×224 pixels, with p=1.0.
- Horizontal Flip: Applied with p=0.5.
- Color Jittering: Alters brightness, contrast, saturation, and hue by uniformly sampling adjustment factors, with p=0.8.
- Grayscale: Applied with p=0.2.
- Gaussian Blur: Applied with a kernel size of 10% of the image height/width, with p=0.5.

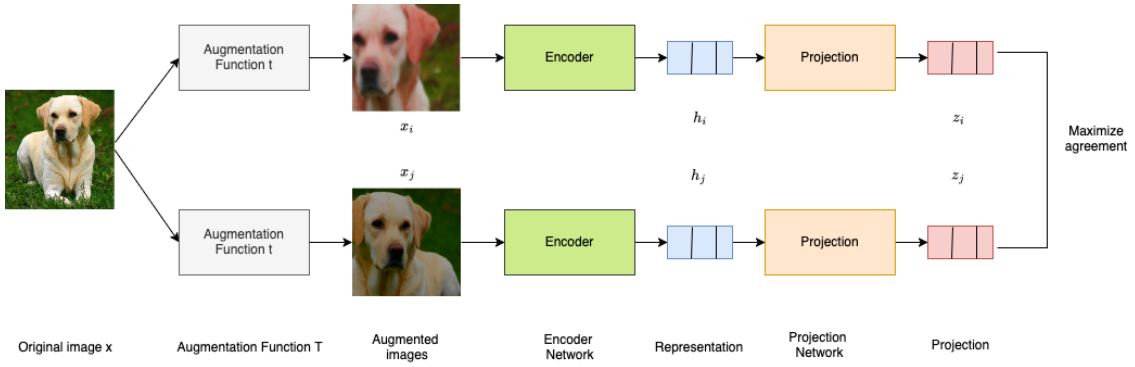

Figure 6: SimCLR original training pipeline.

4.2 Segmentation-based Augmentation Pipeline

In this section, we introduce our segmentation-based augmentation strategy for SimCLR, expanding on the standard pipeline described in Section 4.1. The main goal is to preserve critical object features by applying augmentations selectively to different regions in the image, rather than treating the entire image uniformly. This approach allows us to incorporate stronger data transformations while reducing the risk of distorting object-relevant information.

This chapter is structured as follows: Section 4.2.1 provides the motivation for using segmentation-based approaches in contrastive learning. Section 4.2.2 outlines our chosen segmentation model (DeepLabV3). Section 4.2.3 explains how region-specific augmentations are selected and applied. Finally, Section 4.2.4 describes how these augmentations are integrated into the SimCLR framework, forming our complete segmentation-based SimCLR pipeline.

4.2.1 Motivation

Self-supervised learning frameworks such as SimCLR rely on data augmentations to generate different views of the same image, thus enforcing similarity constraints between representations. Traditional augmentation pipelines apply transformations such as random cropping, color jittering, and Gaussian blur uniformly across the entire image. However, this approach does not account for the foreground-background structure of natural images.

To address this, we introduce segmentation-based augmentations, which separately process the foreground and background regions. This method preserves critical object information by applying foreground-specific augmentations, while background-specific transformations help mitigate overfitting to non-informative structures.

4.2.2 Segmentation Method

DeepLabV3 (Chen et al., 2017) is used as the primary segmentation model for generating foreground-background masks. DeepLabV3 is a state-of-the-art semantic segmentation network that employs atrous convolution to balance detailed boundary detection with global context capture. It has demonstrated strong performance across various benchmarks. In Figure 7 segmentation results can be seen, where the main object is captured successfully. In the experiments, we use pre-trained DeepLabV3 models, which have been trained on large-scale datasets such as MS COCO (Lin et al., 2015) or Cityscapes (Cordts et al., 2016). These models offer broad applicability across various object classes, making them suitable for general segmentation tasks. However, for highly specialized domains like biomedical imaging, fine-tuning the model on a domain-specific dataset or training it from scratch to achieve better performance is recommended (Ronneberger et al., 2015). DeepLabV3 outputs a probability map over semantic classes. For simplicity, we reduce this to a binary segmentation mask (foreground vs. background). For each pixel, we pick the class of highest probability.

Figure 7: Visualizations of Deeplabv3 segmentations. Reproduced from Chen et al. (2017).

4.2.3 Region-specific Augmentations

Once we obtain a binary mask distinguishing the foreground from the background, we apply region-specific augmentations. Each region, foreground and background, can receive different transformations. For example, we can apply subtle transformations to the foreground to preserve details while using stronger modifications on the background to enhance robustness. By customizing these augmentations per region, we control how much distortion is introduced in crucial areas, such as an object's boundary. This stands in contrast to purely global augmentations where a strong blur might degrade or even remove an object's defining features.

As shown in Figure 8, segmentation-based augmentations help enhance feature learning by applying different transformations to the foreground and background. In the figure, (a) presents the original image, while (b) displays the segmentation mask, where the foreground is highlighted in red. Finally, (c) illustrates the augmented versions of the image: in the top example, the background is converted to grayscale,

Figure 8: Visualizations of segmentation based augmentations

while in the bottom example, Gaussian blur is applied to the background. In both cases, Color Jitter is used to modify the foreground, introducing subtle variations while preserving important details.

4.2.4 Segmentation-Based Augmentation SimCLR Pipeline

The following steps, as shown in Figure 9, are integrated into the SimCLR pipeline with region-specific augmentations defined:

1. **Segmentation Step**
   - The input image x is processed by DeepLabV3.
   - We obtain a binary mask $m \in \{0,1\}^{H \times W}$, where $m_{ij} = 1$ indicates that pixel (i, j) belongs to the foreground, while $m_{ij} = 0$ represents the background.
2. **Applying Region-Specific Augmentations**
   - The image is segmented into foreground and background based on the mask m.
   - Foreground: A predefined set of transformations is applied to the foreground.
   - Background: A separate set of transformations is applied to the background.
3. **Recombining Augmented Regions**
   - The augmented foreground and background are recombined into a single image, x_{segaug}.

4. **Generating Augmented Views**

   - Similar to the standard SimCLR approach, we generate two augmented views by applying random cropping to x_{segaug}. Every view is from a different version of x_{segaug}.

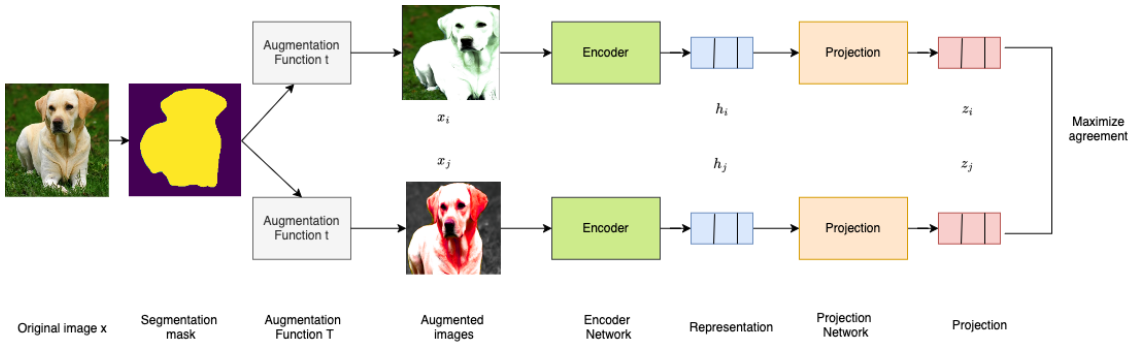

Figure 9: SimCLR training pipeline with segmentation-based augmentations.

Handling Poor Segmentation While DeepLabV3 performs well in many cases, certain images may yield poor segmentation quality:

- If the foreground region is too small (e.g., $< 5\%$ of total pixels), the model receives insufficient object information, leading to unstable feature representations and weak generalization.
- If the foreground region is too large (e.g., $> 90\%$ of total pixels), the segmentation fails to differentiate between the object and its surroundings, effectively classifying the entire image as foreground.

Figures 10, 11, and 12 illustrate these cases. In Figure 10, we present an example of a well-segmented image where the foreground is correctly identified and sufficiently large to contribute to effective feature learning. However, in Figure 11, the segmentation process incorrectly classifies nearly the entire image as foreground, making it difficult for the model to learn meaningful distinctions. Similarly, Figure 12 shows a case where the entire image is classified as background, effectively removing the object of interest from the training process.

To quantify segmentation quality, we define the foreground coverage metric as follows:

$$F_{\text{ratio}} = \frac{\sum m_{ij}}{H \times W} \tag{3}$$

where m_{ij} represents the binary segmentation mask at pixel location (i, j), and $H \times W$ is the total number of pixels in the image. This metric computes the proportion of the image classified as foreground.

To maintain training stability, we set thresholds on the foreground ratio. If $F_{\text{ratio}} < 0.05$ (foreground too small) or $F_{\text{ratio}} > 0.90$ (foreground too large), we discard the segmentation results and instead apply the standard SimCLR augmentation pipeline. This fallback strategy ensures training stability by preventing the model from learning unreliable masks in cases of poor segmentation quality.

By implementing this adaptive approach, we ensure that segmentation-based augmentations are only applied when they provide meaningful benefits, while defaulting to conventional augmentations in cases of segmentation failure.

Figure 10: Visualization of a well-segmented result.

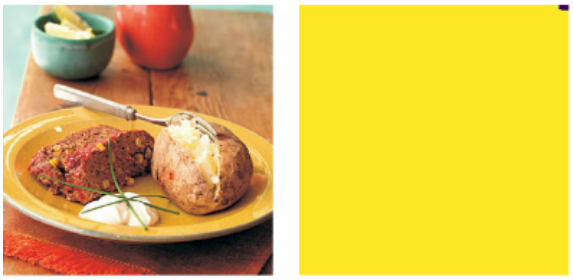

Figure 11: Visualization of a poor segmentation result (all foreground).

Figure 12: Visualization of a poor segmentation result (all background).

5 Experiments and Results

The primary objective of this section is to reproduce key experimental results from the original SimCLR paper to validate the reproducibility of the framework and establish a reliable baseline. This ensures that our implementation functions correctly and aligns with prior findings, establishing a strong foundation for further experiments.

We begin by replicating Figure B7 and Tables 6, 7, and 8 from the original SimCLR study.

- Figure B7: Validating the correct functionality of the pipeline.
- Table 6: Linear evaluation of the base encoder to assess representation quality.
- Table 7: Fine-tuning the base encoder.
- Table 8: Transfer learning accuracy to evaluate how well learned representations generalize to new tasks.

After reproducing the standard SimCLR results, we extend our analysis by applying segmentation-based augmentations and re-evaluating Table 6 to measure their impact on contrastive learning performance. Additionally, we compare the segmentation-based base encoder with the standard encoder on transfer learning tasks.

The following sections provide a detailed breakdown of each experiment and its corresponding results.

5.1 Datasets

The primary datasets used in our experiments are ImageNet and CIFAR-10 (Krizhevsky, 2009), both widely used benchmarks for evaluating deep learning models in image classification and representation learning. These datasets provide a diverse set of natural images and serve as strong benchmarks for self-supervised learning approaches like SimCLR.

ImageNet consists of 1.28 million training images and 50,000 validation images, spanning 1000 object categories. The dataset provides a diverse set of high-resolution natural images, making it an ideal benchmark for evaluating large-scale self-supervised learning frameworks like SimCLR. For the experiments, we made a time-driven adjustment to the ImageNet training set: instead of using the full training dataset, we sampled 10% of the training images from each class. This reduction was necessary to keep the training time within feasible limits (with one training iteration taking approximately 1.5 days for 10% of training data). Importantly, the full ImageNet validation set was used to ensure a reliable evaluation of the model's performance.

CIFAR-10, on the other hand, is a smaller dataset consisting of 60,000 images across 10 object categories, with 50,000 training samples and 10,000 validation samples. Each image has a resolution of 32×32 pixels, making CIFAR-10 suitable for testing model performance on lower-resolution datasets and evaluating generalization across different image scales.

For transfer learning CIFAR-100 (Krizhevsky, 2009), Oxford 102 Flowers (Nilsback and Zisserman, 2008), Oxford-IIIT Pets (Parkhi et al., 2012), SUN397 (Xiao et al., 2010) and Caltech-101 (Fei-Fei et al., 2004) are used.

5.2 Setup and Implementation

5.2.1 Setup for Pretraining

ImageNet For ImageNet training, we use the following setup to align with standard SimCLR training configurations:

Hyperparameters:

- Batch Size: 4096
- Epochs: 100
- Optimizer: LARS (Layer-wise Adaptive Rate Scaling)
  - Learning Rate: 4.8 (calculated as $0.3 \times \frac{\text{Batch Size}}{256}$)
  - Weight Decay: 10^{-6}
  - Learning Rate Schedule: Linear warmup for the first 10 epochs, followed by a cosine decay schedule without restarts.

CIFAR-10 Since CIFAR-10 images are significantly smaller (32×32 pixels) compared to ImageNet, several modifications were applied to adapt the training setup:

ResNet Architecture:

- Replaced the first 7×7 convolution (stride 2) with a 3×3 convolution (stride 1) to better fit the small image resolution.
- Removed the first max pooling operation.
- Additionally, we tested a standard ResNet model (without modifications). To adapt the standard ResNet model to CIFAR-10's smaller resolution, all images were resized to 224×224 pixels after cropping.

Data Augmentation:

- The same augmentation strategy as ImageNet training was applied.
- Images were resized to 32×32 pixels after cropping.
- Gaussian Blur was omitted since it is less effective on small images.

Hyperparameters:

- Batch Size: 1024, 128
- Learning Rate: 1.0
- Temperature Parameter: 0.1

All other settings, including the optimizer, weight decay, and overall training schedule, remain the same as those used in ImageNet training.

Segmentation-based augmentations For segmentation-based augmentations, we applied strong ColorJitter to the foreground to introduce appearance variations while preserving key object features. For the background, we applied Gaussian Blur with a probability of 0.85, reducing background details to encourage the model to focus on the main object. Additionally, we applied Grayscale conversion to the background with a probability of 0.3 to further enhance contrast between foreground and background. All other aspects of the training setup remained unchanged.

5.2.2 Setup for Fine-tuning

Fine-tuning is a crucial step in evaluating the effectiveness of self-supervised representations. To ensure a fair comparison with the original SimCLR fine-tuning procedure, we closely follow their training setup while adapting it to our pretraining constraints. We fine-tune the pretrained base encoder using the Nesterov momentum optimizer with the following hyperparameters:

- Batch Size: 4096
- Momentum: 0.9
- Learning Rate: 0.8
- No weight decay or additional regularization.

For data augmentation, we apply only random cropping, horizontal flipping, and resizing to 224×224. For validation, images are resized to 256×256, and a single center crop 224×224 is used. Fine-tuning is conducted for 60 epochs with 1% labeled data and 30 epochs with 10% labeled data.

Adapting Fine-Tuning to the Pretraining Setup A key distinction in our reproduction is that our SimCLR models were pretrained on only 10% of ImageNet training data, whereas the original study used 100% of ImageNet. As a result, our fine-tuning label fractions differ conceptually:

- Fine-tuning on 10% labeled data in our setup is equivalent to fine-tuning on 100% of our pretraining dataset.

  - This means that our 10% fine-tuning condition corresponds to fully supervised training on our 10% pretraining subset.
  - Although labeled as "fine-tuning 10%", in practice, it represents supervised learning on our entire pretraining subset.
- Fine-tuning on 1% labels corresponds to 10% of our pretraining dataset.

This adjustment is necessary because applying 1% of the original ImageNet labels to only 10% of the dataset would produce an extremely small labeled subset, making fine-tuning ineffective. While this adjustment distorts direct comparability with the original results, it still provides valuable insights into the effectiveness of fine-tuning self-supervised representations under limited data availability.

5.2.3 Setup for Transfer Learning

For transfer learning, we closely follow the experimental setup outlined in the original SimCLR study. However, due to computational constraints, we use Stochastic Gradient Descent (SGD) instead of the L-BFGS optimizer used in the original study.

Preprocessing All images were resized to 224 pixels along the shorter side using bicubic resampling. Next, we applied a 224×224 center crop to maintain consistent image dimensions.

To assess the quality of the learned representations, no additional data augmentations were applied during transfer learning.

5.2.4 Evaluation Metrics

After pretraining, we evaluate the learned representations using linear evaluation. This involves extracting frozen representations from the pretrained encoder. Next, a linear classifier is trained on these representations using the training set. Classification accuracy is then measured on the validation set to assess performance. Finally, the Top-1 and Top-5 accuracy scores are reported as the main results.

5.2.5 Implementation

For the implementation of our training pipeline, we primarily use PyTorch due to its flexibility and efficiency for deep learning applications. Several additional frameworks and libraries are integrated to optimize training and augmentation processes.
Frameworks and Libraries:

- PyTorch: Used as the core deep learning framework for model training and inference.

- PyTorch Lightning Flash: Utilized for the LARS optimizer, which improves large-batch training stability.
- Torchvision.transforms: Used for data augmentations, including random cropping, flipping, color jittering, grayscale conversion, and Gaussian blur.

Gradient Accumulation To efficiently train with large batch sizes while managing GPU memory constraints, gradient accumulation is implemented. Instead of updating model weights after every batch, gradients are accumulated over multiple mini-batches before performing a single optimizer step. Specifically, the loss is divided by the accumulation step size before backpropagation:

$$\text{Loss}_{\text{adjusted}} = \frac{\text{Loss}}{\text{accumulate\_steps}} \tag{4}$$

This approach effectively simulates a larger batch size without exceeding memory limits, allowing for more stable training. The following is the core PyTorch training loop that implements gradient accumulation:

Algorithm 1 SimCLR Training Loop with Gradient Accumulation

```
for batch_idx, ((view_1, view_2), _) in enumerate(train_loader):

    # Forward pass through the encoder
    h_1, h_2 = encoder(view_1), encoder(view_2)

    # Pass through the projection head
    z_1, z_2 = projection_head(h_1), projection_head(h_2)

    # Compute contrastive loss (normalized for accumulation)
    loss = loss_fn(z_1, z_2) / accumulate_steps
    loss.backward()
    total_loss += loss.item() * accumlate_steps

    # Perform optimizer step after accumulating gradients
    if (batch_idx + 1) % accumulate_steps == 0
    or (batch_idx + 1 == len(train_loader)):
        optimizer.step()
        scheduler.step()
        optimizer.zero_grad()

return total_loss / len(train_loader)
```

5.3 Validating the Pipeline: Reproducing CIFAR-10 Results

Due to computational constraints, training SimCLR on the full ImageNet dataset is not feasible. To ensure the correctness and reproducibility of our implementation, we use CIFAR-10 as a benchmark to validate our framework in a computationally manageable setting. This allows us to assess the consistency of our self-supervised learning pipeline before applying it to larger datasets.

Reproducing SimCLR results on CIFAR-10 is a crucial verification step to ensure our self-supervised learning pipeline adheres to the original methodology. By confirming the consistency of learned representations on CIFAR-10, we establish confidence in our framework before scaling up to more complex datasets such as ImageNet.

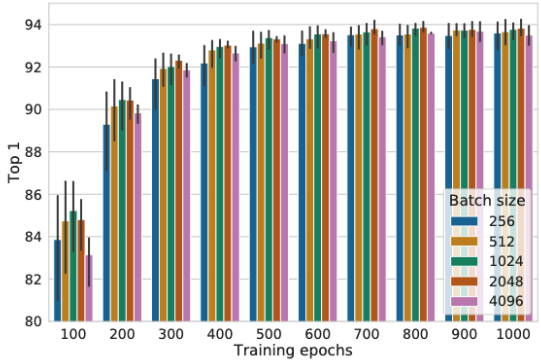

Figure 13: Linear evaluation of BaseEncoder trained with different batch sizes and epochs on the CIFAR-10 dataset. Each bar represents an average over three runs with learning rates of 0.5, 1.0, and 1.5, and a temperature of 0.5. Reproduced from Chen et al. (2020a).

The original SimCLR study reports a Top-1 accuracy range of 83% to 85% when trained for 100 epochs on CIFAR-10.

5.3.1 Pretraining Loss Analysis

To assess the convergence behavior of our self-supervised training, we monitor the NT-Xent loss throughout the pretraining phase. Figure 14 presents the loss progression over 100 epochs, indicating how well the model optimizes its contrastive objective.

The consistent decrease in the loss function indicates that the model is effectively optimizing the contrastive loss, resulting in meaningful feature representations. A well-optimized contrastive learning process is essential for ensuring that the encoder learns discriminative representations, which we further evaluate through linear classification.

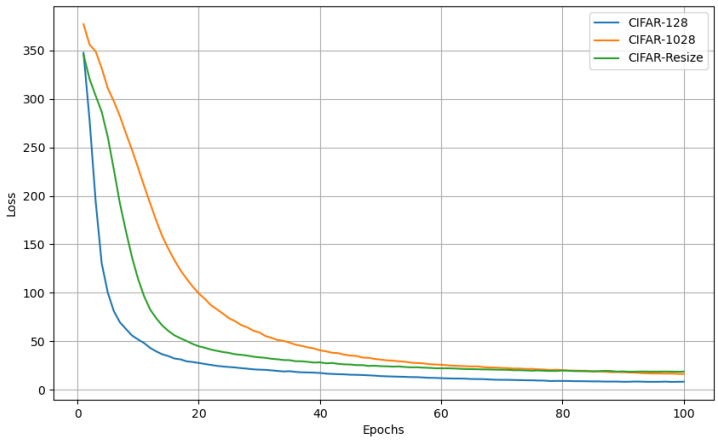

Figure 14: Pretraining loss curve for SimCLR on CIFAR-10. The NT-Xent loss decreases over epochs, indicating improved representation learning.

5.3.2 Training and Validation Accuracy

To assess the quality of learned representations, we conduct a linear evaluation by training a classifier on top of the frozen representations. Figure 15 shows the Top-1 accuracy achieved by models trained under different settings.

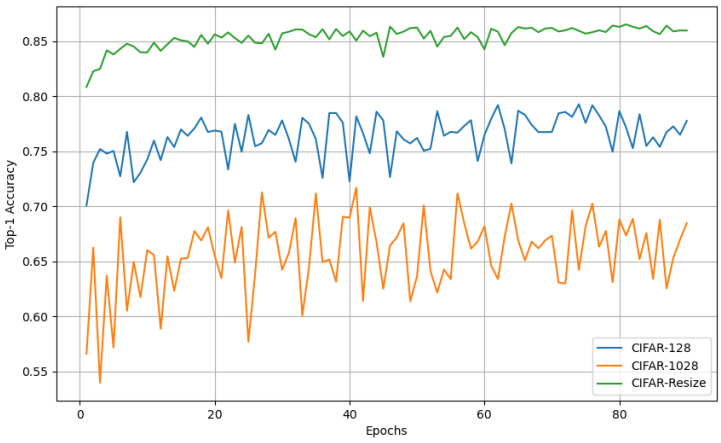

Figure 15: CIFAR-10 Top-1 accuracy of different pretrained models.

The results of our CIFAR-10 linear evaluation are as follows:

- SimCLR with batch size 1024: Achieves a Top-1 accuracy of 68.5%.
- SimCLR with batch size 128: Achieves a significantly higher Top-1 accuracy of 77.8%.

- SimCLR with resized input: Yields the best performance with a Top-1 accuracy of 86.0%.

These results demonstrate that our SimCLR implementation effectively captures meaningful features, as indicated by the increasing trend in linear evaluation accuracy. The difference in performance across batch sizes highlights the sensitivity of contrastive learning to training hyperparameters.

Successfully reproducing SimCLR's results on CIFAR-10 confirms the correctness of our self-supervised learning pipeline in a controlled setting. This serves as a crucial verification step before extending our approach to larger datasets such as ImageNet.

5.4 Evaluating the Base Encoder: Reproducing Table 6

To assess the effectiveness and reproducibility of SimCLR's self-supervised learning approach under limited data constraints, we train and evaluate the base encoder on a 10% subset of ImageNet. This section reproduces Table 6 from the original SimCLR paper, comparing the linear classifier accuracy of models trained on self-supervised representations using different ResNet architectures:

- ResNet-50
- ResNet-50 (2×)
- ResNet-50 (4×)

These experiments also aim to quantify how well the learned representations generalize when trained on a reduced dataset and evaluated with a linear classifier on ImageNet. Table 1 presents the linear classification accuracy of the reproduced ResNet-50 model compared to the original SimCLR results. As expected, training on only 10% of the ImageNet dataset significantly reduces classification accuracy due to the limited amount of data available.

Table 1: ImageNet accuracies of linear classifiers.

Method	**Architecture**	**Param (M)**	**Top 1**	**Top 5**
SimCLR original results:				
SimCLR	ResNet-50	24	69.3	89.0
SimCLR	ResNet-50 (2×)	94	74.2	92.0
SimCLR	ResNet-50 (4×)	375	76.5	93.2
SimCLR reproduced results:				
SimCLR	ResNet-50	24	34.6	59.3
SimCLR	ResNet-50 (2×)	94	35.8	60.6
SimCLR	ResNet-50 (4×)	375	36.4	61.2

5.4.1 ResNet-50

Pretraining Performance Figure 16 illustrates the pretraining loss curve for the standard ResNet-50 model, showing the optimization progress during contrastive learning. The loss starts at 4.9 and decreases steadily to 0.1, indicating effective minimization of the contrastive objective.

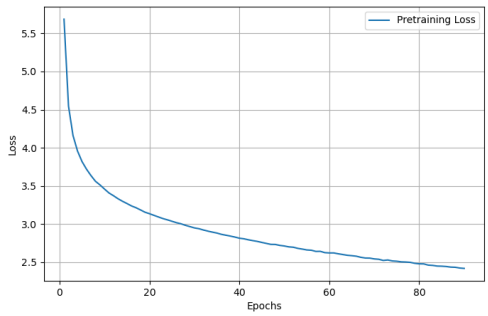

Figure 16: Pretraining loss curve for ResNet-50 trained on 10% of ImageNet.

Linear Evaluation Performance The linear evaluation results, illustrated in Figure 17, demonstrate the training and validation accuracy progression. The model starts with a validation accuracy of 15.2%, which gradually improves to 34.6%. While the absolute accuracy is lower due to the reduced dataset size, the learning trend remains consistent with full ImageNet training, confirming that self-supervised learning effectively extracts useful representations even with limited data.

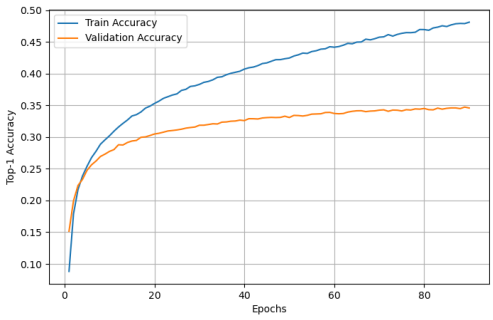

Figure 17: Linear evaluation accuracy of BaseEncoder pretrained with ResNet-50 (1×).

5.4.2 ResNet-50 (2×)

Pretraining Performance Increasing the model capacity to ResNet-50 (2×) results in a slightly improved pretraining performance. As shown in Figure 18, the contrastive loss starts at 4.5 and decreases to 0.09, reaching a lower final value compared to ResNet-50. This suggests that the increased network capacity leads to better feature extraction during self-supervised training.

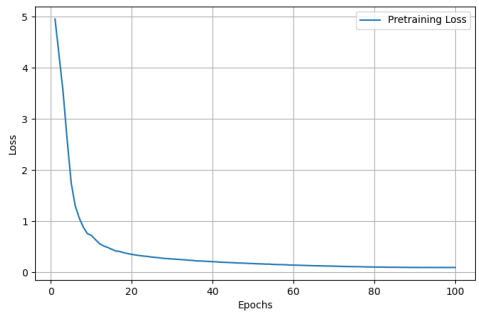

Figure 18: Pretraining loss curve for ResNet-50 (2×) trained on 10% of ImageNet.

Linear Evaluation Performance The linear evaluation results for ResNet-50 (2×) are presented in Figure 19. The model begins with a validation accuracy of 17.0% and improves to 35.8%. Compared to ResNet-50, the Top-1 accuracy increases from 34.6% to 35.8%, reflecting a 1.2% improvement.

For reference, in the full ImageNet setting, SimCLR achieves a 4.9% increase in performance when scaling from ResNet-50 to ResNet-50 (2×). However, in our data-constrained 10% ImageNet experiment, the improvement is only 1.2%. Although the gain is smaller, the learning trend remains consistent with the original results, confirming that wider models can still improve representation quality, although with diminishing returns under limited data conditions.

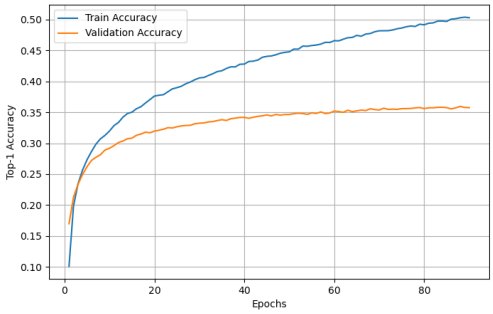

Figure 19: Linear evaluation accuracy of BaseEncoder pretrained with ResNet-50 (2×).

5.4.3 ResNet-50 (4×)

Pretraining Performance The largest model tested, ResNet-50 (4×), achieves the lowest pretraining loss among all architectures. As illustrated in Figure 20, the loss starts at 4.48 and decreases to 0.089. The more substantial reduction in contrastive loss suggests that scaling the model width further enhances feature separability during contrastive learning.

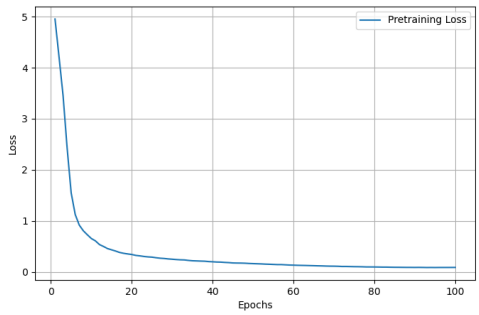

Figure 20: Pretraining loss curve for ResNet-50 (4×) trained on 10% of ImageNet.

Linear Evaluation Performance The training and validation accuracy curves for ResNet-50 (4×) are presented in Figure 21. The model begins with a validation accuracy of 18.2%, improving to 36.4% after training. Compared to ResNet-50 (2×), the Top-1 accuracy increases from 35.8% to 36.4%, representing a 0.6% gain.

For reference, in the full ImageNet setting, increasing the model size from ResNet-50 (2×) to ResNet-50 (4×) improves accuracy by 2.3%. However, in our 10% ImageNet experiment, the gain is only 0.6%. Despite the smaller improvement, the performance trend remains aligned with the original findings, demonstrating that increasing model capacity continues to enhance representation learning.

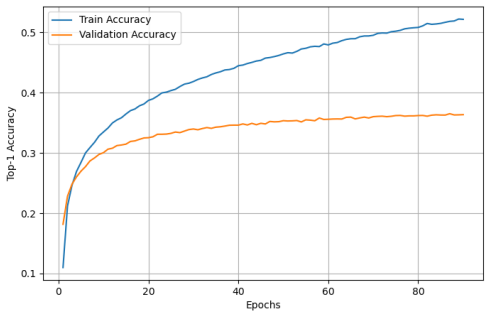

Figure 21: Linear evaluation accuracy of BaseEncoder pretrained with ResNet-50 (4×).

5.5 Evaluating the Segmentation-Based Augmentation Pipeline: Reproducing Table 6

To assess the effectiveness of our segmentation-based augmentation pipeline, we compare its performance against the standard SimCLR augmentation strategy using the linear evaluation protocol on ImageNet. This experiment evaluates whether applying region-specific augmentations to foreground and background regions can improve self-supervised representation learning over the standard augmentation pipeline.

Table 2 presents the Top-1 and Top-5 classification accuracies of two training pipelines:

1. Reproduced SimCLR results trained using the standard SimCLR augmentation pipeline.
2. SimCLR with Segmentation-Based Augmentations, where we apply different augmentation strategies to the foreground and background, as described in Section 4.2.

Table 2: ImageNet accuracies of standard reproduced results and segmentation-based augmentations results.

Method	Architecture	Param (M)	Top 1	Top 5
SimCLR reproduced results:				
SimCLR	ResNet-50	24	34.6	59.3
SimCLR segmentation based augmentations results:				
SimCLR	ResNet-50	24	35.2	60.11

The results indicate a minor improvement in linear classifier accuracy when using segmentation-based augmentations. The Top-1 accuracy increases from 34.6% to 35.2%, and the Top-5 accuracy improves from 59.3% to 60.11%.

5.6 Reproducing Table 7: Fine-tuning the Base Encoder

Table 3 presents the Top-1 and Top-5 accuracy for different ResNet architectures at 1% and 10% label fractions, comparing the original SimCLR results to our reproduced results.

Fine-tuning with 10% labeled data significantly improves performance. For instance, the Top-1 accuracy for ResNet-50 (1×) increases from 34.6% in the linear evaluation setting to 44.3% when fine-tuned with 10% labeled data. However, in our case, fine-tuning with 10% labeled data is effectively equivalent to training on 100% of the pretraining dataset, as our pretraining was conducted on only 10% of ImageNet. This means our 10% fine-tuning setup is conceptually closer to a fully supervised setup within the pretraining constraints rather than a direct replication of the original SimCLR setting.

Method	Architecture	Label Fraction 1%		Label Fraction 10%	
		Top 1	Top 5	Top 1	Top 5
SimCLR original results:					
SimCLR	ResNet-50	48.3	75.5	65.6	87.8
SimCLR	ResNet-50 (2×)	58.5	83.0	71.7	91.2
SimCLR	ResNet-50 (4×)	63.0	76.5	74.7	92.6
SimCLR reproduced results:					
SimCLR	ResNet-50	19.2	40.0	44.3	69.9
SimCLR	ResNet-50 (2×)	20.3	41.9	46.5	71.8
SimCLR	ResNet-50 (4×)	20.9	42.8	46.6	71.9

Table 3: Comparison of different methods on ImageNet with varying label fractions.

In our setup, fine-tuning with 1% labeled data corresponds to using 10% of the pretraining dataset. Since we only pretrained on 10% of ImageNet, applying 1% of ImageNet labels directly would yield an extremely small labeled subset, limiting fine-tuning effectiveness. To ensure a meaningful comparison, we adjust our 1% fine-tuning condition to correspond proportionally to 10% of our pretraining dataset, providing a more representative evaluation of label efficiency under limited pretraining data.

Comparing fine-tuned results with linear evaluation reveals the direct impact of additional labeled supervision. In the 1% fine-tuning condition, the Top-1 accuracy for ResNet-50 (1×) reaches 19.2%, which is significantly lower than the 34.6% obtained through linear evaluation. Similarly, for ResNet-50 (2×), fine-tuning with 1% labels achieves 20.3% accuracy, whereas the linear evaluation model without fine-tuning achieves 35.8%. The ResNet-50 (4×) model follows the same trend, with 1% fine-tuning resulting in 20.9% accuracy compared to 36.4% in the linear evaluation setting. These results suggest that with limited self-supervised pretraining, fine-tuning on a small labeled subset may not always surpass the performance of a linear classifier applied to frozen representations.

Increasing network capacity (ResNet-50 $\rightarrow$ 2× $\rightarrow$ 4×) improves performance, but with diminishing returns. Larger models consistently achieve higher accuracy, reinforcing that increasing model width enhances representation learning. However, under constrained pretraining data, the relative performance gains from wider architectures are smaller than in the full ImageNet setting. While scaling the model size remains beneficial, the performance increase is more limited when self-supervised learning is performed on a reduced dataset.

5.7 Transfer learning performance: Reproducing Table 8

Table 4 presents the linear evaluation accuracy of SimCLR, comparing standard and segmentation-based augmentations across six datasets. For CIFAR10, SimCLR (segaug) achieves a slightly higher accuracy (82.8%) compared to SimCLR (normal) at 82.4%. On CIFAR100, the difference remains modest, with SimCLR (segaug) at 59.4% and SimCLR (normal) at 59.1%. For Flowers, the segmentation-based approach achieves 67.9% accuracy, outperforming the standard method by approximately 1 percentage point.

In contrast, for Pets, SimCLR (normal) performs slightly better (55.0%) than SimCLR (segaug) at 54.2%. A similar trend is observed in Caltech-101, where the standard pipeline achieves 83.2% compared to 81.3% for the segmentation variant. On SUN397, both approaches yield similar results, with segmentation-based SimCLR scoring 52.0% and the standard pipeline achieving 51.9%.

These findings indicate that segmentation-based augmentations can enhance performance on datasets with diverse or complex object compositions, such as CIFAR10, CIFAR100, and Flowers. However, for certain datasets like Pets and Caltech-101, the standard approach yields slightly better results. In general, segmentation-based augmentations enhance data variability while preserving semantic structure; however, their effectiveness depends on dataset characteristics.

	CIFAR10	**CIFAR100**	**Flowers**	**Pets**	**SUN397**	**Caltech-101**
Linear evaluation:						
SimCLR (normal)	82.4	59.1	66.9	**55.0**	51.9	**83.2**
SimCLR (segaug)	**82.8**	**59.4**	**67.9**	54.2	**52.0**	81.3

Table 4: Linear evaluation results of standard SimCLR compared to the segmentation-based SimCLR pipeline.

6 Discussion

This section synthesizes the findings from our experiments on self-supervised representation learning using SimCLR and its segmentation-based augmentation variant. We examined model performance under constrained data settings, explored the impact of different ResNet architectures, and evaluated how segmentation-informed augmentations may benefit transfer learning.

6.1 Experiments in Detail

Evaluating the Base Encoder: Reproducing Table 6 In reproducing the original SimCLR experiments at a reduced scale (10% of ImageNet), we observe substantial drops in accuracy compared to the original results. While the loss curves indicate that the networks effectively minimize the contrastive objective, the final linear evaluation accuracies (approximately 34.6%–36.4% Top-1) are significantly lower than the 69.3%–76.5% range reported for full ImageNet. This underscores the data-intensive nature of deep learning models: without sufficient data variety, the networks struggle to capture the full breadth of visual concepts.

However, incorporating segmentation-based augmentations led to a slight increase in linear evaluation accuracy. The Top-1 accuracy improved from 34.6% to 35.2% when applying region-specific augmentations. This suggests that selective transformations applied to the foreground and background may introduce more useful variance while preserving essential object features. The Top-5 accuracy also improved from 59.3% to 60.1%, reinforcing the idea that segmentation-informed augmentations might provide some benefits in a contrastive learning setting.

Interestingly, we still find a consistent improvement with larger models (ResNet-50 1×, 2×, and 4×), signaled by both progressively lower pretraining losses and gradually higher linear evaluation accuracies. This scaling trend aligns with the original SimCLR findings, but with a smaller absolute performance gain under data constraints. Increasing capacity provides more representational flexibility, enabling better feature disentanglement even in a limited-data regime. However, the marginal gains (1.2% from 1× to 2×, and 0.6% from 2× to 4×) are far lower than in the full ImageNet setting, suggesting that larger networks cannot fully leverage their potential with only 10% of the data.

Fine-Tuning the Base Encoder: Reproducing Table 7 When comparing fine-tuning results (1% vs. 10% labeled data) with linear evaluation, we see a complex interplay between labeled data availability and the quality of self-supervised representations. Fine-tuning on 10% of ImageNet substantially boosts performance over linear evaluation. For ResNet-50 (1×), the Top-1 accuracy increases from 34.6% to 44.3%, reflecting the benefit of supervised learning. However, in our setup, 10% fine-tuning is effectively a fully supervised scenario relative to the 10% pretraining

subset, making it less comparable to the original SimCLR experiments where the model had exposure to the entire ImageNet distribution.

By contrast, 1% fine-tuning performs noticeably worse than linear evaluation. For instance, the ResNet-50 (1×) accuracy drops to 19.2% under 1% fine-tuning compared to 34.6% in linear evaluation. This outcome suggests that when both pretraining data and fine-tuning labels are extremely limited, the advantage of supervised adaptation diminishes, possibly due to overfitting or insufficient labeled examples to guide the training process. In essence, if the base encoder has only seen a fraction of ImageNet during pretraining, providing even fewer labeled samples during fine-tuning may not be sufficient to unlock further performance gains.

Transfer Learning with Segmentation-Based Augmentations: Reproducing Table 8 Our experiments on six downstream datasets (CIFAR10, CIFAR100, Flowers, Pets, SUN397, and Caltech-101) show that segmentation-based augmentations can yield modest gains in datasets characterized by diverse or complex object compositions (e.g., CIFAR10, CIFAR100, Flowers). However, for Pets and Caltech-101, the classical SimCLR pipeline remains slightly ahead. These mixed results highlight the context-dependent nature of segmentation-informed data augmentation. In some settings, focusing augmentations on foreground or background regions preserves crucial features and enriches the model's learned representations. In other cases, especially in object-centric datasets with simpler compositions, uniform global augmentations may already capture enough variance.

A potential explanation for these variations lies in the quality of segmentation. If the segmentation model used for generating masks misclassifies objects and backgrounds, the localized augmentations might distort relevant information or omit beneficial background details. Additionally, certain datasets, such as Caltech-101, often place clear primary objects in near-uniform backgrounds, reducing the net benefit of segmentation-based augmentations. Hence, segmentation-based approaches may be most valuable in datasets where important structures are scattered throughout the scene and segmentation can reliably isolate the object(s) of interest.

6.2 Overall Findings and Future Considerations

From a broader perspective, these experiments underscore several important points:

- **Data Scale is Crucial.** Even a robust self-supervised method like SimCLR suffers substantially when pretraining data is reduced to 10% of ImageNet, underscoring the well-known data-hungry nature of deep learning models.
- **Model Scaling Yields Diminishing Returns with Limited Data.** Wider architectures continue to improve performance but only marginally at reduced scales, suggesting a clear interplay between network capacity and dataset diversity.
- **Segmentation-Based Augmentations Offer Selective Benefits.** While segmenting images before augmentation can produce gains in certain datasets, its effectiveness is heavily dependent on dataset structure, segmentation quality and the complexity of images.
- **Fine-Tuning vs. Linear Evaluation Under Limited Data.** In extremely data-constrained scenarios, linear evaluation sometimes outperforms fine-tuning, implying that the label efficiency gains from self-supervised features might not always materialize if the fine-tuning set is too small.

Looking ahead, more precise or adaptive segmentation methods could further enhance localized augmentations, especially in heterogeneous datasets where important details span different regions of the image. Additionally, scaling up pretraining data, while maintaining segmentation-based augmentation, could clarify whether the relatively modest gains observed here scale more convincingly when the model sees a larger variety of unlabeled examples. Finally, exploring semi-supervised or active learning strategies might yield additional insights into how best to harness limited labeled data alongside self-supervised or segmentation-driven pretraining.

7 Conclusion

In this thesis, we have investigated the potential of SSL to overcome the challenges posed by limited annotated data, as well as the effectiveness of segmentation-based augmentations in enhancing contrastive representation learning. Our experiments focused primarily on SimCLR, a widely adopted SSL framework, and evaluated its performance across various scales, network architectures, and augmentation strategies.

First, by reproducing select experiments from SimCLR at a reduced scale (e.g., training with only 10% of ImageNet), we verified that SSL can still extract meaningful features under data scarcity, but with lower absolute performance. We observed that larger network architectures (ResNet-50 1×, 2×, 4×) continue to offer incremental gains, although the margin of improvement is smaller than in the full ImageNet setting. This finding highlights the interplay between data size and model capacity: insufficient data can constrain the benefits gained from increasing model width.

Second, we compared linear evaluation and fine-tuning protocols under reduced data conditions. As expected, fine-tuning with 10% labeled data significantly outperformed linear evaluation, since this setting corresponds to fully supervised training on the entire pretraining dataset. This confirms that when enough labeled data is available, fine-tuning meaningfully improves performance by refining self-supervised features. However, fine-tuning with 1% labeled data (which effectively corresponds to 10% of our pretraining dataset) performed significantly worse than linear evaluation. This suggests that when both pretraining data and fine-tuning labels are severely limited, the advantage of supervised adaptation diminishes.

Third, we integrated segmentation-based augmentations into SimCLR to investigate whether localizing augmentations to specific regions (foreground vs. background) could improve feature learning. Although these segmentation-driven approaches yielded modest gains in some datasets (e.g., ImageNet, CIFAR10, CIFAR100, Flowers, SUN397), the benefits were less consistent in object-centric datasets like Caltech-101 and Pets. Our observations suggest that segmentation quality and dataset composition heavily influence the degree to which region-specific augmentations offer an advantage. In particular, segmentation-based augmentation appears most beneficial in scenarios with complex object structures and diverse backgrounds, provided the segmentation masks accurately capture meaningful regions.

Overall, our findings emphasize the significant role that data diversity and augmentation techniques play in the success of SSL. Data-hungry deep networks may still yield strong results under limited data conditions. Segmentation-based augmentations represent a promising direction for future research, particularly if paired with robust segmentation models and applied to datasets where local context is critical. Moreover, exploring semi-supervised or active learning paradigms may further enhance the utilization of SSL in real-world scenarios, where annotated data remains a scarce and costly resource.

A Appendix

A.1 Code Availability

The implementation of this thesis is open-source and available on GitHub:

https://github.com/saswo/SimCLR---segmentation-based

Researchers and practitioners are encouraged to explore, reproduce, and extend the work presented in this thesis.

Bibliography

Randall Balestriero, Mark Ibrahim, Vlad Sobal, Ari Morcos, Shashank Shekhar, Tom Goldstein, Florian Bordes, Adrien Bardes, Gregoire Mialon, Yuandong Tian, Avi Schwarzschild, Andrew Gordon Wilson, Jonas Geiping, Quentin Garrido, Pierre Fernandez, Amir Bar, Hamed Pirsiavash, Yann LeCun, and Micah Goldblum. A cookbook of self-supervised learning, 2023. URL https://arxiv.org/abs/2304.12210.

Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments, 2021. URL https://arxiv.org/abs/2006.09882.

O. Chapelle, B. Scholkopf, and A. Zien, Eds. Semi-supervised learning (chapelle, o. et al., eds.; 2006) [book reviews]. *IEEE Transactions on Neural Networks*, 20(3): 542–542, 2009. doi: 10.1109/TNN.2009.2015974.

Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation, 2017. URL https://arxiv.org/abs/1706.05587.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020a. URL https://arxiv.org/abs/2002.05709.

Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning, 2020b. URL https://arxiv.org/abs/2003.04297.

Adam Coates, Andrew Y Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011. URL https://cs.stanford.edu/~acoates/papers/coatesleeng_aistats_2011.pdf.

Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding, 2016. URL https://arxiv.org/abs/1604.01685.

Ekin D. Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation policies from data, 2019a. URL https://arxiv.org/abs/1805.09501.

Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space, 2019b. URL https://arxiv.org/abs/1909.13719.

Pádraig Cunningham, Matthieu Cord, and Sarah Jane Delany. Supervised learning. In *Machine learning techniques for multimedia: case studies on organization and retrieval*, pages 21–49. Springer, 2008.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. doi: 10.1109/CVPR.2009.5206848.

Andre Esteva, Alexandre Robicquet, Bharath Ramsundar, Volodymyr Kuleshov, Mark DePristo, Katherine Chou, Claire Cui, Greg Corrado, Sebastian Thrun, and Jeff Dean. A guide to deep learning in healthcare. *Nature Medicine*, 25, 01 2019. doi: 10.1038/s41591-018-0316-z.

Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *2004 Conference on Computer Vision and Pattern Recognition Workshop*, pages 178–178, 2004. URL `https://api.semanticscholar.org/CorpusID:2156851`.

Ruohan Gao Fei-Fei Li, Jiajun Wu. Self-supervised learning, 2022. URL `https://cs231n.stanford.edu/slides/2022/lecture_14_jiajun.pdf`. PowerPoint slides.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012. doi: 10.1109/CVPR.2012.6248074.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning, 2020. URL `https://arxiv.org/abs/2006.07733`.

Michael Gutmann and Aapo Hyvärinen. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 297–304, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL `https://proceedings.mlr.press/v9/gutmann10a.html`.

Raia Hadsell, Sumit Chopra, and Yann Lecun. Dimensionality reduction by learning an invariant mapping. pages 1735 – 1742, 02 2006. ISBN 0-7695-2597-0. doi: 10.1109/CVPR.2006.100.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning, 2020. URL https://arxiv.org/abs/1911.05722.

Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey, 2019. URL https://arxiv.org/abs/1902.06162.

Youmin Kim, A. F. M. Shahab Uddin, and Sung-Ho Bae. Local augment: Utilizing local bias property of convolutional neural networks for data augmentation. *IEEE Access*, 9:15191–15199, 2021. doi: 10.1109/ACCESS.2021.3050758.

Alexander Kirillov, Kaiming He, Ross Girshick, Carsten Rother, and Piotr Dollár. Panoptic segmentation, 2019. URL https://arxiv.org/abs/1801.00868.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything, 2023. URL https://arxiv.org/abs/2304.02643.

Rayan Krishnan, Pranav Rajpurkar, and Eric J. Topol. Self-supervised learning in medicine and healthcare. *Nature Biomedical Engineering*, 6(12):1346–1352, Dec 2022. doi: 10.1038/s41551-022-00914-1. URL https://doi.org/10.1038/s41551-022-00914-1. Epub 2022 Aug 11.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521 (7553):436–444, 2015.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2015. URL https://arxiv.org/abs/1405.0312.

Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, and Demetri Terzopoulos. Image segmentation using deep learning: A survey, 2020. URL https://arxiv.org/abs/2001.05566.

Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics Image Processing*, pages 722–729, 2008. doi: 10.1109/ICVGIP.2008.47.

Viktor Olsson, Wilhelm Tranheden, Juliano Pinto, and Lennart Svensson. Classmix: Segmentation-based data augmentation for semi-supervised learning, 2020. URL https://arxiv.org/abs/2007.07936.

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22:1345–1359, 2010. URL https://api.semanticscholar.org/CorpusID:740063.

Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3498–3505, 2012. URL https://api.semanticscholar.org/CorpusID:383200.

G. Reenu and M. A. Saleem Durai. Intelligent video surveillance: a review through deep learning techniques for crowd analysis. *Journal of Big Data*, 6:48, 2019. doi: 10.1186/s40537-019-0212-5. URL https://doi.org/10.1186/s40537-019-0212-5.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation, 2015. URL https://arxiv.org/abs/1505.04597.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, page 815–823. IEEE, June 2015. doi: 10.1109/cvpr.2015.7298682. URL http://dx.doi.org/10.1109/CVPR.2015.7298682.

Hyun Oh Song, Yu Xiang, Stefanie Jegelka, and Silvio Savarese. Deep metric learning via lifted structured feature embedding, 2015. URL https://arxiv.org/abs/1511.06452.

Yonglong Tian, Chen Sun, Ben Poole, Dilip Krishnan, Cordelia Schmid, and Phillip Isola. What makes for good views for contrastive learning?, 2020. URL https://arxiv.org/abs/2005.10243.

Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2019. URL https://arxiv.org/abs/1807.03748.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2023. URL https://arxiv.org/abs/1706.03762.

Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 3485–3492, 2010. URL https://api.semanticscholar.org/CorpusID:1309931.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features, 2019. URL https://arxiv.org/abs/1905.04899.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks, 2017. URL https://arxiv.org/abs/1605.07146.

Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction, 2021. URL https://arxiv.org/abs/2103.03230.

Declaration of Authorship

Ich erkläre hiermit gemäß §9 Abs. 12 APO, dass ich die vorstehende Abschlussarbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe. Des Weiteren erkläre ich, dass die digitale Fassung der gedruckten Ausfertigung der Abschlussarbeit ausnahmslos in Inhalt und Wortlaut entspricht und zur Kenntnis genommen wurde, dass diese digitale Fassung einer durch Software unterstützten, anonymisierten Prüfung auf Plagiate unterzogen werden kann.

Place, Date

Signature