Imperial College London

BEng Individual Project

Imperial College London

Department of Computing

Evaluating SimCLR for Medical Image Classification

Author:
Freddy Jiang

Supervisor:
Dr. Ben Glocker

Second Marker:
Dr. Benjamin Hou

June 19, 2023

Abstract

Computer-aided diagnosis (CADx) plays a crucial role in assisting radiologists with interpreting medical images. Over recent years, there has been significant advancements in image classification models, such as deep neural networks and Vision Transformers. Training such models require lots of labelled data, a prerequisite often not met in medical environments as labelling images is time-consuming and requires expertise.

An alternative training paradigm is self-supervised learning, which involves pretraining a model with unlabelled data followed by finetuning it with labelled data. This paradigm has achieved strong performance on classifying natural images, even with limited labelled data.

This thesis aims to explore the potential of SimCLR, a state-of-the-art self-supervised learning framework, for medical image classification. We evaluate this framework on a wide range of medical imaging modalities, including colon pathology, dermatology, blood cells, retina fundus and other medical scans. We find significant improvement over baseline supervised metrics (an increase of up to 30.6% in accuracy). We simulate different data settings and explore tackling class imbalance, as well as transfer learning on different datasets. We find downsampling images to be a viable solution for some modalities in bringing down training times (12 hours to pretrain a model for classifying blood cells that achieves over 0.95 AUC after finetuning). We propose a novel augmentation sequence which shows consistent improvement over the original framework.

Acknowledgements

I would like to thank my supervisor, Dr. Ben Glocker, for advising and guiding me throughout my thesis, and for introducing me to the fascinating world of self-supervised learning and biomedical imaging.

I am also grateful to my second marker, Dr. Benjamin Hou, and my personal tutor, Dr. Josiah Wang, for offering valuable advice during my project.

Finally, I would like to recognise the unwavering support of my family and friends, who have always been there for me in university and beyond.

Contents

Chapter 1

Introduction

Machine learning and computer vision plays a significant role in computer-aided diagnosis (CADx) to assist radiologists with interpreting medical images[7, 8] and early detection of lesions[9]. Using supervised learning methods for image classification and segmentation tasks poses major challenges, as performance is hindered by scarcity of labelled data[10, 11]. Unfortunately, this is a common issue as annotating medical images are time-consuming and requires expertise.

Over recent years, there has been significant advancements in self-supervised learning (SSL) (Figure 1.1), a machine learning paradigm involving pretraining a model with unlabelled data followed by finetuning with labelled data. In particular, a subcategory of SSL is contrastive learning, which involves applying data augmentations to an image, then comparing similarities between the augmented images. These methods have demonstrated the capabilty of classifying natural images with high accuracy[4, 12, 13], even with a limited amount of labelled training data. Research[14, 15, 16] has revealed promising prospects for self-supervised learning for

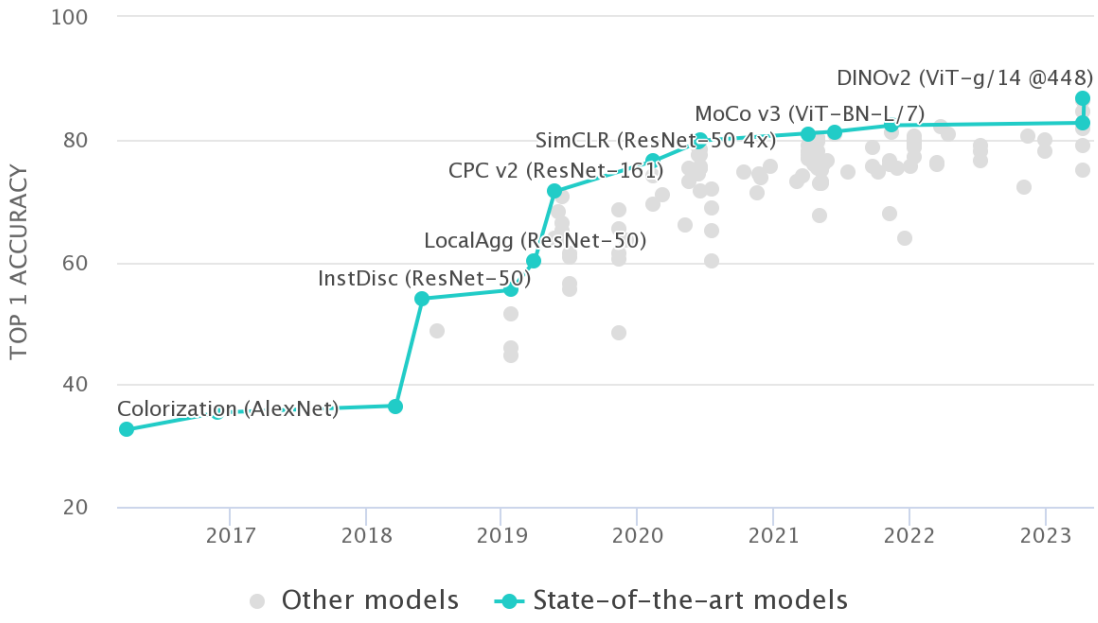


Figure 1.1: Progression of top-1 accuracy of self-supervised methods for image classification on ImageNet. As of June 2023, DINO[1] leads at 86.7% accuarcy, using Vision Transformers (ViT). ReLIC[2, 3] and SimCLR[4, 5] leads for ResNet models.[6]

medical imaging given an abundance of unlabelled data and a small amount of labelled data. However, features of medical images prove to be a difficult challenge. Adaptations to existing methods have been proposed[17, 18] to account for features such as greyscale colour space and low contrast.

In this thesis, we present a comprehensive evaluation on applying a state-of-the-art contrastive learning framework called SimCLR[4, 5] to medical image classification. We work towards determining the extent to which SimCLR is feasible given different environments (modalities, available data), as well as the extent to which it is necessary. We propose a novel augmentation sequence which improves classification performance in many modalities, as well as novel adaptations to apply SimCLR to a setting with a lack of unlabelled and labelled data. We hope our work contributes to a better understanding of the capabilities of SimCLR and contrastive learning for medical imaging tasks.

1.1 Objectives and Contributions

In this thesis, our overarching objective is to determine the potential of SimCLR for medical image classification. We break this into smaller objectives below and summarise our contributions for each objective.

- **Determine the extent to which the existing SimCLR framework transfers to medical image classification.**

  The original papers[4, 5] present SimCLR for image classification and evaluate the framework using natural images from ImageNet. We adopt their specifications and build a complete SimCLR framework in Python and PyTorch Lightning. We also build a collection of evaluation tools such as top-1 accuracy and AUC ROC metrics, PCA and t-SNE.

  Our results reveal that SimCLR pretraining improves over baseline supervised metrics by up to 30.6% accuracy for colon pathology and 15.3% accuracy for blood cells.

- **Devise improved settings for applying SimCLR to different medical imaging modalities.**

  We consider different augmentations sequences, including dropping random horizontal flip and random greyscale. We propose a novel augmentation sequence for medical images involving random histogram equalisation and random sharpness, as well as removing colour distortion for greyscale medical scans. We observe up to 2.9% increase in accuracy over the original sequence. We also observe pretraining with our proposed sequence is effective for greyscale images, improving over baseline supervised metrics by up to 28.5% accuracy for retinal OCT and 8.3% accuracy for tissue cells.

  Previous works[14, 17] indicated long training times as a limitation, especially with some modalities consisting of very large images. We use medical images from MedMNIST[19, 20], which downsamples source images to 28×28. We find SimCLR to be effective on these downsampled images.

- **In terms of availability of medical images, determine the extent to which SimCLR pretraining is necessary and/or feasible.**

  We simulate environments with 100, 250, 1000 labelled data and consider different approaches to finetuning. We observe that freezing the backbone (fixing the parameters of the CNN after pretraining) yields best results given a small amount of labelled data. We also perform experiments with large amounts of labelled data and conclude that pretraining can still improve over supervised learning by up to 2.8% accuracy.

We evaluate whether SimCLR pretraining is feasible given a lack of data by using a retina fundus dataset of 1,080 unlabelled and labelled images. We propose a workflow involving initial pretraining on a different, larger dataset before finetuning with the specialised retina fundus dataset. This workflow achieves up to 3.2% increase in accuracy over baseline metrics. The baselise metrics involve supervised learning, as well as pretraining with retina fundus images only.

Finally, we evaluate the feasibility of SimCLR on a heavily imbalanced dataset. We attempt to balance the dataset by performing undersampling. We conclude that using the full dataset yields better performance.

1.2 Challenges

The main challenges faced during this project were:

- **Long training times**

  The original paper[4] states that SimCLR benefits from large batch sizes and long training times. Pretraining a single model on a NVIDIA GPU cluster for CUDA-optimised AI frameworks takes 12 hours, even when source images are downsampled to 28×28. This introduces challenges on hyperparameter tuning and exploring different augmentations. For the former, we instead adopt hyperparameters previously tuned on the STL-10 dataset.

- **Diverse data modalities**

  A comprehensive evaluation of SimCLR on medical imaging involves performing experiments on many data modalities. This is a shortcoming of many previous works[15, 16, 18, 14]. In this thesis, we train over 170 ResNet models in 6 different modalities and use cloud computing (Bitbucket) to store them. We focus on 2D modalities only.

- **Explainability of results**

  Neural networks are notorious for being black-box models. They often lack transparency and interpretability. A significant hurdle lies in devising an augmentation sequence that is effective for medical images. While possessing a good understanding of medical features help, it remains difficult to ascertain the effectiveness of different augmentations and previous works would verify their proposals with a grid search. Unfortunately, this approach is infeasible for us due to limited time.

- **Unbiasedness of results**

  This thesis entails comparative analysis between SimCLR and supervised baselines. We train all models and collect all metrics internally to mitigate bias and external factors in our evaluation.

1.3 Ethical Considerations

This thesis is research-focused and uses data from the MedMNIST[20, 19] database. Training various models involves processing of previously collected personal data. All images are completely anonymised and personal information is unidentifiable. This research is exempt from ethical approval as the analysis is based on secondary data which is publicly available, and no permission is required to access the data.

SimCLR is open source and can be used and extended for research purposes[4].

Models in this thesis are presented as means of research only. It is not applicable in industrial practice. Contrastive and non-contrastive learning frameworks have a wide range of applications, as they learn representations that can be used for a variety of downstream tasks[21]. Although this thesis focuses on medical imaging, there remains a small potential for presented findings to be misused in other industries, such as for military purposes. We do not warrant nor take liability of any misuse of results associated with this thesis.

This thesis involves heavy computation to train and finetune hundreds of deep neural networks, which consumes significant amounts of energy and may raise environmental concerns. We make diligent efforts to reduce environmental impact by limiting model sizes and using efficient, optimised systems for AI like CUDA frameworks and Slurm Workload Manager.

Chapter 2

Background

2.1 Deep Neural Networks

A deep neural network (DNN) is a deep learning architecture comprised of a multi-layered artificial neural network. Due to the number of parameters, training a DNN requires large datasets and heavy computational power. Over recent years, the availability of cheap data storage and computation has made training DNNs feasible, achieving high performance and accuracy in fields such as automatic speech recognition[22], image recognition[23] and natural language processing[24]. DNNs are powerful due to their scalability and ability to extract features from data.

2.1.1 Multilayer Perceptron

An artificial neuron, or neuron for short, is a computational model devised from the behaviour of a biological neuron. We provide its formal definition as follows.

Definition 2.1 (Artificial Neuron)**.** *An artificial neuron is a function f that linearly transforms an input vector x, then applies an activation function ϕ, as described by* (2.1) *where w, b are constants.*

$$f(x) = \phi\left(w^T x + b\right) \tag{2.1}$$

w and b are referred to as weights and bias respectively.

An example of a neuron is the perceptron, where ϕ is the threshold function. A perceptron can learn any linearly separable function[25].

$$\text{Perceptron}(x) = \begin{cases} 1 & \text{if } w^T x + b \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.2}$$

A layer is defined as multiple neurons connected in parallel. Each neuron transforms the same input x, hence a layer is capable of learning multiple features of x. To learn non-linear patterns, hidden layers are introduced between input and output layers, connecting outputs of the previous layer to inputs of the next layer. This forms a network of neurons and is defined as an artificial neural network[1], or neural network (NN) for short. A deep neural network (DNN) is an NN with many hidden layers.

[1] Finetuning parameters in a multilayered NN involves backpropagation. This requires ϕ to be differentiable, unlike the perceptron. Commonly used activation functions in MLPs are ReLU and sigmoid.

A multilayer perceptron (MLP) is a fully-connected (FC) deep neural network. We provide a mathematical definition as follows.

Definition 2.2 (Multilayer perceptron)**.** *A multilayer perceptron with L hidden layers is defined as:*

$$h^{(0)} := x \tag{2.3}$$

$$h^{(k)} = \phi_k\left(\left(w^{(k-1)}\right)^T h^{(k-1)} + b^{(k-1)}\right) \qquad k = 1,\dots,L \tag{2.4}$$

$$\hat{y} = \phi_{out}\left(\left(w^{(L)}\right)^T h^{(L)} + b^{(L)}\right) \tag{2.5}$$

MLPs are capable of learning complex non-linear functions. However, their number of parameters scale exponentially with dimension of feature space $X \ni x$. This causes the model to be susceptible to overfitting and not generalising well to unseen data. Furthermore, like all NNs, training an MLP involves updating parameters $w^{(k)}, b^{(k)}$. Therefore, the amount of training data required grows exponentially, a phenomenon known as the "curse of dimensionality, dimcurse". We explore how shortcomings of MLPs are addressed in Section 2.1.3.

2.1.2 Feature Learning

An important stage in deep learning is representing raw data as numerical vector inputs. This is referred to as feature extraction. In the context of image classification, a $w \times h$ bitmap image with 3 colour channels can be unambiguously described by a vector $v \in \mathbb{R}^{3wh}$ that can act as input to train an MLP. However, such a representation is sensitive to absolute intensity: the same image may look different during daytime and nighttime. The representation is not discriminative, and information on neighbouring pixels is lost when a 2D image is flattened to a vector.

Up until 2012, feature extraction pipelines would be predominantly manually engineered by researchers[26]. The workflow is shown in Figure 2.1.

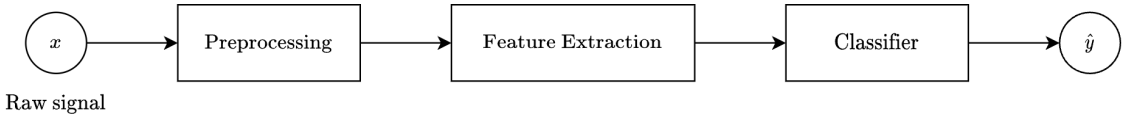


Figure 2.1: Schematic process for applying feature extraction to raw data and feeding the extracted representation into a machine learning classifier.

An example of a feature extraction algorithm is the scale-invariant feature transform[27] (SIFT) for detecting and describing generalisable, local features in images. This descriptor is robust to scaling, rotation and intensity. We give a brief overview of SIFT in Appendix A.

Recent advancements in computational speed have facilitated practical use of deep feed-forward networks like residual neural networks (ResNet) as discussed in Section 2.1.3. Feature extraction algorithms like SIFT are being dropped in favour of such networks which perform automated feature learning in initial layers. This allows numerical data such as vector of pixel intensities to be fed directly into the classifier model without a feature extraction algorithm as a precursory step.

2.1.3 Convolutional Neural Networks

A convolutional neural network (CNN) is a deep learning architecture that tackles overfitting by encoding certain properties such as local connectivity and weight sharing to reduce the number of parameters in the network. This approach is effective for image classification tasks[28].

Bitmap images are encoded as a 2D grid of pixels. The colour of each pixel is encoded as three channels that store red, green and blue intensity. This third-order tensor must be reshaped to a 1-dimensional vector to input into an MLP. However, flattening a matrix to a vector loses information on some neighbouring pixels, a shortcoming pointed out in Section 2.1.2.

Dive into Deep Learning (Zhang and others)[26] offers an elegant analogy: consider the puzzle game "Where's Wally?" where readers are challenged to find Wally, donned in his bobble hat, striped red shirt and round glasses. A critical observation is that Wally has this distinctive appearance regardless of where he appears in the image. This observation generalises to image classification tasks, for example, on detecting whether an image contains a certain object. This is the intuition behind two principles of CNNs: translation invariance and local connectivity.

- **Translation invariance** - This refers to how the network responds similarly to the same patch, regardless of its position in the input image. In CNNs, this is achieved through the convolutional layers, which uses a fixed-size kernel to scan over the input image and extract features.

  Translation invariance is an important principle as it allows the model to generalise well to unseen data. This is because it learns to recognise objects regardless of where they appear in the image. That said, CNNs are not completely translation invariant, as the position of objects are sometimes significant, e.g. eyes with respect to a face.

- **Local connectivity** - The locality principle states that a neuron in a given layer is influenced only by a small number of neurons in the previous layer that are close to it. This is enforced in the early layers of a CNN so the model can learn patterns that are local in nature.

Local connectivity substantially reduces the number of parameters in a CNN in comparison to an MLP. In the early convolutional layers, each neuron f has a small receptive field. Let z denote the output of the previous layer. The input x is the output of a small $m \times m$ local region in z. As the $m \times m$ kernel shifts across z, the kernel weights remain the same. This is known as weight sharing and further reduces the number of parameters.

The first instances of CNNs achieving state-of-the-art performance in image classification include AlexNet[29] (Figure 2.2) which won the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2012. Advancements in CNNs reveal network depth is an important factor in performance in non-trivial visual recognition tasks[30, 31]. However, very deep models suffer from vanishing or exploding gradients[32, 33]. Some models address this by introducing intermediate normalisation layers[34] and shortcut connections[35, 36, 31]. In the subsequent section, we

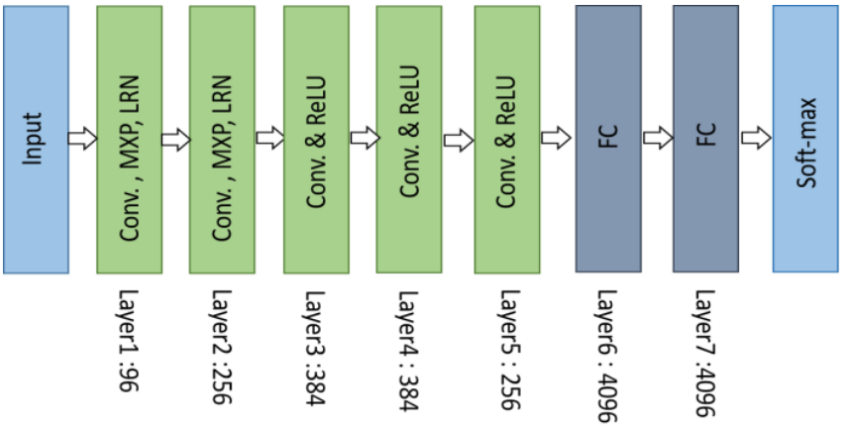


Figure 2.2: AlexNet comprises of convolution, max-pooling, local response normalisation and FC layer[29, p. 11]

discuss a state-of-the-art deep CNN model used in self-supervised learning algorithms: ResNet.

ResNet

The underlying principle of a residual neural network (ResNet) involves creating explicit references to let layers fit a residual mapping[37]. The idea is that it is easier for the model to optimise the residual mapping than a traditional unreferenced mapping. This is achieved through building blocks, as presented in Figure 2.3. We provide a formal definition as follows.

Definition 2.3 (Building Block)**.** *A building block connects the start and end of a stack of convolutional layers. Given input vector x, a building block is defined as:*

$$y = \mathcal{F}\left(x, \{W_i\}\right) + x \tag{2.6}$$

$\mathcal{F}\left(x, \{W_i\}\right)$ *is the residual mapping learned by the stack of layers.*

(2.6) *assumes input and output dimensions are equal. If not, we can either match the dimensions by performing a linear projection W_s or add zero entries padding.*

$$y = \mathcal{F}\left(x, \{W_i\}\right) + W_s x \tag{2.7}$$

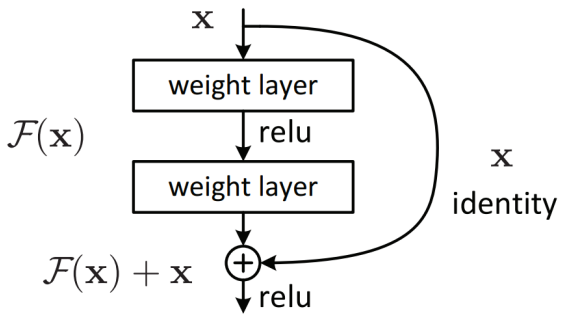


Figure 2.3: A building block serves as a shortcut connection in residual learning[37, p. 2].

Given a plain deep feedforward CNN model, a residual neural network is the same network with added building blocks to every few stacked layers. The building blocks serve as shortcut connections. These connections perform the identity mapping which does not increase parameter count nor computational complexity.

ResNet achieved state-of-the-art performance on ImageNet and won 1st place on the ILSVRC 2015 classification task. As a proof of concept, the original paper[37] explored a 1202-layer ResNet which trained with no optimisation difficulty and achieved good performance.

2.2 Self-Supervised Learning

Deep neural networks are developed as a scalable architecture that automates the labour-intensive process of manually engineering the feature extraction pipeline. Self-supervised learning is a paradigm motivated by an insufficient amount of labelled data.

Supervised learning problems involve training a model to learn a mapping between input and output space. Traditionally, this requires labelled examples: each data point is associated with a label, and the output space is the set of possible labels. Conversely, unsupervised learning aims to detect patterns within unlabelled data, such as clustering and dimensionality reduction.

A problematic scenario is when we want to tackle supervised learning problems like image classification with a lack of labelled examples and an abundance of unlabelled examaples. This

arises often in practice. For example, we can collect unlabelled data easily by filming a car journey, but lack of labels make this data infeasible to be used to train a semantic segmentation model for autonomous driving. One solution is to label data manually, but this is inefficient, prone to human error and not scalable.

Over recent years, there has been development and success in self-supervised learning: a machine learning paradigm that uses unlabelled data to learn useful representations for downstream tasks. Self-supervised learning frameworks such as SimCLR, MoCo and BYOL have outperformed their supervised learning counterparts[4, 12, 13] in downstream transfer learning (see Section 2.2.1). Recent success and development in self-supervised models include Bidirectional Encoder Representations from Transformers (BERT) NLP model used in Google's search engine[38] and OpenAI's Generative Pre-trained Transformer 3 (GPT-3) autoregressive model for completing text prompts and answering questions[39].

2.2.1 Pretraining and Transfer Learning

Pretraining is the process of performing initial training a machine learning model on one dataset. Downstream transfer learning involves taking this model and further training or finetuning it with a modified environment, for example, a different dataset or loss function. With initial pretraining, the model may learn generalisable features that can be useful for the specific downstream task. An example application is taking the pretrained GPT-3 model and finetuning it to specifically improve performance on sentiment analysis.

In Section 2.2, we mention lack of labelled examples as a common problem in supervised learning, but we may have an abundance of unlabelled examples. A workflow is to perform self-supervised learning using unlabelled data, then finetune the resulting model with supervised learning using the small amount of labelled data.

2.2.2 Data Augmentation

Self-supervised learning involves training a model to extract features. Ideally, the learned features are generalisable so the model is applicable to unseen data. As discussed in Section 2.1.3, invariance and locality are important principles in feature learning. The idea is to enforce models to learn generalisable features by applying label-preserving modifications to an image, which produces images that are similar but not identical to each other. These modifications

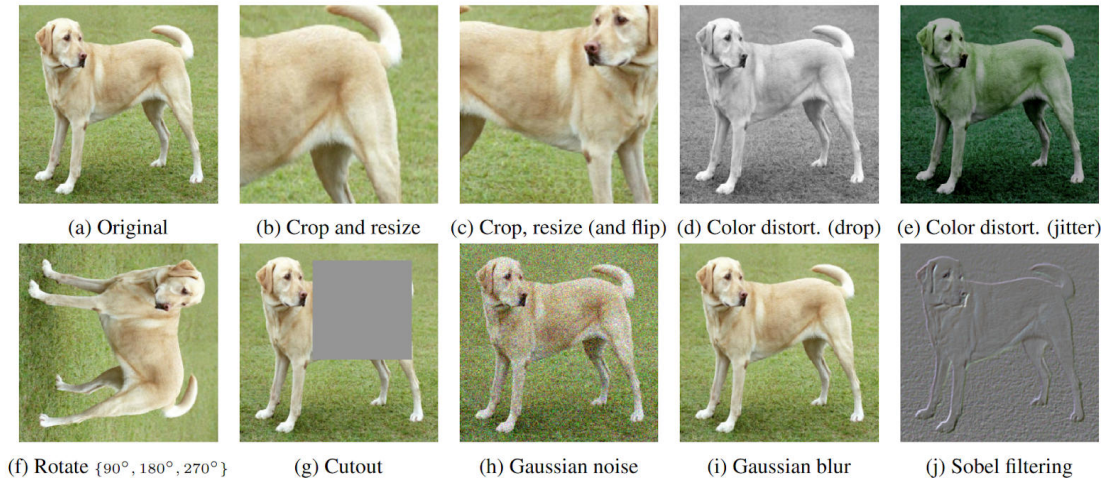


Figure 2.4: Popular data augmentations applied to self-supervised learning methods.[4, p. 4] (Original image cc-by: Von.grzanka)

are known as data augmentations. The models are trained to learn similarities between the augmented images. Figure 2.4 presents a non-exhaustive list of data augmentations that can be applied in self-supervised learning methods. In Section 2.2.4, we analyse augmentations applied to SimCLR, a self-supervised contrastive learning framework.

Data augmentation is not novel to self-supervised learning. In traditional machine learning techniques, it is used to artificially increase the size of a dataset. This is used extensively in training convolutional neural networks[40].

2.2.3 Contrastive Learning

Self-supervised models can be described as generative or representative[41]. Generative models like GPT-3 aim to produce diverse, realistic outputs, while representation learning like BERT aim to learn useful features for downstream tasks (see Section 2.2.1). We focus on two state-of-the-art representation learning techniques: contrastive and non-contrastive learning.

Contrastive learning stems from the idea that pairs of examples sharing similar features (positive examples) are close to each other in the embedded space, while dissimilar pairs (negative examples) are further apart. This idea is prevalent in NLP and computer vision[42]. Examples of contrastive learning methods include SimCLR and Moco, both of which provide competitive results on image classification[12, 4] (see Figure 2.5).

One of the challenges of contrastive learning is preventing dimension collapse. This is the phenomenon where embedded vectors in the learning representation spans a lower-dimensional subspace of the entire embedding space. This also describes the case where all embedded vectors collapse to a single point, referred to as a "complete collapse". Most contrastive learning methods have used techniques like momentum encoders[12] and large batch sizes[4] to prevent collapse, but in turn makes the training process very computationally intensive.

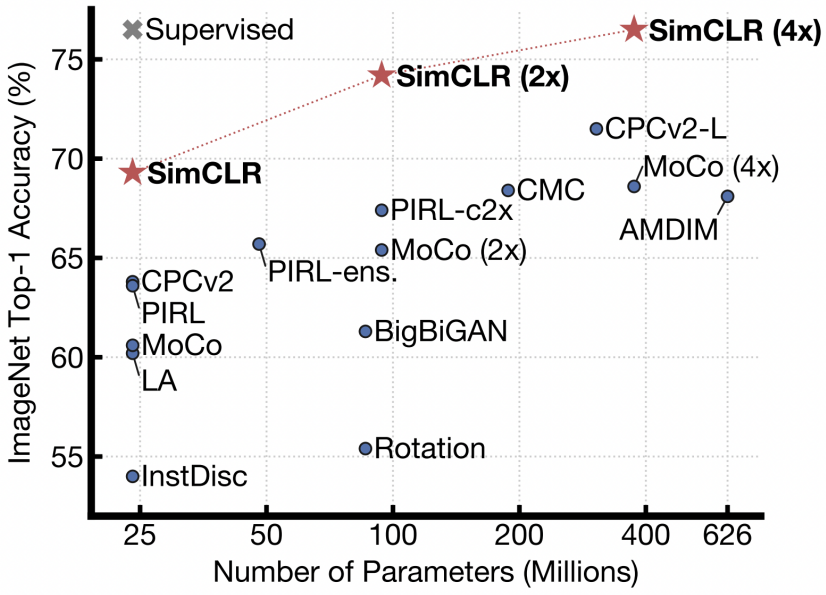


Figure 2.5: ImageNet top-1 accuracy of linear classifiers with self-supervised pretraining on ImageNet, against supervised ResNet-50 (grey cross). SimCLR achieves 76.5% accuracy while SimCLRv2 (not shown in figure) achieves 79.8% accuracy[5]. As of May 2023, SimCLRv2 has the runner up top-1 accuracy for self-supervised image classification on ImageNet with ResNet encoder, topped only by ReLICv2 with 80.6% accuracy[3]. [4, p. 1]

In Section 2.2.5, we discuss non-contrastive learning as an alternative to contrastive learning, which does not rely on explicit positive-negative pairs.

2.2.4 SimCLR

SimCLR is a contrastive learning method that aims to learn useful representations of images by comparing augmented data via a convolutional neural network[4]. The CNN is trained to recognise similarities between data points that are transformed versions of the same input image, as well as dissimilarities between data points derived from different input images. Using this contrastive method, the network can learn to extract useful representations that can be used in downstream tasks, as discussed in Section 2.2.1. SimCLR has reached state-of-the-art performance in image classification, as presented in Figure 2.5.

Overview of Architecture

Figure 2.6 visualises the architecture of SimCLR. We begin by applying a fixed augmentation sequence twice to produce a pair of augmented images. The transformed images are different, as the augmentations involves randomness. The fixed sequence is a very important hyperparameter and it is discussed in Section 2.2.4. The resulting images are each fed into a base encoder (convolutional neural network) which outputs a feature vector (learned representation). This is then fed into a small MLP (i.e. the projection head) that outputs a reduced embedded vector. The CNN is trained to minimise the distance of the embedded vectors of the positive pair while maximising the distance of the vectors of negative pairs.

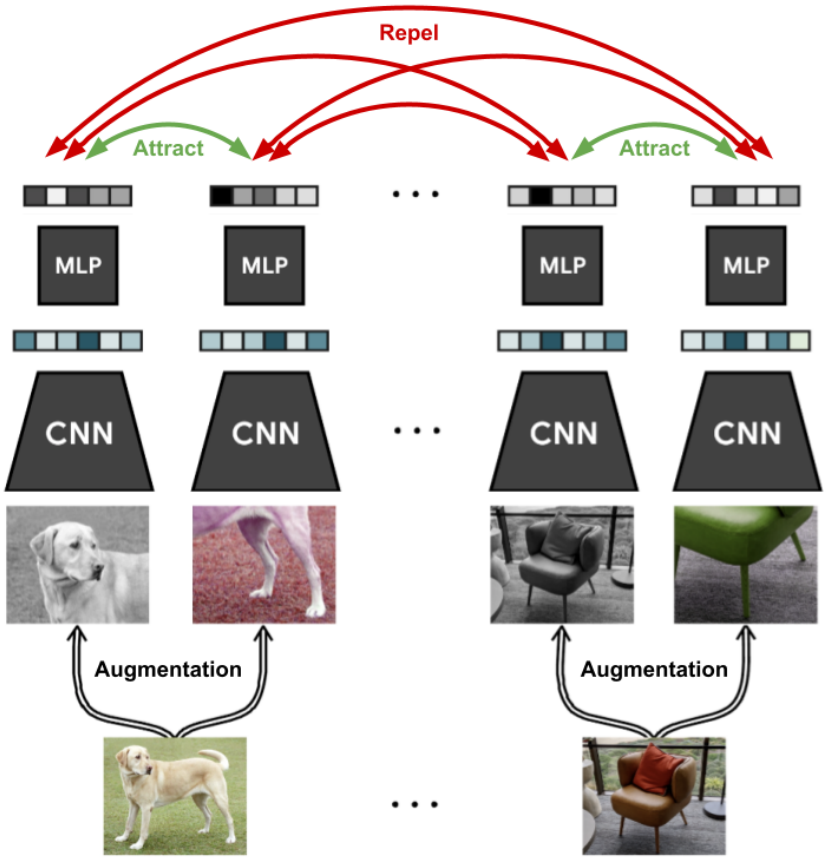


Figure 2.6: A friendly illustration of SimCLR architecture comprised of a base encoder, projection head and effect of image augmentations[43].

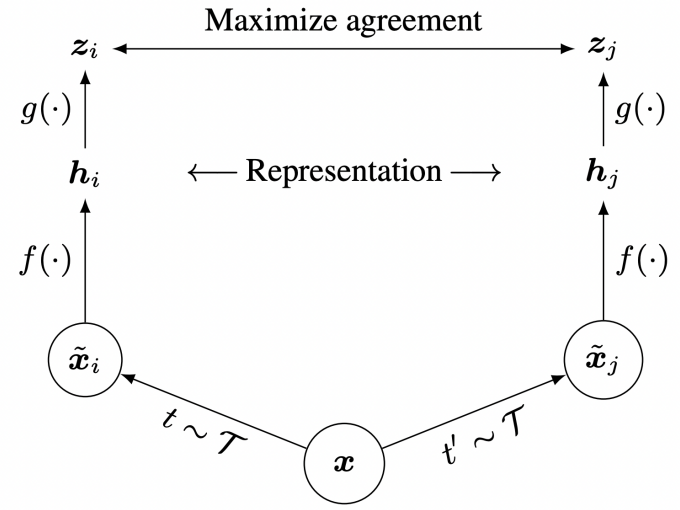


Figure 2.7: A simple framework for contrastive learning of visual representations. Two data augmentation operators are sampled from the same augmentation sequence $t \sim T$ and $t' \sim T$ and applied to initial image x. The base encoder $f(\cdot)$ is trained to maximise agreement using contrastive loss alongside a projection head $g(\cdot)$.[4, p. 2]

SimCLRv2[5] adopts the same architecture and explores deeper and wider base encoders and a slightly larger projection head. SimCLRv2 also applies semi-supervised learning by performing distillation[2] with unlabelled examples after supervised downstream learning.

Figure 2.7 encapsulates the architecture as follows. We adopt the same notation in this paper.

$$x := \text{original image} \tag{2.8}$$
$$\tau := \text{fixed augmentation sequence} \tag{2.9}$$
$$\tilde{x}_i := \text{result of applying } t \in \tau \text{ to } x \tag{2.10}$$
$$\tilde{x}_j := \text{result of applying } t' \in \tau \text{ to } x \tag{2.11}$$
$$f(\cdot) := \text{base encoder} \tag{2.12}$$
$$h_k := f(\tilde{x}_k) \quad \forall k \in \{i, j\} \quad \text{i.e. feature vector of augmented images} \tag{2.13}$$
$$g(\cdot) := \text{projection head} \tag{2.14}$$
$$z_k := g(\tilde{x}_k) \quad \forall k \in \{i, j\} \quad \text{i.e. embedded vector to apply contrastive loss} \tag{2.15}$$

Data Augmentation

In Section 2.2.2, we discuss the role of data augmentation in self-supervised learning and present a list of techniques in Figure 2.4. The choice of augmentations is a very important hyperparameter for SimCLR as it directly affects the representation space and what patterns the model learns.

The original paper[4] presents SimCLR with the following fixed sequence: crop and resize, colour distortion then Gaussian blur. The same sequence is adopted for SimCLRv2[5]. This sequence is applied on x twice to produce $\tilde{x}_i$ and $\tilde{x}_j$. It achieves strong performance, and the use of crop and resize in combination with colour distortion is crucial. We provide formal definitions[3] of these augmentations in Section 3.2.1. We give a short discussion below.

[2] Distillation is a technique to compress the knowledge in multiple trained models to a single model[44].

[3] The numbers have been slightly modified from the original paper to obtain better performance and stability for medical images.

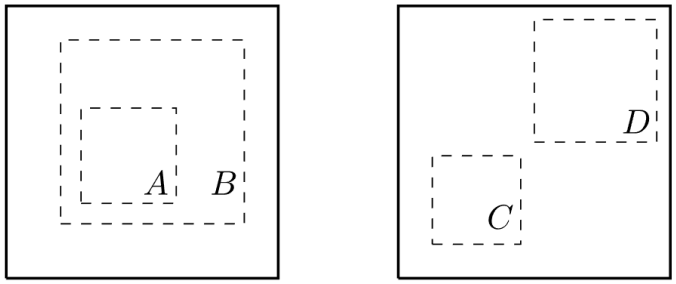


Figure 2.8: Possible results of performing crop and resize twice on the same image. One case is where A is a local view of B. The other case is where C and D are neighbouring views[4, p. 3].

Crop and resize takes a subset of the original image, then expanding it to its original size. This results in 2 scenarios as presented in Figure 2.8:

1. WLOG assume $\tilde{x}_i \geq \tilde{x}_j$ in size. $\tilde{x}_j$ is a local view of $\tilde{x}_j$. This forces the model to learn scale invariance to recognise other similarities between $\tilde{x}_i$ and $\tilde{x}_j$.
2. $\tilde{x}_i, \tilde{x}_j$ are neighbouring views. This is more challenging to learn, but if no other augmentations are performed, the model can exploit the similarity of colour space between $\tilde{x}_i$ and $\tilde{x}_j$[5].

Crop and resize is followed by colour distortion and Gaussian blur to prevent the model from exploiting colour space and learn more generalisable features. Examples of augmented images are shown in Figure 2.9.

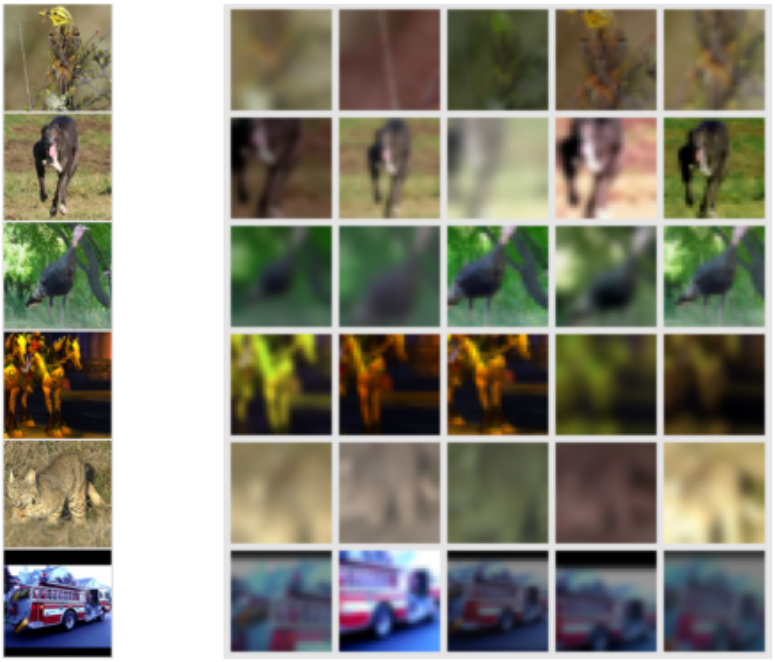


Figure 2.9: Examples of augmented images from the STL-10 dataset[45], applying random crop and resize, colour distortion and Gaussian blur.

Base Encoder: Convolutional Neural Network

The base encoder network $f(\cdot)$ consists of convolutional layers and acts as a feature extractor. SimCLR uses ResNet-50[4] while SimCLRv2 explores larger models like ResNet-128 and ResNet-152[5]. An overview of ResNet can be found in Section 2.1.3.

Projection Head: Multilayer Perceptron

The projection head $g(\cdot)$ is a nonlinear transformation that maps representations h_i, h_j to an embedded space where we compare their similarities. The original paper for SimCLR proposes a two-layer MLP with ReLU activation in the hidden layer, while SimCLRv2 extends g to three layers.

We provide some preliminary definitions before moving onto how SimCLR optimises parameter values through use of the projection head.

Definition 2.4 (Cosine Similarity)**.** *$sim(\cdot,\cdot)$ denotes the cosine similarity between two vectors.*

$$sim\Big(z_i, z_j\Big) = \frac{z_i^T z_j}{\Big\|z_i\Big\|\Big\|z_j\Big\|} \tag{2.16}$$

The InfoNCE loss[46] is based on NCE (Noise-Contrastive Estimation). It is proposed as a loss function for representation learning frameworks. We provide a formal definition as follows.

Definition 2.5 (InfoNCE)**.** *Given a set $Z = \{z_1, \ldots, z_{2N}\}$ of $2N$ samples or N pairs of samples, with one positive pair and $N-1$ negative pairs, we optimise* (2.17)*, where τ is a temperature scalar hyperparameter.*

$$l_{i,j} = -\log \frac{\exp\Big(sim\Big(z_i, z_j/\tau\Big)\Big)}{\sum_{k=1}^{2N} \mathbb{1}_{[k\neq 1]} \exp\Big(sim\Big(z_i, z_k/\tau\Big)\Big)} \tag{2.17}$$

SimCLR applies the InfoNCE loss with batch size N, which compares the similarities of the augmented pair of images represented as vectors z_i, z_j and contrasts them with representations of other augmented images in the same batch by performing softmax.

The original paper proposes dropping g during downstream transfer learning as the representations z tend to perform worse than h. SimCLRv2 uses a larger g and experiments suggest part of g can be preserved for downstream learning.

Limitations

SimCLR requires a large batch size (256 to 8192, sizes far exceeding commercial GPUs[4]) to guarantee enough negative pairs during batch training. As consequence, pretraining takes a very long time. Google uses Tensor Processing Units (TPUs) to train a ResNet-50 SimCLR model with batch size of 4096[4].

Alternatively, representations h from different batches can be merged together during training in an attempt to reduce batch size. However, f gets updated every minibatch, so some representations become outdated and suboptimal. This phenomenon is known as inconsistent representation generation. MoCo combats this problem by introducing a memory network, which we discuss in Section 2.2.5. Note that SimCLRv2 adopted this memory mechanism which yielded a 1% improvement in accuracy.

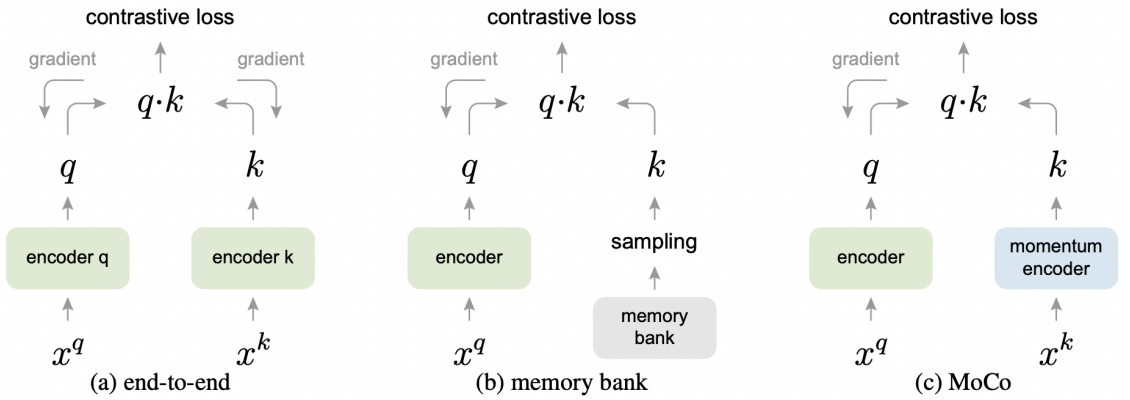


Figure 2.10: MoCo uses a momentum encoder to combat inconsistent representation generation. We compare this with two existing mechanisms. (a): end-to-end update by backpropagation requires a large batch size to be effective. (b): memory bank[47] consists of representations of all image samples in the dataset and no backpropagation is needed. A momentum update is used to maintain some consistency. (c): momentum contrast is memory efficient and capable of supporting billion-scale data[12, p. 3]

2.2.5 Various Works

MoCo: High Performance with Small Batch Size

Momentum Contrast[12, 48] (MoCo) is a contrastive learning method that produces competitive results without the need for large machines during training[4]. Like SimCLR, MoCo uses a ResNet query encoder denoted as f_q. It also uses a momentum encoder (a.k.a. key encoder, f_k) that is updated via linear interpolation of the two encoders, as defined in (2.18). This allows generated tokens (representations) from f_k to be consistent across multiple batches, making a small batch size feasible.

$$\theta_k \leftarrow m\theta_k + (1-m)\theta_q \tag{2.18}$$

where θ_k are parameters for f_k, θ_q are parameters for f_q.

Figure 2.10 presents the architecture of MoCo.

MoCo uses batch normalisation as in standard ResNet. As consequence, a common scenario is when the parameters of f_k are similar to f_q, since they are fed the same input data, which can potentially cause the intra-batch communication to leak information[12]. To combat this, MoCo shuffles the sample order in each mini-batch for f_k before distribution among GPUs and after encoding. The sample order for f_q is not shuffled.

The projection head of MoCo involves using InfoNCE loss (2.17) with cosine similarity (2.16). MoCo v2[48] involves using stronger augmenetations during pretraining and adopts the 2-layer MLP setup from SimCLR.

MoCo has a lower accuracy than SimCLR on the ImageNet top-1 accuracy benchmark (71.1% accuracy[48] using ResNet-50 query encoder compared to SimCLR's 76.5%) but offers a high-performing alternative to organisations that lack the necessary hardware to train a SimCLR model.

[4] Due to small batch size, MoCo v2 baselines for ImageNet can run on an 8-GPU machine[48].

Non-Contrastive Learning: BYOL

In Section 2.2.3, we discuss contrastive learning methods which compares positive and negative examples. Negative pairs introduce inconsistency problems, which can be resolved with large batch sizes, a memory bank or momentum encoder. The other state-of-the-art self-supervised representation learning method is non-contrastive learning, which uses positive examples only.

Bootstrap your own latent (BYOL)[13] is a non-contrastive method that uses two neural networks that interact with each other to predict images with different augmentations applied.

Figure 2.11 visualies the architecture of BYOL. We refer to the two networks as "online" and "target" network.

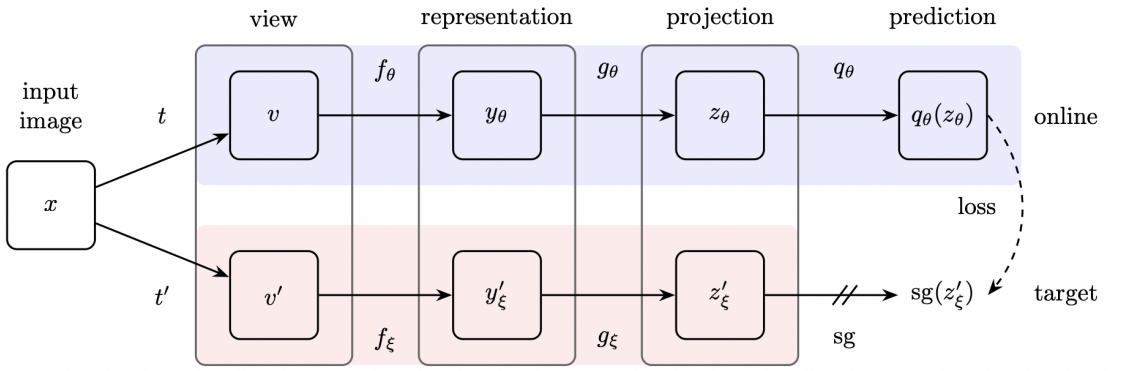


Figure 2.11: BYOL consists of an online and target network, each of which comprises of an encoder f and a projector g. The online network also consists of a predictor q.[13, p. 4]

The online network has weights θ and consists of an encoder f_θ, a projector g_θ and a predictor q_θ. The target network has the same architecture with the exception of a predictor, and has weights ξ, defined as an exponential moving average of θ as defined by (2.19).

$$\xi \leftarrow \tau\xi + (1-\tau)\theta \tag{2.19}$$

where $\tau \in [0,1]$ is the decay rate.

BYOL produces a pair of augmented views v, v' from an image. v is fed into the online encoder f_θ which outputs a representation y_θ, then fed into g_θ to output z_θ. z_θ is fed into q_θ. The same process is applied to v' in the target network. The online predictions and target projections are normalised as follows:

$$\overline{q_\theta}(z_\theta) = \frac{q_\theta(z_\theta)}{\left\| q_\theta(z_\theta) \right\|_2} \tag{2.20}$$

$$\overline{z'}_\xi = \frac{z'_\xi}{\left\| z'_\xi \right\|_2} \tag{2.21}$$

BYOL uses a mean-squared error loss as defined by (2.22). Furthermore, v' and v are separately fed to the online and target network respectively to compute $\tilde{\mathscr{L}}_{\theta,\xi}$.

$$\mathscr{L}_{\theta,\xi} = \left\| \overline{q_\theta}(z_\theta) - \overline{z'}_\xi \right\|_2 \tag{2.22}$$

$$\mathscr{L}^{\mathrm{BYOL}}_{\theta,\xi} = \mathscr{L}_{\theta,\xi} + \tilde{\mathscr{L}}_{\theta,\xi} \tag{2.23}$$

The parameters θ are updated as defined by (2.24).

$$\theta \leftarrow \mathrm{optimiser}(\theta, \nabla_\theta \mathscr{L}^{\mathrm{BYOL}}_{\theta,\xi}, \nu) \tag{2.24}$$

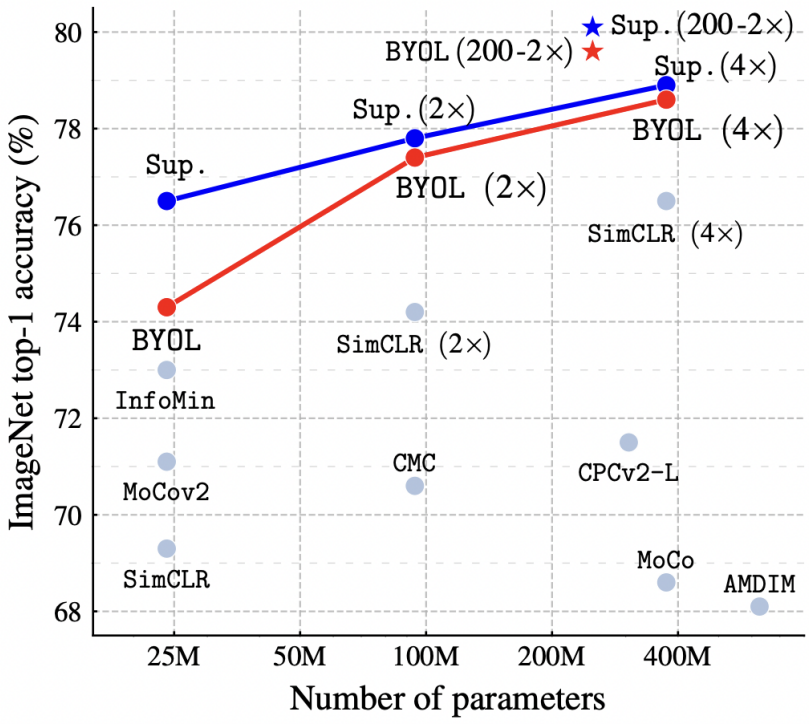


Figure 2.12: Performance of BYOL on ImageNet top-1 accuracy using ResNet-50 and ResNet-200 (2×).

where ν describes the learning rate.

Only f_θ is kept in downstream transfer learning. Figure 2.12 compares the performance of BYOL on ImageNet top-1 accuracy against state-of-the-art methods. BYOL achieves 79.6% accuracy, slightly lower than SimCLRv2's 79.8% (not marked on figure).

Non-contrastive learning has been criticised for having an abundance of non-collapsed global optima in the loss objective that may not learn the correct ground truth features[42].

2.3 Self-Supervised Learning in Medical Imaging

Section 2.2, describes concepts and techniques in self-supervised learning. Similar principles are applied for medical image classification, with existing techniques tuned to accommodate for characteristics such as low contrast and noise, greyscale colour space and large dimensions. Images are analysed for diagnostic and therapeutic purposes. The objective is to train models capable of being deployed in computer-aided diagnosis[9, 7, 8, 49, 50] (CADx) to support doctors in interpreting medical images and early detection of malignant lesions. We discuss various works that extend self-supervised frameworks to tackle such challenges.

2.3.1 Various Works

In 2019, Chen et al.[15] proposed an approach for medical imaging classification based on context restoration, where pairs of randomly chosen image patches are swapped within an image. This preserves intensity distribution but changes spatial information. A CNN is trained to restore the original image and learns useful representations. This method achieves good performance under lack of data (285 labelled images for brain tumour segmentation[15, p. 9]) compared to existing pretraining methods. However, evaluation is performed on limited data modality. Namely, performance of classification tasks is evaluated on fetal 2D ultrasound only,

localisation on abdominal CT and segmentation on brain MR images. Furthermore, the paper did not verify whether these tasks outperformed their supervised learning counterpart.

In 2019, Zhou et al.[16] proposed an approach for 3D medical imaging tasks using an encoder-decoder architecture and applying augmentations such as non-linear intensity transformation and local pixel shuffling. The models aim to learn rich representations with limited annotated data. This method achieves strong empirical results, surpassing performance of models trained from scratch. The paper focuses on chest CT. Future work involves evaluating this method for other data modalities. The paper adapts this approach to 2D versions, and is evaluated to offer performance similar to supervised pretrained models on ImageNet. Further work is required to determine the feasibility of applying this method to 2D medical images.

In 2021, Azizi et al.[18] proposed an approach using SimCLR and Multi-Instance Contrastive Learning (MICLe) where initial pretraining uses unlabelled natural images before using unlabelled medical images. However, evaluation only consists of two medical imaging categories (dermatology and chest scans) and evaluation under out-of-distribution (OOD) data is not considered.

2.3.2 Big-Data Training

In 2022, Ghesu et al.[14] proposed a methodology for self-supervised learning on large medical imaging datasets (over 105,000,000 multi-modality images consisting of X-ray, CT, MR, US) based on contrastive learning and online feature clustering[51].

A schematic overview is presented in Figure 2.13. The framework adopts existing contrastive learning techniques like applying augmentations and using ResNet[14, p. 6] as a learning model. The optimisation criterion is formed by swapping projected representations based on cross-entropy loss, as defined by (2.25).

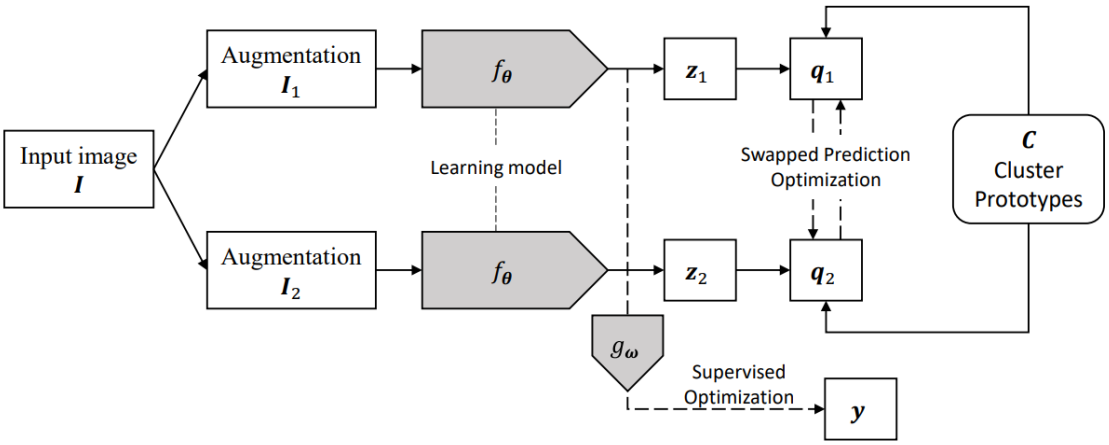


Figure 2.13: Architecture consists of augmentation operators, learning model f_θ which outputs features z_1, z_2. The features are mapped to their cluster assignments q_1, q_2 and used for optimisation.[14, p. 3]

$$\mathscr{L}(z_1, z_2) = -\sum_i q_2^{(i)} \log \frac{\exp \frac{1}{\tau} z_1^T c_i}{\sum_j \exp \frac{1}{\tau} z_1^T c_j} - -\sum_i q_1^{(i)} \log \frac{\exp \frac{1}{\tau} z_2^T c_i}{\sum_j \exp \frac{1}{\tau} z_2^T c_j} \tag{2.25}$$

where τ is the temperature parameter and $\{c_1, \ldots, c_K\}$ is the set of cluster prototype vectors that each pair (z_1, z_2) can be assigned to, for some hyperparameter K.

The paper offers two online clustering algorithms based on whether the training dataset consists of images from the same medical imaging category (i.e. single-modality) or multiple

categories (i.e. multi-modality). The goal is to estimate the visual representations learned by f_θ via assigning the projected vectors q to cluster codes. Similar clustering strategies are developed for unsupervised representation learning, for example, DeepCluster[52] uses k-means as optimisation criteria, which is further developed to scale effectively to large datasets[53, 54].

Data Augmentation

The following augmentation strategies are used: image rescaling, energy-based augmentation, linear and non-linear intensity rescaling, cropping. This has a similar intuition to the chosen augmentations for SimCLR as discussed in Section 2.2.4, with energy-based augmentation and intensity rescaling over colour distortion due to the prevalence of greyscale images in many medical imaging categories.

Energy-based augmentation is based on the image normalisation algorithm[55] developed for chest radiography. An image I is divided into energy bands $I^{(1)}, \ldots, I^{(B)}$ with Gaussian filtering. The normalised image with respect to the image crop Ω is calculated as:

$$\hat{I}(\Omega) = \sum_{i=1}^{B} \frac{\overline{e_i(I,\Omega)}}{e_i(I,\Omega)} I^{(i)} \tag{2.26}$$

where $\overline{e_i(I,\Omega)}$ is the arithmetic mean of the energy levels of a set of predefined reference images with respect to band $I^{(i)}$.

Performance Results

The proposed methodology yields results exceeding supervised learning and SimCLR in AUC performance (see Table 2.1) for lesion detection in chest radiographs. However, there is limited amount of experiments performed due to long training times (up to 14 days). Additional optimisation is necessary to make this method more efficient and scalable, which will enable further investigations to be carried out, for example, on different modalities.

	AUC Performance (LIDC-staged)			
	100%	50%	25%	10%
No pretraining	0.77	0.73	0.65	0.53
SimCLR	0.90	0.88	0.82	0.79
New methodology	**0.94**	**0.91**	**0.85**	**0.85**

Table 2.1: AUC performance for lesion detection using data from Lung Image Database Consortium (LIDC). Comparison of results with SimCLR using 100%, 50%, 25%, 10% of training data. [14, p. 8]

2.3.3 REMEDIS

In 2022, Azizi et al. proposed REMEDIS[17]: a unified representation learning framework for medical imaging. REMEDIS uses SimCLR (Section 2.2.4) for feature learning and builds on previous work[18], using pretraining on unlabelled natural images before using unlabelled medical image data. REMEDIS achieves strong relative improvement in multiple medical imaging categories (see Table 2.2). The team suggests that other contrastive models for feature learning may achieve similar performance results.

Data Augmentation

REMEDIS uses standard SimCLR augmentations, namely random cropping, colour distortion, rotation and random Gaussian blur.

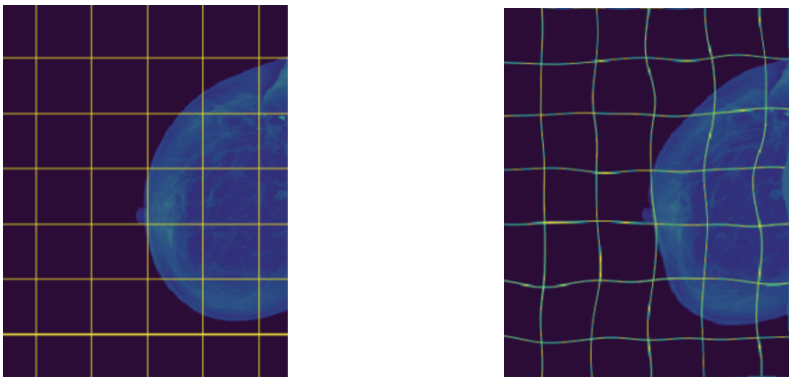
Task	Metric	Abs. Improvement	% Improvement
Dermatology	Top-3 Accuracy	0.026	2.9
Chest X-Ray	AUC	0.015	1.7
PSP	AUC	0.014	1.8
DME	AUC	0.098	11.5
PMD	AUC	0.034	4.7
Mammo. Classification	AUC	0.018	2.1

Table 2.2: In-distribution improvement between REMEDIS and baseline: ImageNet-1K supervised pretrained ResNet. PSP is Pathology Survival Prediction, PMD is Pathology Metastases Detection, DME is Diabetic Macular Edema.[17, p. 48]

Mammography and chest x-rays have greyscale colour space. For these images, REMEDIS also uses elastic deformation (Figure 2.14) and histogram equalisation (Figure 2.15) to reduce overfitting.

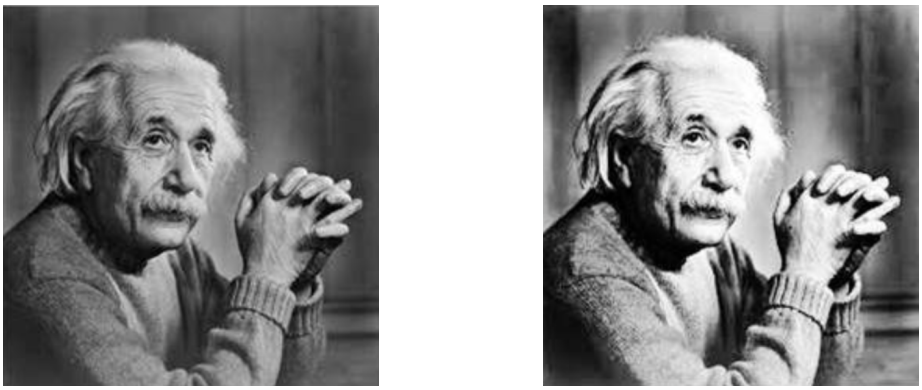
Conclusions

The paper demonstrates initial pretraining using non-medical images followed by additional pretraining using medical images can lead to improvements in performance. State-of-the-art contrastive learning methods like SimCLR can be used in the field of medical imaging, although



(a) Original (b) Deformed

Figure 2.14: Elastic deformation on a mammogram [56, p. 3]



(a) Original (b) New

Figure 2.15: Histogram equalisation enhances image contrast of Einstein [57, p. 9]

a few modifications like data augmentation choices are necessary to accommodate for medical imaging characteristics such as greyscale colour space.

The study uses retrospective data. Future research is needed to develop more compute-efficient learning methods.

Chapter 3

Standard SimCLR Setup

In this chapter, we adopt a SimCLR setup that performs well for classifying natural images[43] and investigate the extent to which this setup can be transferred to medical imaging. We begin by establishing some preliminaries, including data source, hyperparameter values and justifying ResNet-18 as our base encoder choice. We then describe our setup written in Python with PyTorch. We run this setup for colon pathology, dermatology and blood cells and consider two downstream environments for transfer learning: freezing the backbone[1] and not freezing the backbone. We compare our findings to baseline results obtained from a supervised learning context.

3.1 Preliminaries

3.1.1 Data Source

We source medical images from the MedMNIST database[20]. This database consists of lightweight, standardised biomedical images downsampled to 28×28 and includes binary, multi-class and multi-label classification tasks, as well as ordinal regression. We focus exclusively on 2D datasets.

The MedMNIST data points are labelled. We can create an environment that uses the labels of a subset of the available data to simulate settings with a lack of labelled data and an abundance of unlabelled data. Since we have availability of labelled data, we can also evaluate the effectiveness of SimCLR pretraining with large amounts of labelled data, allowing us to investigate the extent to which pretraining is necessary.

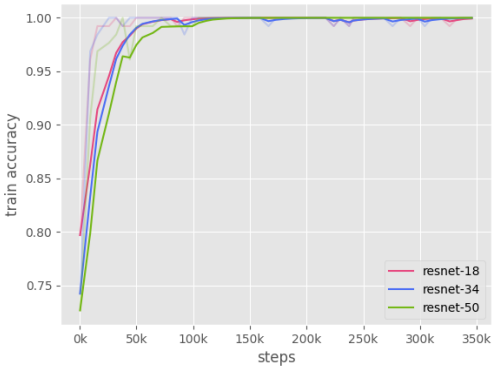
The class distribution of various datasets is presented in Appendix B.2.

3.1.2 Base Encoder Choice

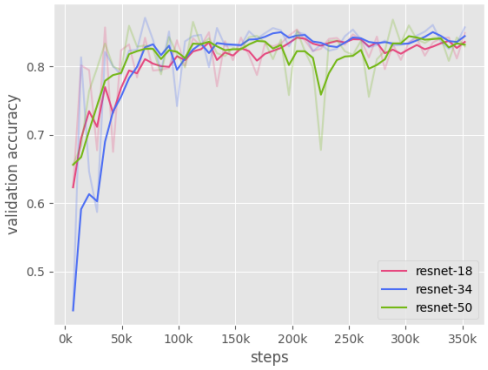
We choose ResNet as our encoder for image classification. To choose an appropriate depth, we perform supervised learning on various medical imaging modalities using depths supported by torchvision. Namely, the supported models are ResNet-18, ResNet-34, ResNet-50, ResNet-101 and ResNet-152, but we consider the three smallest models only, as the MedMNIST database is described as lightweight[20].

In Section 3.2.2, we explain that we will run experiments for colon pathology, dermatology and blood cells. For each category, we run the supervised baseline setup (as described in Section 3.2.4) with ResNet-18, ResNet-34 and ResNet-50. We present the results in Figure 3.1. The

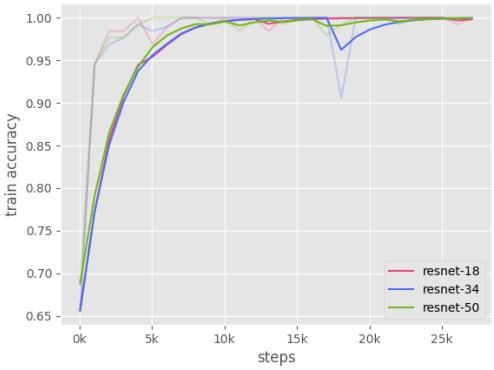
[1] The backbone in downstream tasks is the base encoder f in SimCLR pretraining.



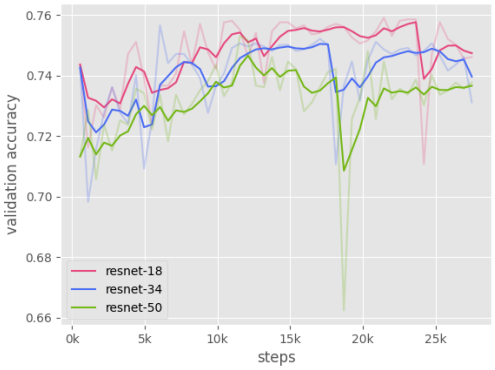
(a) Colon Pathology: Train



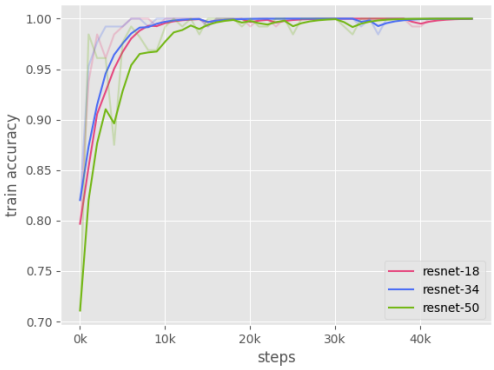
(b) Colon Pathology: Validation



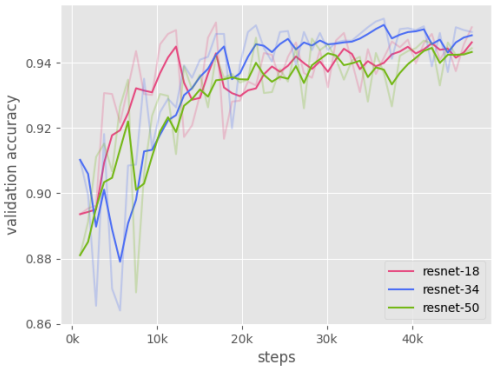
(c) Dermatology: Train



(d) Dermatology: Validation



(e) Blood Cells: Train



(f) Blood Cells: Validation

Figure 3.1: Comparison between top-1 accuracy of ResNet-18, ResNet-34 and ResNet-50 on classifying medical images by performing supervised learning with 100% of available labelled data from MedMNIST. For each category, validation accuracy between ResNet models interweave with each other when train accuracy reaches 100%, suggesting that larger depths do not increase performance. Note that lines are exponentially smoothed with $\alpha = 0.6$ (see Appendix D) and the true lines are semi-transparent.

results suggest that increasing depth of model does not result in better performance. We choose ResNet-18 as our base encoder.

3.1.3 Hyperparameter Tuning

SimCLR benefits from lots of pretraining with a large amount of unlabelled data[5]. It takes 12 hours to produce one pretrained model on a NVIDIA GPU cluster. We choose not to perform hyperparameter tuning as a grid search with cross-validation is infeasible within the time constraints of this project.

Instead, we adopt standard hyperparameter values for performing SimCLR in the context of natural images[43]. Full details can be found in Table 3.1.

3.2 Setup

We build a complete SimCLR framework in Python using PyTorch Lightning, supporting initial pretraining with unlabelled data and a downstream environment to finetune f[2].

Our initial approach involves adopting a standard setup for classifying natural images. We slightly modify the setup described in the original paper[4] and evaluate the extent to which this setup can be adopted for classifying medical images.

The framework consists of the following key classes:

- **SimCLRLM** - A PyTorch Lightning module for performing contrastive learning with SimCLR.
- **ContrastiveDownloader** - Manages data pipelining from MedMNIST. During training, applies augmentation sequence to a data point to generate a pair of augmented images.
- **LogisticRegressionLM** - A PyTorch Lightning module representing a logistic regression model. This model is appended at the end of f when performing transfer learning with a frozen backbone. Refer to Section 3.2.3.
- **ResNetTransferLM** - A PyTorch Lightning module representing a ResNet model. Used for transfer learning with an unfrozen backbone. Refer to Section 3.2.3.

3.2.1 Data Augmentations

We adopt the augmentation sequence from a setup[43] that achieved high performance on the STL-10[45] dataset of natural images. We refer to this sequence as τ_{nat}. τ_{nat} is slightly modified from the sequence described in the original paper[4], namely, we apply less intense colour distortion. These changes obtained better performance and stability when applied to a dataset with low brightness variance[43], a trait found in medical images.

We describe τ_{nat} below. For each augmentation, let f define the augmentation application and x be the input image. Let $x_{i,j}$ be the (i,j) pixel of x.

1. **Random horizontal flip** - There is a 50% chance the image gets horizontally flipped. Formally, let $c \in C \sim U(0,1)$ be random.

$$f(x) = \begin{cases} x' & c < 0.5 \\ x & \text{otherwise} \end{cases} \tag{3.1}$$

   where $x'_{i,j} = x_{w-i,j}$ and w is width of x.

[2] Base encoder

2. **Random crop-and-resize** - Take a rectangular subset of the image and expand it to the original dimensions. Formally, let $s \in S \sim U(0.08, 1)$ be random. Let w, h be width and height of x.

   Choose random $i \in w - sw$ and $j \in h - sh$. We start defining f as follows.

$$f(x_{0,0}) = x_{i,j}$$
$$f(x_{w,h}) = x_{i+sw,j+sh}$$

   The rest of the points are defined via linear interpolation. The output image is concisely expressed by (3.2).

$$f(x_{\alpha w,\beta h}) = x_{i+\alpha sw,j+\beta sh} \quad \alpha, \beta \in [0, 1] \tag{3.2}$$

3. **Random colour distortion** - Apply jitter to brightness, contrast, saturation and hue. Formally, let $c_1, c_2, c_3 \in C \sim U(0.5, 1.5)$ and $c_4 \in D \sim U(-0.1, 0.1)$ be random.

$$f(x) = h(s(c(b(x)))) \tag{3.3}$$

   where $b(x)$ changes the brightness of x to $c_1 x$, $c(x)$ changes the contrast of x to $c_2 x$, $s(x)$ changes the saturation of x to $c_3 x$ and $h(x)$ changes the hue of x by c_4.

4. **Random greyscale** - There is a 20% chance the image becomes greyscale. Formally, let $x_{i,j}$ have colour channel values $r_{i,j}, g_{i,j}, b_{i,j}$. Let h define a greyscale transformation.

$$h(r)_{i,j} = h(g)_{i,j} = h(b)_{i,j} = \frac{r_{i,j} + g_{i,j} + b_{i,j}}{3} \tag{3.4}$$

   Let $c \in C \sim U(0, 1)$ be random.

$$f(x) = \begin{cases} h(x) & c < 0.2 \\ x & \text{otherwise} \end{cases} \tag{3.5}$$

5. **Gaussian blur** - Blur the image using a Gaussian kernel. Formally, let $\sigma \in \Sigma \sim U(0.1, 2.0)$ be random. The kernel size is 9×9.

$$f(x) = h * x \tag{3.6}$$
$$h(i, j) = \frac{1}{2\pi\sigma^2} e^{-\frac{i^2+j^2}{2\sigma^2}} \tag{3.7}$$

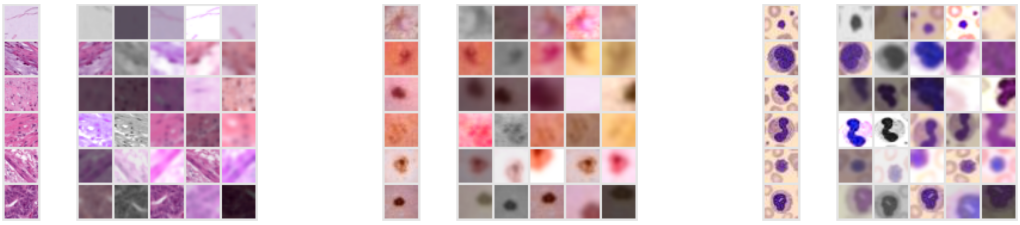
Examples of τ_{nat} applied to medical images can be found in Figure 3.2.

3.2.2 Experiments

Medical Imaging Modalities

τ_{nat} involves colour distortion. Furthermore, SimCLR requires a setting with a large amount of (unlabelled) data. Therefore, we will evaluate the standard SimCLR setup on the three MedMNIST categories that have an abundance of multi-coloured images: colon pathology, dermatology and blood cells.

Figure 3.2 presents samples of images from these categories (original and after augmentation).



(a) Colon Pathology (b) Dermatology (c) Blood Cells

Figure 3.2: Comparison between original and augmented MedMNIST samples. For each category, 6 original samples are shown on the left column. For each sample x, 5 augmented images derived from x are shown on the right.

Downstream Transfer Learning

We consider two approaches to transfer learning. Both approaches involve taking f and discarding g[3], then appending a linear layer to map representations h to output medical classes. Our first approach to freeze f, so only the parameters of the linear layer are tuned. This preserves feature representations learned during pretraining. The second approach is to not freeze f, so parameters of f are also finetuned.

During pretraining, parameters of f are trained using unlabelled data only. Since labelled data hold more information, logically we would want to utilise them to finetune encoder parameters. However, we argue that this setting may be sensitive with limited labelled images and the finetuned parameters may not generalise well. Therefore, we will evaluate performance of both approaches.

3.2.3 Architecture Overview and Implementation

The architecture consists of two stages. The first stage is to perform SimCLR pretraining on the entire training dataset with τ_{nat}. This stage is presented in Figure 3.3. We adopt pre-tuned hyperparameter values as described in Section 3.1.3. Details of our setup can be found in Table 3.1.

The second stage is to perform downstream transfer learning on f. As discussed in Section 3.2.2, we build two downstream environments: one with a frozen backbone and one with an unfrozen backbone. Both environments are depicted in Figure 3.4.

[3]Projection head

Batch size	256
Dimension of latent space	128
Temperature	0.07
Learning rate	5×10^{-4}
Optimiser	AdamW, $\lambda = 10^{-4}$
Scheduler	Cosine annealing, $\eta_{\min} = 10^{-5}$
Weight decay	10^{-4}

Table 3.1: Hyperparameters and other details of our SimCLR environment. See Appendix C for number of epochs.

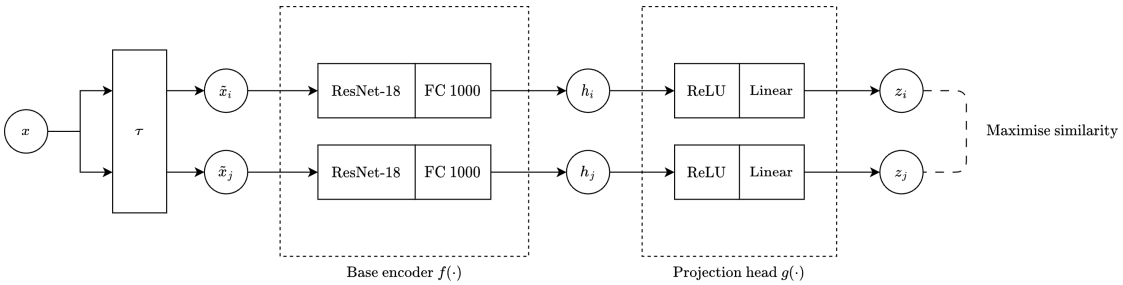
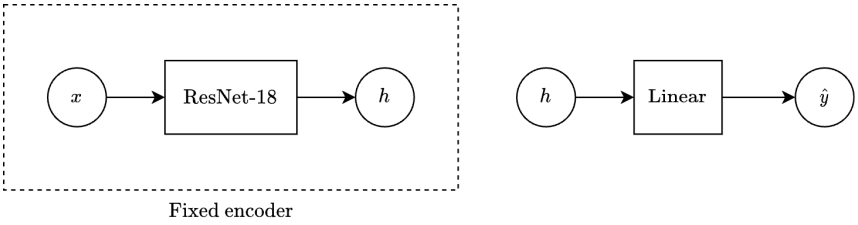
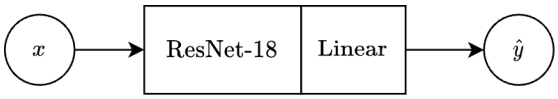


Figure 3.3: SimCLR pretraining pipeline. Given an image x, a pair of augmented images are created through a predefined augmentation sequence τ. The two augmented images are passed through a base encoder, then a projection head. We use ResNet-18 as the encoder. Fully connected layers consist of FC 1000, ReLU and linear layer. Note that FC 1000 is the last layer of ResNet-18.



(a) Setup with frozen backbone



(b) Setup with unfrozen backbone

Figure 3.4: Two downstream transfer learning environments. In both environments, the FC layer is removed from ResNet-18 and the projection head is replaced with a linear layer that maps representation h to predicted label $\hat{y}$.

Frozen backbone

We use a linear evaluation protocol where we attach a linear classifier to the end of the frozen base network and train it.

Since the encoder parameters remain fixed during transfer learning, we design this downstream environment to take in h. We can get f by deep cloning the SimCLR network and removing the projection head g. Then, the logistic regression model is simply a linear layer.

Details can be found in Table 3.2.

Unfrozen backbone

This downstream environment involves extracting f from SimCLR and redefining g as a linear layer. Given non-augmented image x, the predicted label is $g(f(x))$. g is attached at the end of f and parameters of both g and f get finetuned during training.

Details can be found in Table 3.2.

Both downstream environments use cross-entropy as loss function, as defined in (3.8).

[4] Milestones: 0.6 × max epochs, 0.8 × max epochs

	Frozen backbone	Unfrozen backbone
Batch size	64	128
Learning rate	0.001	0.001
Optimiser	AdamW, $\lambda = 10^{-4}$	Adam
Scheduler	MultiStepLR[4], $\gamma = 0.1$	None

Table 3.2: Hyperparameter and other details of both downstream environments.

$$H(p,q) = -\sum_{y \in C} p(y) \log q(y) \tag{3.8}$$

where p is the true probability (one-hot encoded), q is our predicted probability and C is the set of classes.

3.2.4 Baseline Environment

To measure increase in performance when using SimCLR pretraining, we design a baseline environment that performs supervised learning on the available labelled data only. This setup exactly matches the unfrozen backbone downstream environment described in Section 3.2.3, except with f being a newly initialised ResNet model.

3.2.5 Results

A comprehensive evaluation can be found in Section 7.3. We provide a brief summary here.

Our findings reveal that the standard SimCLR setup for natural images can yield significant improvement over baseline supervised learning when applied to medical imaging, in particular, when there is a deficiency of labelled images. With 100 labelled images, SimCLR pretraining gains 30.6% increase in accuracy for colon pathology and 15.3% increase for blood cells classification. A downstream environment with a frozen backbone outperforms the unfrozen backbone when there is a very small amount of labelled images, but the frozen backbone has better performance when there are more labelled images.

We observe models perform well for blood cells (up to 95.6% accuracy and 0.997 AUC) and colon pathology (up to 0.974 AUC). We observe SimCLR pretraining yield significant improvement despite the medical images being downsampled to 28×28.

With this setup, we have a less significant increase in performance for dermatology (2.2% increase in accuracy with 250 labelled images and 2.9% increase with 1000 labelled images). We propose the following explanations, which we explore in later sections: the augmentation sequence is suboptimal for dermatology, there is a lack of data and there is a dataset imbalance (see Table B.3).

Chapter 4

Exploring Augmentation Sequences

The choice of data augmentations are important in contrastive learning as it directly affects the features the encoder learns[4]. In this chapter, we investigate the impact of different augmentation sequences on model performance. We start by investigating the effect of random horizontal flip and random greyscale. We then propose a novel augmentation sequence that uses random histogram equalisation and random sharpness to enhance contrast of medical images. We run these setups for colon pathology, dermatology and blood cells. We compare our findings to results optained in Chapter 3.

4.1 Shorter Sequence

In Chapter 3, we investigate the extent to which a standard SimCLR setup for natural images can be applied to medical images. We use the augmentation sequence τ_{nat} as described in Section 3.2.1.

In this section, we use exactly the same setup as described in Chapter 3 (refer to Section 3.2.3 for overview of architecture), except that we do not use random horizontal flip and random greyscale. We argue that a greyscale filter causes information loss, while horizontal flip is not necessary as it does not contribute to altering gradient magnitude, nor does it help the encoder to learn local invariance. We refer to this shorter augmentation sequence as τ_{short}. For downstream tasks, we perform experiments with frozen backbone and unfrozen backbone. We compare our findings to results obtained from our standard setup to investigate whether these extra augmentations during pretraining improve classification performance.

4.1.1 Data Augmentations

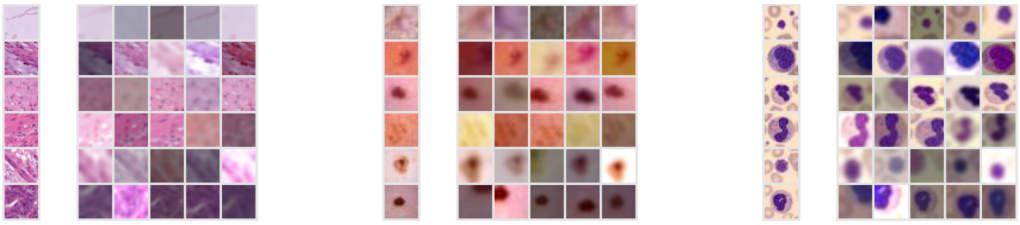
Refer to Section 3.2.1, except τ_{short} does not include **random crop-and-resize** and **random greyscale**.

Figure 4.1 presents samples of images from pathology, dermatology and blood cells (original and after augmentation).

4.1.2 Results

For full details and a more comprehensive evaluation, refer to Section 7.4.

We find that this setup yields similar performance levels to the standard setup for colon pathology and dermatology, but yields lower accuracy for blood cells (2.6% decrease with 250 labelled images). We conclude the use of random horizontal flip and greyscale allows richer representations for certain medical imaging modalities.



(a) Colon Pathology (b) Dermatology (c) Blood Cells

Figure 4.1: Comparison between original and augmented MedMNIST samples. For each category, 6 original samples are shown on the left column. For each sample x, 5 augmented images derived from x are shown on the right.

4.2 Novel Sequence

Up until this point, we have considered setups devised from natural image classification. In this section, we propose a novel sequence devised from considering traits prevalent in medical imaging, in particular, by modifying τ_{nat} to enhance contrast in images.

We use the same setup as described in Chapter 3 (refer to Section 3.2.3 for overview of architecture), except that we use our novel augmentation sequence, which we refer to as τ_{nov}. We compare our findings to results obtained from our standard setup[1].

4.2.1 Data Augmentations

τ_{nov} is adapted from τ_{nat} (the latter is described in Section 3.2.1). We start by applying random horizontal flip, random crop-and-resize, random colour distortion, random greyscale and Gaussian blur. Then, we further apply random histogram equalisation followed by random sharpness.

Random colour distortion is applied with $c_1, c_2, c_3 \in C \sim U(0.8, 1.2)$ and $c_4 \in D \sim U(-0.04, 0.04)$ as defined by (3.3). We choose to lower the effect of colour distortion as histogram equalisation also alters the gradient of colour space. The rest of the adopted augmentations are applied with the same parameters.

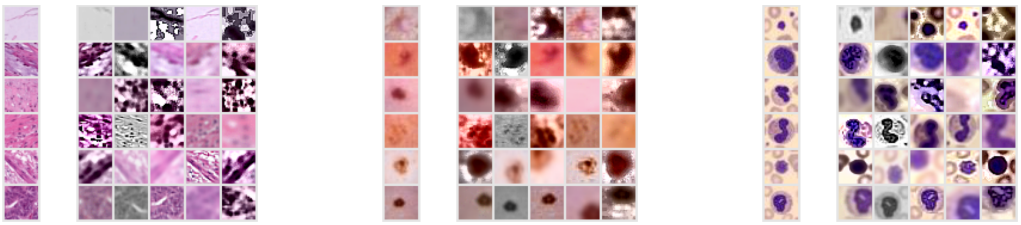
We provide a brief intuition of our proposition before describing histogram equalisation and sharpness in more depth. Compared to natural images, some medical imaging modalities have low contrast and noise. Gaussian blur is kept to smooth out noise. Random horizontal flip and random greyscale is kept as our investigation in Section 4.1 suggest that they may improve performance during pretraining. Histogram equalisation is introduced to tackle low contrast. To introduce variety, we considered two options.

1. Perform histogram equalisation on a random subset of the image.
2. Perform histogram equalisation with p chance.

We opted to perform histogram equalisation with 50% chance, since a previous augmentation involved cropping the original image. Nonetheless, the former option is worth investigating in future works. To address the case where no equalisation is performed, we introduce sharpness to increase brightness contrast.

[1] In the evaluation, we conclude that random horizontal flip and random greyscale are important augmentations, so we favour τ_{nat} over τ_{simple}.

Figure 4.2 presents samples of images from pathology, dermatology and blood cells (original and after augmentation).

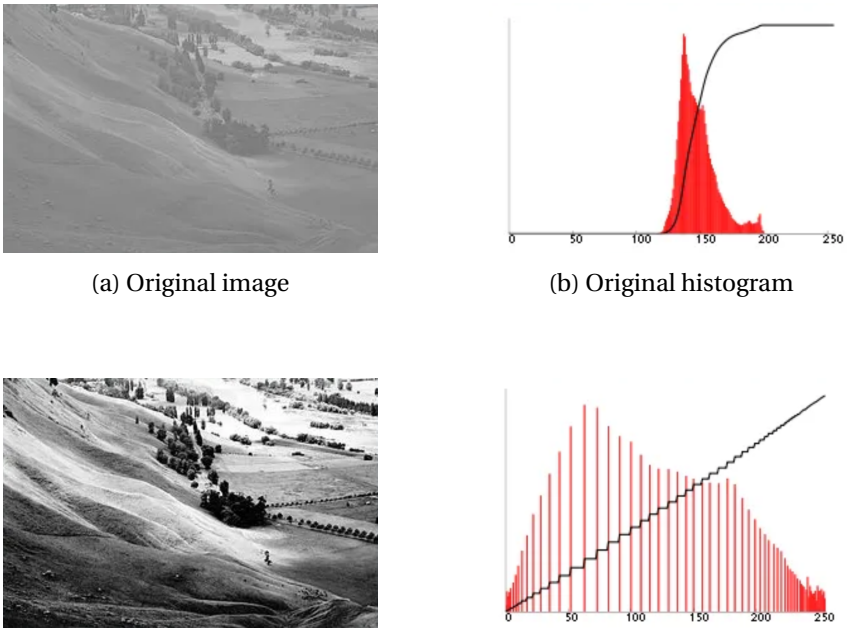


(a) Colon Pathology (b) Dermatology (c) Blood Cells

Figure 4.2: Comparison between original and augmented MedMNIST samples. For each category, 6 original samples are shown on the left column. For each sample x, 5 augmented images derived from x are shown on the right.

Histogram Equalisation

Histogram equalisation is an image processing method to enhance contrast. We start by considering greyscale images only. The intuition behind the method is to apply a transformation to each pixel intensity so that the plot of the cumulative distribution function (CDF) forms a straight line, indicating that pixel intensities are spread out. Figure 4.3 presents an example. We provide formal definitions and formulations below.



(a) Original image (b) Original histogram

(c) Resulting image (d) Resulting histogram

Figure 4.3: Comparison to showcase the effect of applying histogram equalisation to greyscale image [58]

Definition 4.1 (Histogram)**.** *The histogram of a greyscale digital image is a histogram formed from the number of pixels with intensity ϕ for each $\phi \in [0, L-1]$.*

We consider RGB images and their greyscale counterparts, so $L = 256$.

Definition 4.2 (Histogram Equalisation)**.** *Let g define equalisation application and x be the input*

greyscale image. Let $x_{i,j}$ be the intensity of the (i,j) pixel of x. Let h be the normalised histogram of x.

$$h_\phi = \frac{\sum_{x_{i,j} \in x} \mathbb{1}\left(x_{i,j} = \phi\right)}{number\ of\ pixels\ in\ x} \quad \forall \phi \in [0, L-1] \tag{4.1}$$

Then, g is defined by (4.2).

$$g(x_{i,j}) = floor\left((L-1)\sum_{\phi=0}^{x_{i,j}} h_\phi\right) \tag{4.2}$$

In practice, histogram equalisation is generally applied to low-contrast greyscale images to increase contrast[59]. It is a non-linear process and applying it directly to an RGB image disrupts the colour distribution. Therefore, given an RGB image, we first convert it to YC_bC_r which separates the colour space into a luma signal Y and chroma components C_b, C_r. We perform histogram equalisation on the intensity plane Y before converting the resulting image back to RGB.

τ_{nov} involves a 50% chance of applying histogram equalisation. Formally, let f define the augmentation application and g as defined in (4.2).

$$f(x) = \begin{cases} g(x) & c < 0.5 \\ x & \text{otherwise} \end{cases} \tag{4.3}$$

Sharpness

Let x be the input image. Define a smoothing kernel as follows.

$$h = \frac{1}{13}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 5 & 1 \\ 1 & 1 & 1 \end{bmatrix} \tag{4.4}$$

We can adjust the sharpness of x by applying the smoothing kernel and linear interpolating between the original and smoothed image.

$$g(x) = (h * x)(1-\alpha) + x\alpha \tag{4.5}$$

Note that $*$ is the convolution operator as defined in Appendix A.1. Since $h * x$ is the smoothed image, applying g with any α value larger than 1 results in a sharpened image.

τ_{nov} involves applying sharpness with α randomly chosen between 1 and 10 uniformly.

4.2.2 Results

For full details and a more comprehensive evaluation, refer to Section 7.5.

Our findings reveal that applying SimCLR with our novel augmentation sequence τ_{nov} result in small, consistent improvement in classification accuracy over using the original sequence τ_{nat}. In particular, it performs well for colon pathology when there is a small amount of labelled images: 2.9% increase in accuracy with 100 labelled images and 2.8% increase with 250 labelled images.

Chapter 5

Adapting to Lack of Data

In this chapter, we investigate the effectiveness of SimCLR pretraining and transfer learning on a small dataset that lacks both labelled and unlabelled data. We start by investigating the retina fundus dataset and attempt to transfer pretrained colon pathology features to retina fundus models. We then address dataset imbalance in dermatology by performing undersampling. We compare our findings to a set of baseline results (including supervised learning) and evaluate whether our proposed workflows improve model performance.

5.1 Setup

We adopt the setup described in Section 4.2[1] and make adaptations to our workflow pipeline which we describe in Section 5.1.2.

We extend our SimCLR framework in PyTorch Lightning as described in Section 3.2 to support these new experiments.

5.1.1 Data Source

We use the retina fundus dataset sourced from MedMNIST database (see Section 3.1.1). Figure 5.1 presents samples of images from this category (original and after applying τ_{nov}).

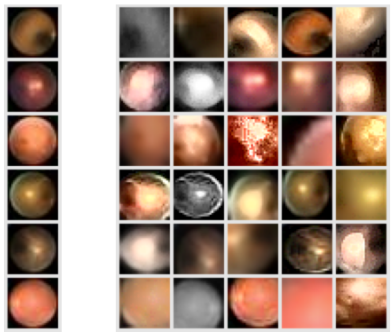


Figure 5.1: Comparison between original and augmented retina samples from MedMNIST. 6 original samples are shown on the left column. For each sample x, 5 augmented images derived from x are shown on the right.

[1]The setup in Section 4.2 is adopted from Chapter 3 but uses a novel set of augmentations τ_{nov} (see Section 4.2.1).

The class distribution of this dataset is presented in Table B.5. The training set consists of 1080 images only, significantly lower than categories we used for previous experiments.

5.1.2 Adaptations

We propose 2 setups that accommodate for lack of data.

1. **Pretrain on different dataset** - Perform initial SimCLR pretraining on a different dataset with lots of unlabelled examples. Then, perform downstream learning with labelled examples from the category that the model designed for. In this case, we perform initial pretraining on the colon pathology dataset with 89,996 training examples before finetuning with the retina fundus dataset.

2. **Pretrain on different dataset then further pretrain on specialised dataset** - Perform initial SimCLR pretraining on a different dataset with lots of unlabelled examples (i.e. colon pathology). Then, perform additional SimCLR pretraining with the specialised dataset (i.e. retina fundus). Finally, perform downstream learning with labelled examples (i.e. retina fundus).

Transfer learning have been explored in the past, by performing initial pretraining on a large set of natural images followed by further pretraining on a large set of medical images[17, 60]. We investigate whether this workflow is viable when there is a lack of images in a specialised category. The idea is that the base encoder f learns features that are generalisable when applied to different image modalities.

5.1.3 Baseline Environments

We compare results optained from our proposed modifications in Section 5.1.2 to 2 baseline environments.

The first baseline environment is to perform supervised learning on the available labelled data only. This is described in Section 3.2.4.

The second baseline environment is to perform SimCLR pretraining on the full dataset of retina images followed by downstream transfer learning on the labelled retina images. We perform downstream experiments for both frozen and unfrozen backbone. Essentially, we make zero changes to our SimCLR setup and apply it to a setting where we lack both labelled and unlabelled data.

5.1.4 Results

For full details and a more comprehensive evaluation, refer to Section 7.6.

We find that our first proposed setup (pretraining on different dataset) with a frozen backbone during downstream learning performs best, consistently outperforming the best baseline results and achieving up to 3.2% increase in classification accuracy. We conclude SimCLR pretraining on a different dataset is a viable option when presented with a lack of data on the specialised dataset.

5.2 Addressing Data Imbalance

The dermatology dataset provided by MedMNIST is heavily imbalanced. Table B.1 presents class distribution of dermatology samples. Out of 7 classes, 66% of the samples are labelled as *melanocytic nevi* (benign moles).

In Chapters 3 and 4, we perform SimCLR pretraining and downstream learning, and observed the dermatology models had less significant improvement over supervised baseline compared to colon pathology and blood cells. We posit that this may be due to data imbalance, so in this section, we investigate the effect of balancing the dataset by performing undersampling in downstream tasks.

Since undersampling the dermatology dataset causes a very limited amount of data, we have a similar setting to the retina dataset. Therefore, we take the adaptations described in Section 5.1.2 for a setting with lack of data and perform experiments for dermatology. As example, one setup involves taking the pretrained colon pathology model (using τ_{nov}) and performing downstream learning with 175 dermatology images (25 from each class).

5.2.1 Results

For full details and a more comprehensive evaluation, refer to Section 7.6.

We find that our setups perform poorly and do not outperform the models trained using the non-undersampled dermatology dataset. We posit that either features learned from colon pathology do not transfer well to dermatology, or that undersampling is not an effective solution. Further experiments are needed to provide more concrete insights.

Chapter 6

Adapting to Greyscale Images

In previous chapters, we investigate the effectiveness of SimCLR pretraining on medical images of colour. In this chapter, we investigate greyscale datasets consisting of medical scans. We outline our setup, modifications and results below.

6.1 Setup

We start by taking our setup described in Section 4.2. This involves using our novel augmentation sequence τ_{nov} for colour medical images. We make a necessary adaptation to this sequence as described in Section 6.1.1.

We use the tissue cells and retinal OCT datasets sourced from MedMNIST database (see Section 3.1.1).

6.1.1 Data Augmentations

τ_{nov} retains the use of colour distortion as a technique to alter the gradient of colour space. The core intuition is to force the encoder to learn more generalisable features of images during pretraining.

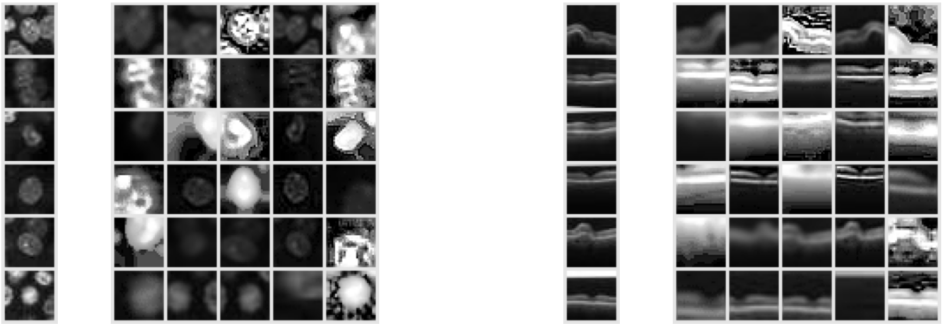
It is possible to apply colour distortion to greyscale images as our pipeline involves converting images to RGB format. However, this introduces new, meaningless information to augmented images. This may cause the encoder to learn extraneous features rather than existing, generalisable features which goes against the core intuition of applying colour distortion. Therefore, we choose to remove colour distortion from τ_{nov} for greyscale images. We hypothesise that histogram equalisation is sufficient in altering the gradient of intensities. We refer to this modified sequence as τ_{grey}.

Figure 6.1 presents samples of images from tissue cells and retinal OCT dataset (original and after applying τ_{grey}).

6.1.2 Results

For full details and a more comprehensive evaluation, refer to Section 7.7.

Our findings reveal that applying SimCLR with τ_{grey} is very effective and has significant improvement over baseline supervised metrics. Specifically, we observe over 8% increase in top-1 accuracy for tissue cells and over 10% increase for retinal OCT with limited labelled data settings. We conclude SimCLR pretraining is beneficial with respect to greyscale medical images.



(a) Tissue Cells

(b) Retinal OCT

Figure 6.1: Comparison between original and augmented MedMNIST samples. For each category, 6 original samples are shown on the left column. For each sample x, 5 augmented images derived from x are shown on the right.

In Section 7.3.1, we had seen SimCLR pretraining being very effective on the colon pathology dataset. Since colon pathology, tissue cells and retinal OCT datasets are by far the largest datasets used in this paper, we conclude the degree of improvement can be attributed to the size of the dataset as well as data modality.

Chapter 7

Evaluation

In this chapter, we begin by describing our evaluation protocol. We then analyse trends in model performance during training to verify that our pretrain and downstream environments are set up appropriately. Finally, we go through each setup and provide a quantitative and qualitative measure of performance of our models with respect to baseline models. We establish the extent to which our methods and proposed changes are successful.

7.1 Evaluation Protocol

Evaluation of downstream models involves computing top-1 accuracy and AUC ROC metrics for classification performance computed on prepartitioned test data. These metrics will be compared to baseline metrics derived from its supervised counterpart setup. As an example, consider a pipeline that involves pretraining ResNet on the entire colon pathology training dataset, then finetuning it on 1000 labelled images. The corresponding baseline setup is performing supervised learning on a newly initialised ResNet model with 1000 labelled colon pathology images.

Evaluation also involves an investigation of learned feature representations using embedding techniques such as PCA and t-SNE.

Note that even if we use a subset of the training dataset to train a model, evaluation is carried out using the entire test dataset.

7.1.1 AUC ROC

AUC ROC is the area under the receiving operating characteristic curve. In this report, we use the abbreviation AUC for this metric.

Definition 7.1 (True Positive (TP))**.** *Model prediction correctly indicates the presence of a condition/class.*

Definition 7.2 (True Negative (TN))**.** *Model prediction correctly indicates the absence of a condition/class.*

Definition 7.3 (False Positive (FP))**.** *Model prediction incorrectly indicates the presence of a condition/class.*

Definition 7.4 (False Negative (FN))**.** *Model prediction incorrectly indicates the absence of a condition/class.*

The true positive rate (TPR) is $\frac{TP}{TP+FN}$. The false positive rate (FPR) is $\frac{FP}{FP+TN}$. The ROC curve is constructed by plotting TPR over FPR at thresholds from 0 to 1.

AUC is inherently a metric for binary classification tasks. For multi-class classification, we calculate AUC using a one-vs-rest scheme that compares each class against all other classes, then takes the mean as the metric. AUC takes range $[0,1]$ where 1 represents a perfect classifier.

We use AUC as a supplementary quantitative measure of performance with top-1 accuracy as some medical imaging categories like dermatology have dataset imbalance, which AUC is less affected by.

7.1.2 Principal Component Analysis

Principal component analysis (PCA) is a dimensionality reduction technique that maps a high-dimension dataset to a lower-dimension dataset while preserving most of the information in the original dataset.

We perform visual analysis on the set of learned representations H. To extract H, for pre-trained models we remove the projection head from f[1]. Then, we encode the entire dataset X[2] to output $H = f(X)$. For baseline supervised models, we remove the final linear layer from f then compute $H = f(X)$.

$H \in \mathbb{R}^{512}$ so we use PCA with 2 principal components to reduce H to $H' \in \mathbb{R}^2$.

7.1.3 t-SNE

t-distributed Stochastic Neighbour Embedding[61] (t-SNE) is a dimensionality reduction visualisation tool that analyses similarities between data points and minimises the KL-divergence score between the joint probabilities of the data and the embedding. t-SNE is non-deterministic: the same dataset yields different visualisations when run multiple times.

We extract H as described in Section 7.1.2. We first perform PCA with 50 principal components to reduce H to $H' \in \mathbb{R}^{50}$ to suppress noise, then perform t-SNE with default parameters (perplexity of 30) to reduce H' to $H'' \in \mathbb{R}^2$.

If no observable clusters or patterns emerge, we tune the perplexity, specifically we test for $p \in \{5, 10, 15, \ldots, 100\}$.

7.1.4 Silhouette Coefficient

The silhouette coefficient[62] measures how similar each data point is to its own cluster (cohesion) with respect to the other clusters (separation). It is a value between -1 and +1, where higher values indicate well-defined clusters. We formally define it below.

Definition 7.5 (Silhouette Coefficient)**.** *Consider a dataset X with K clusters. Measure cohesion by defining a as follows.*

$$a(i) = \frac{1}{|C_I| - 1} \sum_{j \in C_I, j \neq i} d(i, j) \tag{7.1}$$

where i is a datapoint, C_I is the cluster i belongs to and d is a distance metric.

Measure separation by defining b as follows.

$$b(i) = \min_{J \neq I} \frac{1}{|C_I|} \sum_{j \in C_J} d(i, j) \tag{7.2}$$

[1] Base encoder

[2] X refers to either the training set or test set. For example, in Figure 7.3, to perform PCA we use the training set to determine the principal components, then plot reduced representations of test data points.

For data point $i \in C_I$, define its silhouette as:

$$s(i) = \begin{cases} \frac{b(i)-a(i)}{max\{a(i),b(i)\}} & |C_I| > 1 \\ 0 & |C_I| = 1 \end{cases} \tag{7.3}$$

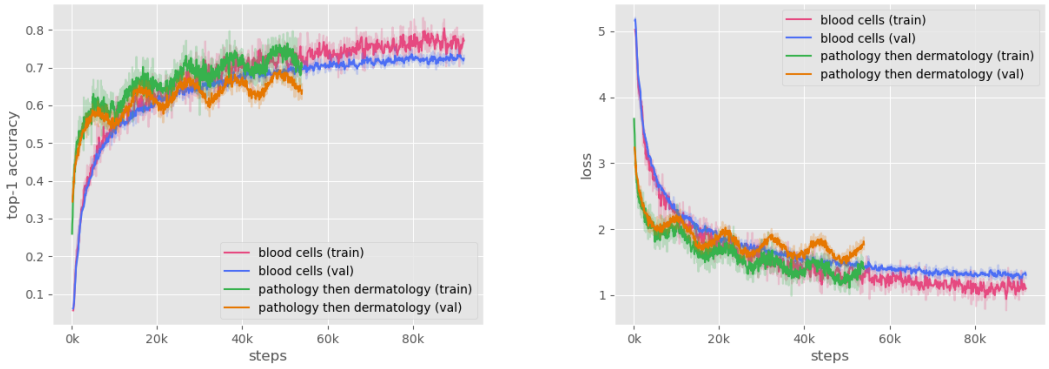
The silhouette coefficient of X is defined as the mean silhouette score for each i over all K clusters.

$$SC = \frac{1}{|X|} \sum_{i \in X} s(i) \tag{7.4}$$

We use the silhouette coefficient as a quantitative measure for clustering quality when it is difficult to empirically analyse learned representations.

7.2 Correctness of Training

In this section, we provide a general overview of the correctness in our pretrain and downstream environments by observing that the trend in accuracy and loss over time for our models are sensible.



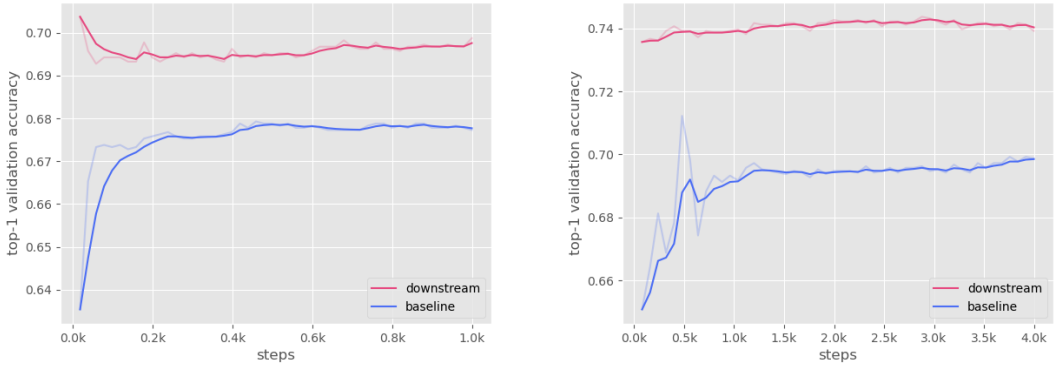
(a) Top-1 Accuracy

(b) Loss

Figure 7.1: Top-1 accuracy and loss for SimCLR pretraining using τ_{nov}. Comparison between pretraining with blood cells, and pretraining with colon pathology followed by further pretraining with dermatology. Accuracy is determined by whether the model correctly matches the positive pairs in a batch. Note that lines are exponentially smoothed with $\alpha = 0.6$ (see Appendix D) and the true lines are semi-transparent.

For pretrained models, the trend in accuracy and loss over time can be summarised with examples in Figure 7.1. In general, we observe that accuracy steadily increases over time, loss progressively decreases over time, training accuracy is higher than validation accuracy and training loss is lower than validation loss. This indicates a proper execution of pretraining.

There is one exception we encountered which Figure 7.1 also captures: pretraining with colon pathology followed by further pretraining with dermatology. The plots present accuracy and loss during the further pretraining stage. We observe an unexpected pattern where the accuracy and loss moves up and down smoothly over time like a sine wave. This phenomenon does not occur in any other pretrained models, nor can we offer an explanation. However, over the long term there is a general increase in accuracy and decrease in loss, suggesting that our setup works.



(a) 250 Labelled Samples

(b) 1000 Labelled Samples

Figure 7.2: Top-1 validation accuracy for dermatology. Comparison between supervised baseline and downstream learning with unfrozen encoder (pretrained using τ_{nov}).

For downstream environments, training accuracy usually reaches or approaches 100%. The trend in validation accuracy over time can be summarised with two examples presented in Figure 7.2. For a small amount of labelled samples, validation accuracy sometimes decreases at the start of training, but the model continues to perform better than its supervised baseline counterpart (as depicted by (a)). This phenomenon usually occurs with an unfrozen backbone. We posit that the backbone encoder is sensitive with a small number of samples.

More generally, we observe that validation accuracy steadily increases over time (as depicted by (b)).

7.3 Standard SimCLR Setup

7.3.1 Metrics

Table 7.1 presents metrics evaluated on the test dataset. We provide interpretations of our results below and perform qualitative analysis in Section 7.3.2.

Table 7.1 suggests that SimCLR pretraining yields a significant increase in performance for models trained on limited amount of labelled data for colon pathology and blood cells. With 100 labelled colon pathology images, pretraining gains 30.6% increase in accuracy over supervised baseline. With 1000 labelled images, pretraining gains 17.2% increase in accuracy. We observe for very limited amount of labelled data, freezing the backbone during downstream learning gives slightly better performance. An interpretation is that finetuning the backbone with a small dataset is sensitive and may cause detriment to the learned representations during pretraining. We previously mention this with Figure 7.2.

Despite medical images being downsampled to 28×28 (original sizes are detailed in Appendix B.1), the models perform well for colon pathology and blood cells. With 100 labelled images, the models achieve 0.939 AUC for colon pathology and 0.957 for blood cells. We previously raised a concern that medical images are large and of very high quality, resulting in long training times. Our findings reveal that an effective solution is to downsample the images during preprocessing.

SimCLR pretraining with the current setup yields much less increase in performance for dermatology. We propose the following interpretations.

1. The augmentation sequence τ is adapted from the original SimCLR paper, applied to natu-

	# Labelled Samples							
	100		250		1000		100%	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
Supervised baseline	0.395	0.780	0.559	0.839	0.658	0.879	0.857	**0.974**
Frozen backbone	**0.701**	**0.939**	**0.785**	**0.965**	**0.830**	**0.977**	-	-
Unfrozen backbone	0.685	0.909	0.762	0.921	**0.829**	0.950	**0.875**	0.964

(a) Colon Pathology

	# Labelled Samples							
	100		250		1000		100%	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
Supervised baseline	0.669	0.725	0.679	0.792	0.712	0.829	0.759	0.904
Frozen backbone	**0.677**	0.735	**0.700**	0.811	0.719	**0.873**	0.739	0.907
Unfrozen backbone	0.674	**0.756**	**0.701**	**0.839**	**0.741**	**0.871**	**0.787**	**0.925**

(b) Dermatology

	# Labelled Samples							
	100		250		1000		100%	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
Supervised baseline	0.618	0.890	0.728	0.919	0.852	0.973	0.952	**0.996**
Frozen backbone	**0.771**	**0.957**	**0.835**	**0.970**	**0.880**	**0.984**	0.911	0.992
Unfrozen backbone	0.732	0.934	0.796	0.953	**0.882**	**0.982**	**0.956**	**0.997**

(c) Blood cells

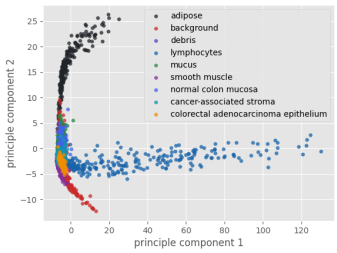
Table 7.1: Classification accuracy and AUC ROC of colon pathology, dermatology and blood cells. Performance of models trained with pretraining then downstream learning with frozen/unfrozen backbone is compared to performance of models from baseline supervised learning. Metrics are calculated using the entire test dataset provided by MedMNIST. Best-performing environments are bolded.

ral images. It may not perform well for some medical imaging modalities like dermatology. We propose a novel augmentation sequence in Chapter 4.

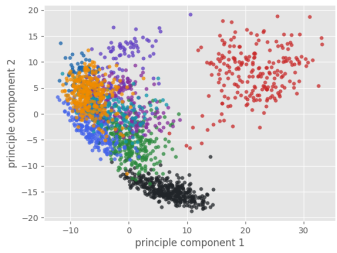
2. SimCLR requires lots of data to learn rigid, generalisable features[4]. We perform pretraining for colon pathology with 89,996 images and for blood cells with 11,959 images. However, only 7,007 images are used for dermatology. We propose solutions to lack of unlabelled data in Chapter 5.

3. The dermatology dataset is heavily imbalanced. Table B.1 presents the number of samples for each class. Over two thirds of dermatology samples labelled as *melanocytic nevi*. In Section 5.2, we attempt to balance this out by performing undersampling.

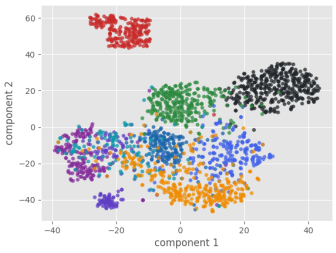
For all three medical imaging categories, metrics suggest pretraining gives small improvement on classification accuracy when 100% of the training dataset is used for downstream tasks. Performance for dermatology is improved most, with the downstream environment with frozen backbone gaining 2.8% increase in accuracy over supervised baseline. The frozen backbone performs worse, suggesting that features learned during pretraining were mostly overridden by features learned during downstream training. We conclude that SimCLR pretraining benefits model performance even when there is no deficiency in labelled data, although supervised



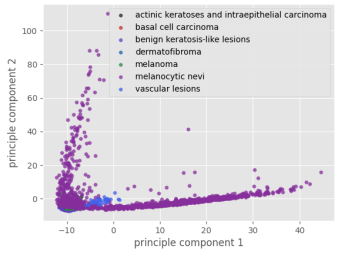
(a) Colon Pathology
PCA | Supervised Baseline



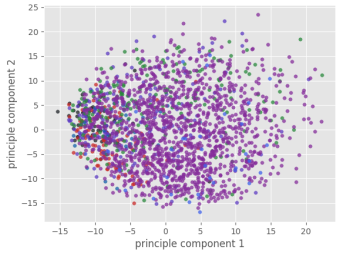
(b) Colon Pathology
PCA | SimCLR Pretraining



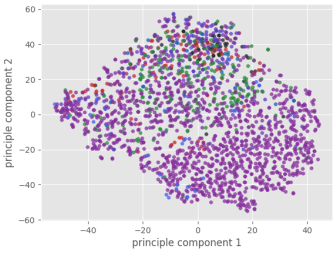
(c) Colon Pathology
t-SNE | SimCLR Pretraining



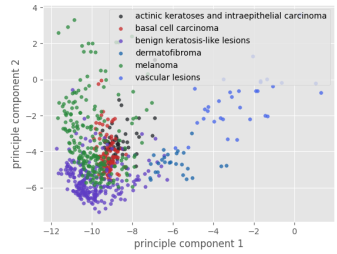
(d) Dermatology
PCA | Supervised Baseline



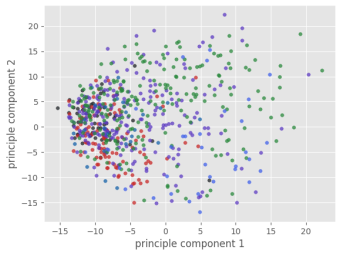
(e) Dermatology
PCA | SimCLR Pretraining



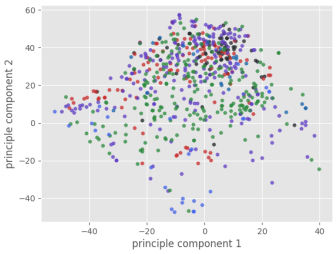
(f) Dermatology
t-SNE | SimCLR Pretraining



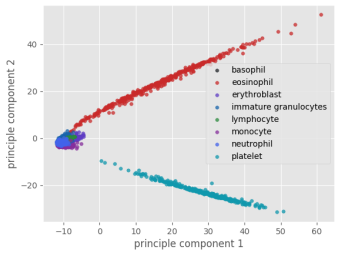
(g) Dermatology (Filtered)
PCA | Supervised Baseline



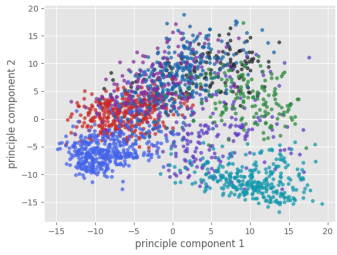
(h) Dermatology (Filtered)
PCA | SimCLR Pretraining



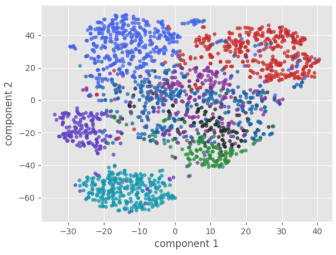
(i) Dermatology (Filtered)
t-SNE | SimCLR Pretraining



(j) Blood Cells
PCA | Supervised Baseline

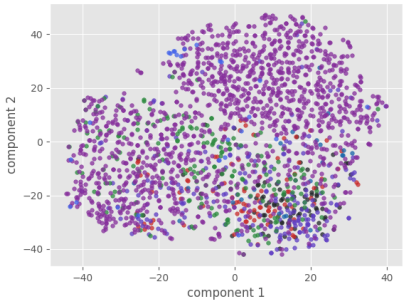


(k) Blood Cells
PCA | SimCLR Pretraining

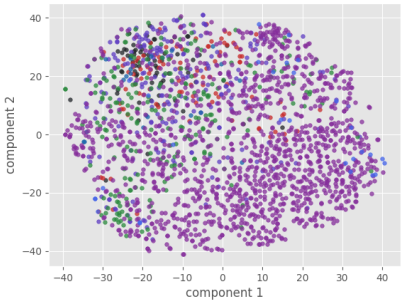


(l) Blood Cells
t-SNE | SimCLR Pretraining

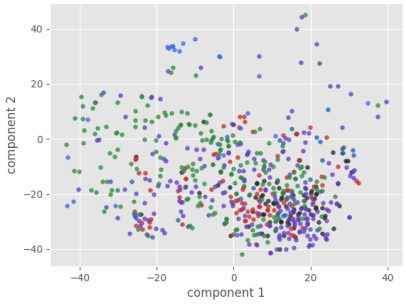
Figure 7.3: PCA and t-SNE of SimCLR pretraining and baseline supervised models on colon pathology, dermatology and blood cells. Principal components are determined using 100% of the training dataset. t-SNE components are determined using 100% training data for PCA, followed by 2000 reduced test data points for t-SNE. Each plot displays the reduced representations of 2000 data points from the test set. Note that (e) and (f) are the same plots as (c) and (d) respectively, but without *melanocytic nevi* data points to observe clusters with greater clarity.



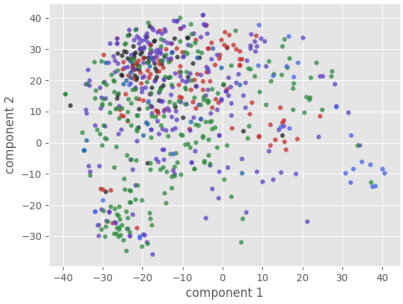
(a) $p = 40$ | Train Data



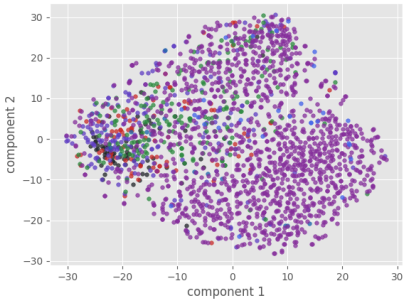
(b) $p = 40$ | Test Data



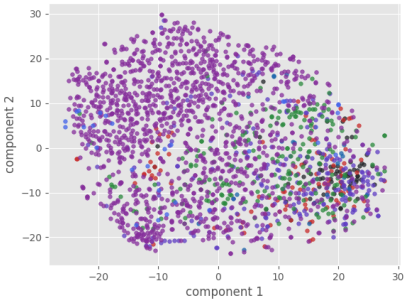
(c) $p = 40$ | Train Data (Filtered)



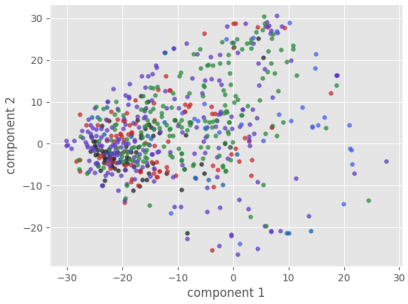
(d) $p = 40$ | Test Data (Filtered)



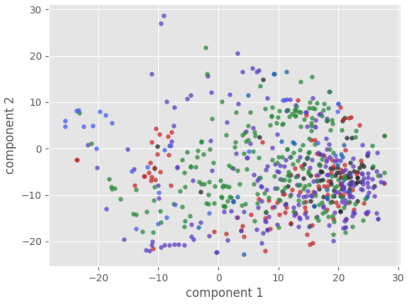
(e) $p = 70$ | Train Data



(f) $p = 70$ | Test Data



(g) $p = 70$ | Train Data (Filtered)



(h) $p = 70$ | Test Data (Filtered)

Figure 7.4: t-SNE of SimCLR pretraining on dermatology with various perplexities. Components are determined using 100% training data for PCA, followed by 2000 reduced data points (either train or test). Each plot displays the reduced representations of the data points. Refer to Figure 7.3 for labels.

learning performs well with much less training time required.

7.3.2 Learned Representations

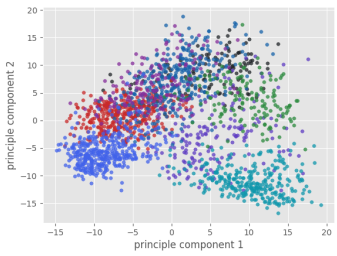
Figure 7.3 presents visualisations of reduced representations learned during pretraining using PCA and t-SNE. We outline our findings below.

When performing PCA, we observe distinct clusters formed for colon pathology and blood cells during pretraining. Clusters formed for the supervised baseline are even more distinct. Noticeably, some clusters for the baseline models are more spread out, whereas the space spanned by the clusters for pretrained models is more compact. Clusters also form when we perform t-SNE on pretrained models with the setup as described in Section 7.1.3.

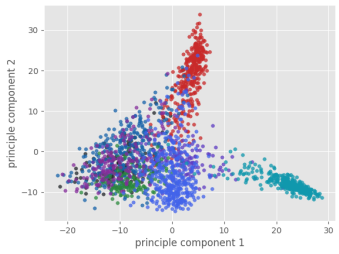
When performing PCA, there is empirically a weak formation of clusters for dermatology during pretraining. However, distinct clusters are formed for the supervised baseline, suggesting there are relationships between dermatology classes which the pretrained model struggled to learn. There is also a weak formation when we perform t-SNE on the pretrained dermatology model.

To investigate whether the pretrained dermatology model learned stronger patterns that were not detected by previous visualisations, we perform t-SNE with perplexities $p \in \{5, 10, 15, \ldots, 100\}$ using both training data and test data. We find the plots showcase very similar features, with a weak formation of clusters and majority of points clustered around a small subspace (with the exception of *melanocytic nevi*). Figure 7.4 presents the results for $p = 40$ and $p = 70$.

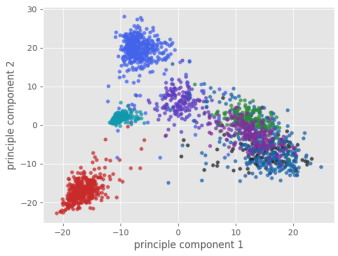
Finally, we analyse whether finetuning the encoder with limited labels detriment learned features. We consider blood cells modality, as freezing f has a 3.9% improvement in accuracy over finetuning f (see Table 7.1). Figure 7.5 presents visualisations of relevant setups. For visualisations produced by PCA, we measure clustering quality with Silhouette score. The pretrained model has SC = 0.153, and finetuning with 250 labels improves SC to 0.172. Despite presence of stronger clustering, the accuracy and AUC metrics indicate that finetuning with frozen f consistently outperforms finetuning with unfrozen f.



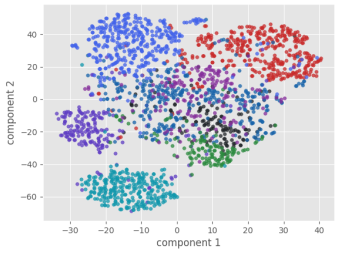
(a) PCA | Pretrain



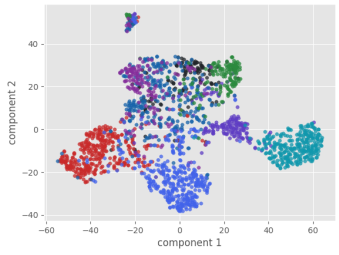
(b) PCA | Finetune: 250 labels



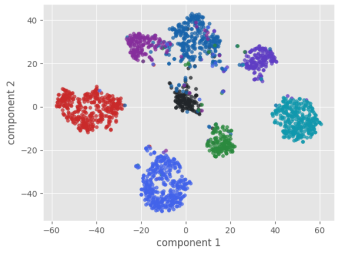
(c) PCA | Finetune: 1000 labels



(d) t-SNE | Pretrain



(e) t-SNE | Finetune: 250 labels



(f) t-SNE | Finetune: 1000 labels

Figure 7.5: Comparison of clustering quality between blood cells models.

7.4 Setup: Shorter Sequence

7.4.1 Metrics

Table 7.2 presents metrics evaluated on the test dataset and compares them to performance of standard SimCLR setup.

The results do not indicate significant change in classification performance for colon pathology and dermatology, and the shorter augmentation sequence performs worse for blood cells with small number of labelled data. For blood cells, we posit that the use of random horizontal flip and greyscale helps the model learn richer representations. We investigate this in Section 7.4.2.

	# Labelled Samples							
	100		250		1000		100%	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
Standard setup	0.701	0.939	0.785	0.965	0.830	0.977	0.875	0.964
Frozen backbone	0.692	0.903	0.787	0.965	0.834	0.979	-	-
Unfrozen backbone	0.646	0.883	0.777	0.939	0.831	0.955	0.868	0.973
Improvement	-0.9%	+0.004	+0.2%	+0.000	+0.4%	+0.002	-0.7%	+0.009

(a) Colon Pathology

	# Labelled Samples							
	100		250		1000		100%	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
Standard setup	0.677	0.756	0.701	0.839	0.741	0.873	0.787	0.925
Frozen backbone	0.668	0.759	0.698	0.840	0.714	0.882	0.741	0.909
Unfrozen backbone	0.682	0.741	0.709	0.846	0.735	0.871	0.780	0.926
Improvement	+0.5%	+0.003	+0.8%	+0.007	-0.6%	+0.009	-0.7%	+0.001

(b) Dermatology

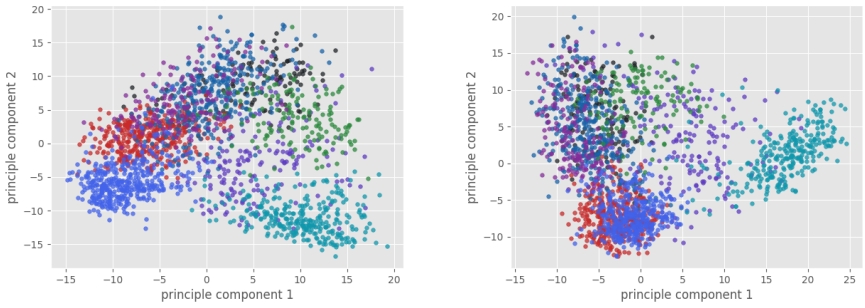
	# Labelled Samples							
	100		250		1000		100%	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
Standard setup	0.771	0.957	0.835	0.970	0.882	0.984	0.956	0.997
Frozen backbone	0.749	0.943	0.809	0.961	0.862	0.981	0.890	0.989
Unfrozen backbone	0.730	0.931	0.793	0.949	0.877	0.981	0.955	0.996
Improvement	-2.2%	-0.014	-2.6%	-0.009	-0.5%	-0.003	-0.1%	-0.001

(c) Blood cells

Table 7.2: Classification accuracy and AUC ROC of colon pathology, dermatology and blood cells. Random horizontal flip and greyscale are used in the standard setup as described in Section 3.2.1. Standard setup metrics are copied from Table 7.1. We investigate performance of models without using those augmentations. Applied to setting with 100, 250, 1000 and 100% labelled training data. Metrics are calculated using the entire test dataset provided by MedMNIST.

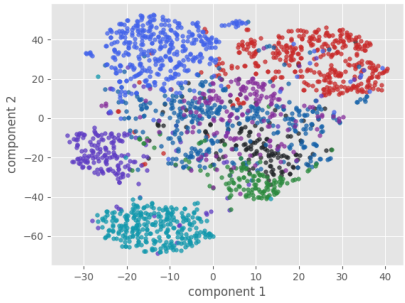
7.4.2 Learned Representations

Figure 7.6 presents visualisations of reduced learned representations during pretraining using PCA and t-SNE for blood cells. Since τ_{Sim} performes slightly worse with few labelled data, we posit that τ_{nat} helps the model to learn richer representations. We expect τ_{Sim} setup to have less well-defined clusters.

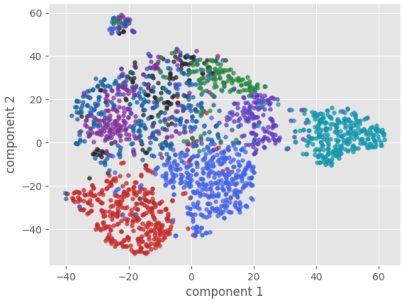


(a) PCA | Pretraining with τ_{nat}

(b) PCA | Pretraining with τ_{Sim}



(c) t-SNE | Pretraining with τ_{nat}



(d) t-SNE | Pretraining with τ_{Sim}

Figure 7.6: Comparison of clustering quality between learned features during pretraining using τ_{nat} and τ_{Sim} for blood cells. Refer to Figure 7.3 for labels.

In the figure, we observe slightly more overlapping in (b) compared to (a). We investigate this quantitatively using the silhouette coefficient.

$\text{SC}_{\text{Sim}} = 0.0568$ and $\text{SC}_{\text{nat}} = 0.156$ with the Euclidean metric. We only analyse the coefficients relative to each other, since the absolute value (which depends on the chosen distance metric) denotes clustering quality which we have analysed via visual inspection. The coefficient is higher for τ_{nat}, suggesting the clusters are more well-defined.

However, performing PCA with 2 components yields an explained variance of 0.149 and 0.153 for the setups with τ_{Sim} and τ_{nat} respectively. We further investigate this by performing PCA with 50 components, giving explained variance of > 0.75. With this setup, $\text{SC}_{\text{Sim}} = 0.0590$ and $\text{SC}_{\text{nat}} = 0.0739$. We conclude that the clusters are more distinct with SC_{nat}, suggesting that random horizontal flip and random greyscale helps the model to learn richer representations.

7.5 Setup: Novel Augmentation Sequence

7.5.1 Metrics

Table 7.3 presents metrics evaluated on the test dataset and compares them to performance of standard SimCLR setup.

The results indicate an overall improvement for colon pathology classification with a small

	# Labelled Samples							
	100		250		1000		100%	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
Standard setup	0.701	0.939	0.785	0.965	0.830	0.977	0.875	0.964
Frozen backbone	0.730	0.957	0.813	0.975	0.842	0.982	-	-
Unfrozen backbone	0.637	0.884	0.758	0.931	0.839	0.959	0.872	0.972
Improvement	+2.9%	+0.018	+2.8%	+0.010	+1.2%	+0.005	-0.3%	+0.008

(a) Colon Pathology

	# Labelled Samples							
	100		250		1000		100%	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
Standard setup	0.677	0.756	0.701	0.839	0.741	0.873	0.787	0.925
Frozen backbone	0.678	0.757	0.702	0.846	0.741	0.896	0.771	0.930
Unfrozen backbone	0.680	0.741	0.704	0.843	0.744	0.879	0.790	0.926
Improvement	+0.3%	+0.001	+0.3%	+0.007	+0.3%	+0.023	+0.3%	+0.004

(b) Dermatology

	# Labelled Samples							
	100		250		1000		100%	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
Standard setup	0.771	0.957	0.835	0.970	0.882	0.984	0.956	0.997
Frozen backbone	0.771	0.958	0.833	0.971	0.875	0.985	0.912	0.992
Unfrozen backbone	0.762	0.946	0.818	0.962	0.887	0.983	0.956	0.996
Improvement	+0.0%	+0.001	-0.2%	+0.001	+0.5%	+0.001	+0.0%	-0.001

(c) Blood cells

Table 7.3: Classification accuracy and AUC ROC of colon pathology, dermatology and blood cells. Standard setup metrics are copied from Table 7.1. We propose a novel augmentation sequence in Section 4.2 and investigate performance of models with frozen and unfrozen backbone. Applied to setting with 100, 250, 1000 and 100% labelled training data. Metrics are calculated using the entire test dataset provided by MedMNIST.

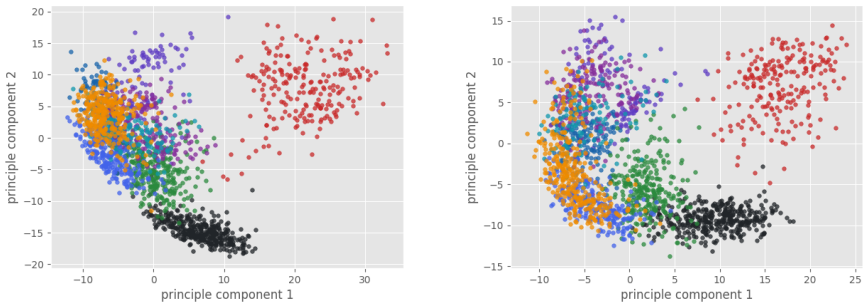
amount of labelled data during training. Our changes to τ are not extreme, so a small, consistent improvement over accuracy and AUC suggests the addition of random histogram equalisation and sharpness has potential to work well.

7.5.2 Learned Representations

Figure 7.7 presents visualisations of reduced learned representations during pretraining using PCA and t-SNE for colon pathology. Similarly, we check if τ_{nov} setup has more well-defined clusters.

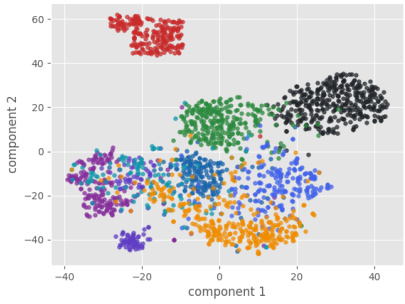
The overall PCA plots for colon pathology with τ_{nov} and τ_{nat} look similar. This is expected as τ_{nov} does not differ significantly from τ_{nat} and all other hyperparameters were kept the same. The t-SNE plots also exhibit similar patterns, such as *background* (red) being isolated and *mucus* (green) adjacent to *adipose* (black).

We measure clustering quality quantitatively with the Silhouette score as defined by Def-

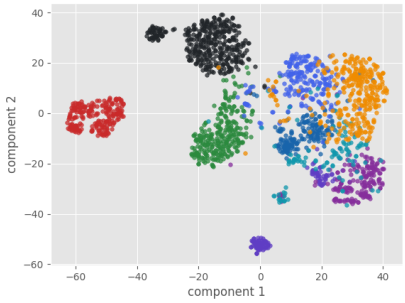


(a) PCA | Pretraining with τ_{nat}

(b) PCA | Pretraining with τ_{nov}



(c) t-SNE | Pretraining with τ_{nat}



(d) t-SNE | Pretraining with τ_{nov}

Figure 7.7: Comparison of clustering quality between learned features during pretraining using τ_{nat} and τ_{nov} for colon pathology. Refer to Figure 7.3 for labels.

inition 7.5. $\text{SC}_{\text{nov}} = 0.162$ and $\text{SC}_{\text{nat}} = 0.190$ with the Euclidean metric. Suprisingly, SC_{nat} is higher despite having lower accuracy in downstream learning. However, performing PCA with 2 components yield a very low explained variance of < 0.2 for both setups.

We perform PCA with 50 components, giving explained variance of > 0.84, and get $\text{SC}_{\text{nov}} = 0.150$ and $\text{SC}_{\text{nat}} = 0.105$. The clusters formed with SC_{nov} are more distinct, suggesting that the colon pathology model benefits from using histogram equalisation and sharpness augmentations to learn richer representations during pretraining.

7.6 Setup: Lack of Data

7.6.1 Metrics

Table 7.4 and Table 7.5 presents metrics evaluated on the test dataset for retina fundus and dermatology respectively. We provide interpretations of our results below and perform qualitative analysis in Section 7.6.2.

Table 7.4 includes baseline metrics with setups as described in Section 5.1.3. In general, pretraining with limited data does not improve classification accuracy. Suprisingly, with 1000 labelled images, pretraining followed by finetuning with frozen f yields a 3% increase in accuracy over the supervised baseline. However, the same comparison has a 3.5% decrease with 250 labelled images, suggesting these differences may be due to fluctuation.

Initial pretraining on a large dataset from a different category (we use colon pathology) followed by finetuning with frozen f yields best improvement over baseline metrics. The rest of the setups involve tuning parameters of f either during pretraining or during downstream learning, and yields similar accuracy to baseline models. From these observations, we posit that

		# Labelled Samples					
		100		250		1000	
Pretrain	Downstream	ACC	AUC	ACC	AUC	ACC	AUC
None	Supervised	0.458	0.648	**0.498**	0.630	0.480	0.668
Retina	Frozen f	**0.460**	0.660	0.463	0.658	**0.510**	**0.697**
Retina	Unfrozen f	0.457	**0.664**	0.455	**0.663**	0.468	0.667

(a) Baseline Metrics

		# Labelled Samples					
		100		250		1000	
Pretrain	Freeze f	ACC	AUC	ACC	AUC	ACC	AUC
Path	yes	**0.488**	0.677	**0.530**	**0.686**	**0.522**	**0.726**
Path	no	0.452	0.644	0.455	0.651	0.480	0.680
Path + Retina	yes	0.455	**0.683**	0.498	**0.686**	0.477	0.688
Path + Retina	no	0.452	0.632	0.450	0.644	0.477	0.695
Improvement		+2.8%	+0.019	+3.2%	+0.023	+1.2%	+0.029

(b) Metrics

Table 7.4: Classification accuracy and AUC ROC of retina fundus images. Applied to setting with 100, 250, 1000 labelled training data. Baseline environments are described in Section 5.1.3. Metrics are calculated using the entire test dataset provided by MedMNIST.

the model learns useful features that contribute to overall improvement during initial pretraining on the large dataset, and that these features are overridden when we use retina fundus images to tune f. We investigate this in Section 7.6.2.

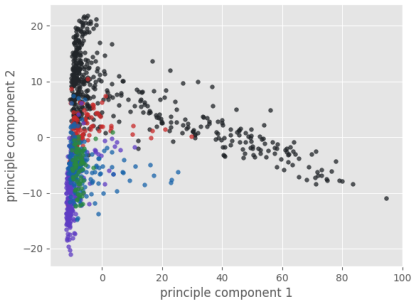
Table 7.5 presents metrics from models finetuned using a balanced dermatology dataset during downstream learning. Consider the models we previously trained using 100 labelled images with an unbalanced dataset, for which we were able to achieve 68% accuracy[3]. Here, even with 100 labelled images from each of the 7 classes (which amounts to almost 700 labelled images in total[4]), we only achieve 65.3% accuracy. We propose the following explanations.

[3] SimCLR pretraining with novel augmentation sequence and downstream learning with unfrozen backbone.

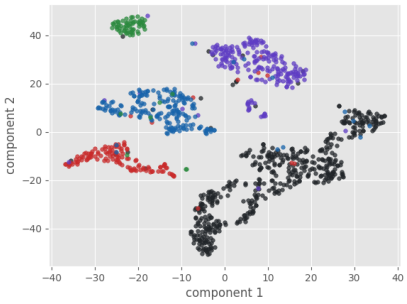
[4] Dermatofibroma has 80 samples only and vascular lesions has 99 samples. Therefore, the experiment uses 679 images.

		# Labelled Samples from Each Class					
		10		25		100	
Pretrain	Freeze f	ACC	AUC	ACC	AUC	ACC	AUC
Path	yes	0.427	**0.823**	0.496	0.767	0.546	**0.850**
Path	no	0.442	0.707	0.590	0.750	0.600	0.776
Path + Derma	yes	0.424	0.795	0.495	**0.866**	0.575	0.803
Path + Derma	no	**0.454**	0.741	**0.637**	0.706	**0.653**	0.776

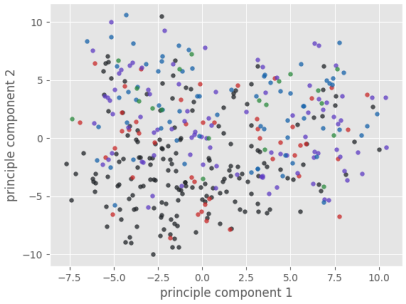
Table 7.5: Classification accuracy and AUC ROC of dermatology. Applied to setting with 10, 25, 100 labelled training data from each class. Metrics are calculated using the entire test dataset provided by MedMNIST.



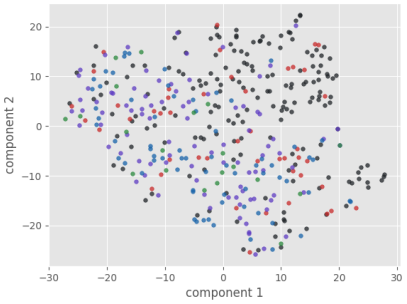
(a) PCA | Supervised Baseline



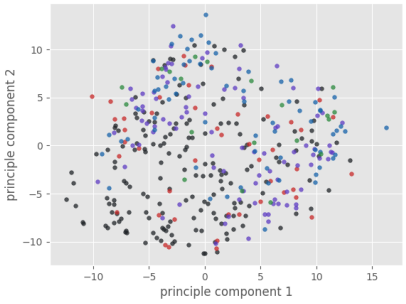
(b) t-SNE | Supervised Baseline



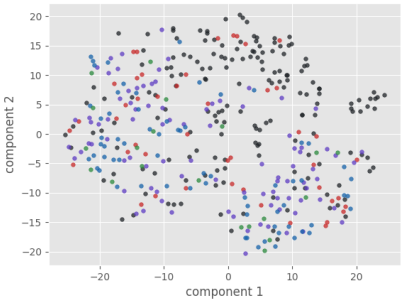
(c) PCA | Pretrain with Retina Fundus



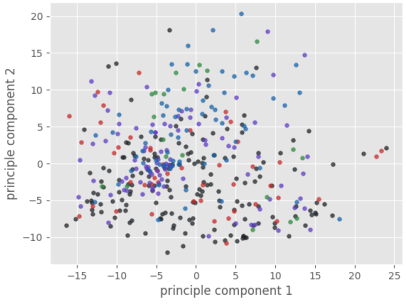
(d) t-SNE | Pretrain with Retina Fundus



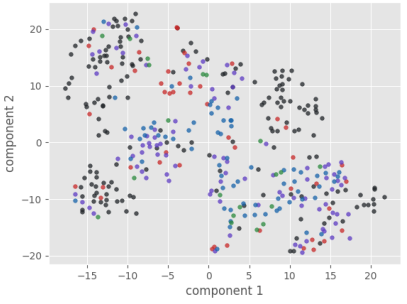
(e) PCA | Pretrain with Pathology
Further Pretrain with Retina Fundus



(f) t-SNE | Pretrain with Pathology
Further Pretrain with Retina Fundus



(g) PCA | Pretrain with Pathology



(h) t-SNE | Pretrain with Pathology

Figure 7.8: PCA and t-SNE of SimCLR pretraining and baseline supervised models for retina fundus (note that labels are ordinal regression). t-SNE components are determined using 100% training data for PCA, followed by 2000 reduced test data points for t-SNE. Each plot displays the reduced representations of 2000 retina fundus data points from the test set.

1. Features learned for colon pathology do not transfer well to dermatology images.
2. It is better to use all available data for training rather than undersampling to balance the dataset.

It is important to note that we have done only a brief investigation on transfer learning across different image categories and imbalanced datasets. Further investigation is required to provide more concrete insights.

7.6.2 Learned Representations

Figure 7.8 presents visualisations of reduced learned representations during pretraining using PCA and t-SNE for models later tuned for retina fundus.

There is no formation of clusters for any of the pretrained models. The Silhouette scores for the plots are close to 0. Since distinct clusters are formed for the baseline supervised model, this may suggest that the pretrained models learned representations that are either not transferable to retina fundus (in the case of pretraining with colon pathology), or that the representations are extremely weak (in the case of pretraining with retina fundus). The latter is supported by how pretraining with retina fundus did not increase classification accuracy of the corresponding downstream models with respect to baseline metrics.

However, we did see improvement over baseline metrics when performing initial pretraining with colon pathology followed by finetuning with frozen f for retina fundus. Performing PCA with 50 components (and explained variance of 0.981) for this setup yields a Silhouette score of -0.0078. We posit that the encoder as a whole is not transferrable, but the early layers of f may have extracted transferable features.

Our results suggest initial SimCLR pretraining on a large dataset followed by finetuning with frozen f for a specialised dataset is effective for medical images.

7.7 Setup: Greyscale Images

7.7.1 Metrics

Table 7.6 presents metrics evaluated on the test dataset. We provide interpretations of our results below and perform qualitative analysis in Section 7.7.2.

Table 7.6 suggests that SimCLR pretraining with our novel augmentation proposition τ_{grey} is very effective on greyscale medical images, specifically for classifying tissue cells and retinal OCT. With a limited amount of labelled data, we observe over 8% increase in accuracy for tissue cells and over 10% increase for retinal OCT across all simulated limited data environments (100, 250 and 1000 labelled samples for finetuning). Likewise, we observe significant increase in AUC metrics.

The metrics suggest finetuning with frozen f yields best performance, although pretraining then finetuning the encoder also improves over baseline supervised metrics.

For retinal OCT classification, pretraining followed by finetuning with 1000 labelled images yields similar performance to performing supervised learning with 97,477[5] labelled images.

We previously had similar success with pretraining on the colon pathology dataset in Section 7.3.1. We draw a connection that the colon pathology, tissue cells and retinal OCT datasets are by far the largest datasets used throughout this paper, with over 89,000 training samples each. The next largest dataset is blood cells at 11,959 training samples. Having performed experiments

[5] 100% of training dataset

	# Labelled Samples							
	100		250		1000		100%	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
Supervised baseline	0.403	0.648	0.437	0.682	0.463	0.728	0.637	**0.917**
Frozen backbone	**0.486**	**0.778**	**0.517**	**0.812**	**0.546**	**0.853**	-	-
Unfrozen backbone	0.459	0.725	0.487	0.761	0.513	0.790	0.636	0.889

(a) Tissue Cells

	# Labelled Samples							
	100		250		1000		100%	
	ACC	AUC	ACC	AUC	ACC	AUC	ACC	AUC
Supervised baseline	0.380	0.659	0.475	0.755	0.601	0.818	0.723	0.901
Frozen backbone	**0.665**	**0.899**	**0.658**	**0.915**	**0.708**	**0.909**	-	-
Unfrozen backbone	0.570	0.786	0.637	0.851	0.663	0.857	**0.736**	0.903

(b) Retinal OCT

Table 7.6: Classification accuracy and AUC ROC of tissue cells and retinal OCT. Performance of models trained with pretraining then downstream learning with frozen/unfrozen backbone is compared to performance of models from baseline supervised learning. Metrics are calculated using the entire test dataset provided by MedMNIST. Best-performing environments are bolded.

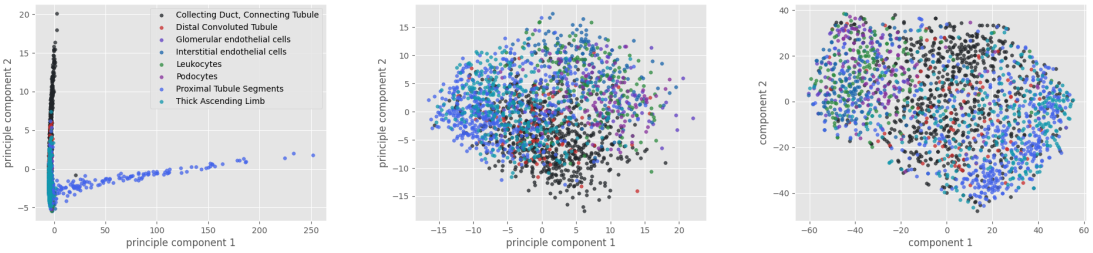
for 7 medical imaging modalities, there is compelling evidence that SimCLR pretraining benefits from very large datasets of the same modality as downstream tasks.

7.7.2 Learned Representations

Figure 7.9 presents visualisations of reduced learned representations during pretraining using PCA and t-SNE. We outline our findings below.

We observe distinct clusters formed for retinal OCT during pretraining. The clusters are compact compared to the reduced representations from the baseline model, suggesting that relevant patterns are learned by the pretrained model, but they are not as rich as the patterns learned with supervised learning.

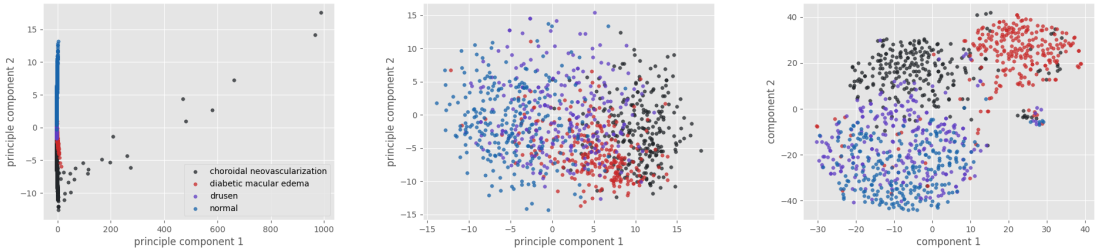
We observe a weaker formation of clusters for tissue cells for the pretrained model, despite distinct clusters forming for the corresponding baseline model. Noticeably, the pretrained model distinguishes between *collecting duct, connecting tubule* and *proximal tubule segments*. We attribute these weak clusters to the fact that the ResNet models struggled with classifying tissue cells. With 100% of the labelled training set, the model trained with supervised learning achieved 63.7% accuracy only.



(a) Tissue Cells
PCA | Supervised Baseline

(b) Tissue Cells
PCA | SimCLR Pretraining

(c) Tissue Cells
t-SNE | SimCLR Pretraining



(d) Retinal OCT
PCA | Supervised Baseline

(e) Retinal OCT
PCA | SimCLR Pretraining

(f) Retinal OCT
t-SNE | SimCLR Pretraining

Figure 7.9: PCA and t-SNE of SimCLR pretraining and baseline supervised models on tissue cells and retinal OCT. Principal components are determined using 100% of the training dataset. t-SNE components are determined using 100% training data for PCA, followed by 2000 reduced test data points for t-SNE. Each plot displays the reduced representations of 2000 data points from the test set.

Chapter 8

Conclusion

In this thesis, we evaluate the SimCLR framework for medical image classification. We implement SimCLR in Python using PyTorch Lightning, as well as two downstream environments: finetuning with a frozen backbone and finetuning with an unfrozen backbone. We implement evaluation tools to provide a comprehensive, unbiased assessment of SimCLR and our proposed changes.

When using the setup as described in the original papers[4, 5], SimCLR pretraining substantially improves performance over supervised baseline on medical imaging modalities of colour when presented with lots of unlabelled data and a lack of labelled data. SimCLR pretraining offers small improvement when lots of labelled data are present.

We find retaining the use of random horizontal flip and random greyscale augmentations causes a small improvement in classifying blood cells in a limited labelled data environment. We propose a novel augmentation sequence involving random histogram equalisation and random sharpness, which consistently outperforms the original augmentation sequence for medical imaging modalities of colour. We also propose a sequence for greyscale medical images which substantially improves performance over supervised models when lack of labelled data is present.

With limited unlabelled and labelled data, we propose initial pretraining with a different, larger dataset, then freezing the backbone and finetuning with the labelled data from the specialised dataset. This approach improves over baseline metrics when evaluated on retina fundus images. We briefly investigate the effect of data imbalance and find that balancing the dermatology dataset with undersampling yields poorer performance than using the entire dataset.

8.1 Future Work

Out-of-distribution Data

In this project, we source medical images from the MedMNIST database for pretraining, finetuning and testing. Future work involves using other datasets and out-of-distribution data for evaluation. For example, we carry out experiments on classifying dermatology images (concerning skin conditions) separated into 7 classes. It may be of interest to assess the effectiveness of SimCLR pretraining against medical imaging modalities that are categorised into more specific subclasses, for instance, a dermatology dataset with 50 classes.

Downsampling-Performance Tradeoff

SimCLR benefits from long training with many epochs. Medical images are often large in size. In this project, we scale the original images down to 28×28. Training models with low resolution images yield strong performance for some modalities like blood cells (95.6% accuracy), but other modalities like tissue cells perform poorly (63.7% accuracy). We did not investigate whether these modalities suffer from downsampling images, or whether CNNs struggle with them generally. A useful problem to investigate is given a fixed amount of pretrain time, identify the downsampled resolution that yields optimal performance.

Augmentation Sequence

In this thesis, we propose a novel augmentation sequence involving random histogram equalisation and sharpness. We perform experiments and evaluations as a proof of concept that the addition of these augmentations work well for many medical imaging modalities. Future work involves hyperparameter tuning to discover the potential of equalisation and sharpness. Future work also includes investigating whether these augmentations can be adopted in other contrastive frameworks, such as MoCo and ReLIC[2, 3], as well as incorporating finetuning techniques introduced in SimCLRv2[5] such as distillation and keeping part of the projection head.

Lack of Data

In this thesis, we explore initial pretraining on a different medical imaging modality before finetuning with the specialised modality if there is a lack of data for the latter. We conduct experiments with colon pathology and retina fundus. Future work involves investigation into other modalities, as well as initial pretraining with natural images.

We briefly investigate tackling data imbalance with undersampling. It is worth investigating into upsampling, potentially with a different set of augmentations.

8.2 Final Remarks

We hope our contributions will help advance the state of the art for medical image classification and see an adoption in using contrastive learning and self-supervised learning in the medical field.

Bibliography

[1] Oquab M, Darcet T, Moutakanni T, Vo H, Szafraniec M, Khalidov V, et al. Dinov2: Learning robust visual features without supervision. arXiv preprint arXiv:230407193. 2023.

[2] Mitrovic J, McWilliams B, Walker J, Buesing L, Blundell C. Representation learning via invariant causal mechanisms. arXiv preprint arXiv:201007922. 2020.

[3] Tomasev N, Bica I, McWilliams B, Buesing L, Pascanu R, Blundell C, et al. Pushing the limits of self-supervised ResNets: Can we outperform supervised learning without labels on ImageNet? arXiv preprint arXiv:220105119. 2022.

[4] Chen T, Kornblith S, Norouzi M, Hinton G. A simple framework for contrastive learning of visual representations. In: International conference on machine learning. PMLR; 2020. p. 1597-607.

[5] Chen T, Kornblith S, Swersky K, Norouzi M, Hinton GE. Big self-supervised models are strong semi-supervised learners. Advances in neural information processing systems. 2020;33:22243-55.

[6] Papers with Code; 2023. Accessed on: June 10, 2023. `https://paperswithcode.com/sota/self-supervised-image-classification-on`.

[7] Lynch CJ, Liston C. New machine-learning technologies for computer-aided diagnosis. Nature medicine. 2018;24(9):1304-5.

[8] Chan HP, Hadjiiski LM, Samala RK. Computer-aided diagnosis in the era of deep learning. Medical physics. 2020;47(5):e218-27.

[9] Lostumbo A, Suzuki K, Dachman AH. Flat lesions in CT colonography. Abdominal imaging. 2010;35:578-83.

[10] Huang Z, Pan Z, Lei B. Transfer learning with deep convolutional neural network for SAR target classification with limited labeled data. Remote Sensing. 2017;9(9):907.

[11] Agrawal T, Gupta R, Narayanan S. On evaluating CNN representations for low resource medical image classification. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE; 2019. p. 1363-7.

[12] He K, Fan H, Wu Y, Xie S, Girshick R. Momentum contrast for unsupervised visual representation learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition; 2020. p. 9729-38.

[13] Grill JB, Strub F, Altché F, Tallec C, Richemond P, Buchatskaya E, et al. Bootstrap your own latent-a new approach to self-supervised learning. Advances in neural information processing systems. 2020;33:21271-84.

[14] Ghesu FC, Georgescu B, Mansoor A, Yoo Y, Neumann D, Patel P, et al. Self-supervised learning from 100 million medical images. arXiv preprint arXiv:220101283. 2022.

[15] Chen L, Bentley P, Mori K, Misawa K, Fujiwara M, Rueckert D. Self-supervised learning for medical image analysis using image context restoration. Medical image analysis. 2019;58:101539.

[16] Zhou Z, Sodha V, Rahman Siddiquee MM, Feng R, Tajbakhsh N, Gotway MB, et al. Models genesis: Generic autodidactic models for 3d medical image analysis. In: Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part IV 22. Springer; 2019. p. 384-93.

[17] Azizi S, Culp L, Freyberg J, Mustafa B, Baur S, Kornblith S, et al. Robust and efficient medical imaging with self-supervision. arXiv preprint arXiv:220509723. 2022.

[18] Azizi S, Mustafa B, Ryan F, Beaver Z, Freyberg J, Deaton J, et al. Big self-supervised models advance medical image classification. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2021. p. 3478-88.

[19] Yang J, Shi R, Ni B. MedMNIST Classification Decathlon: A Lightweight AutoML Benchmark for Medical Image Analysis. In: IEEE 18th International Symposium on Biomedical Imaging (ISBI); 2021. p. 191-5.

[20] Yang J, Shi R, Wei D, Liu Z, Zhao L, Ke B, et al. MedMNIST v2: A Large-Scale Lightweight Benchmark for 2D and 3D Biomedical Image Classification. arXiv preprint arXiv:211014795. 2021.

[21] Balestriero R, LeCun Y. Contrastive and non-contrastive self-supervised learning recover global and local spectral embedding methods. arXiv preprint arXiv:220511508. 2022.

[22] Hannun A, Case C, Casper J, Catanzaro B, Diamos G, Elsen E, et al. Deep speech: Scaling up end-to-end speech recognition. arXiv preprint arXiv:14125567. 2014.

[23] Lu C, Tang X. Surpassing human-level face verification performance on LFW with GaussianFace. In: Proceedings of the AAAI conference on artificial intelligence. vol. 29; 2015. .

[24] Wu Y, Schuster M, Chen Z, Le QV, Norouzi M, Macherey W, et al. Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:160908144. 2016.

[25] Duch W. Separability is not the best goal for machine learning. arXiv preprint arXiv:180702873. 2018.

[26] Zhang A, Lipton ZC, Li M, Smola AJ. Dive into deep learning. arXiv preprint arXiv:210611342. 2021.

[27] Lowe DG. Object recognition from local scale-invariant features. In: Proceedings of the seventh IEEE international conference on computer vision. vol. 2. Ieee; 1999. p. 1150-7.

[28] Sultana F, Sufian A, Dutta P. Advancements in image classification using convolutional neural network. In: 2018 Fourth International Conference on Research in Computational Intelligence and Communication Networks (ICRCICN). IEEE; 2018. p. 122-9.

[29] Alom MZ, Taha TM, Yakopcic C, Westberg S, Sidike P, Nasrin MS, et al. The history began from alexnet: A comprehensive survey on deep learning approaches. arXiv preprint arXiv:180301164. 2018.

[30] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:14091556. 2014.

[31] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S, Anguelov D, et al. Going deeper with convolutions. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2015. p. 1-9.

[32] Bengio Y, Simard P, Frasconi P. Learning long-term dependencies with gradient descent is difficult. IEEE transactions on neural networks. 1994;5(2):157-66.

[33] Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward neural networks. In: Proceedings of the thirteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings; 2010. p. 249-56.

[34] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning. pmlr; 2015. p. 448-56.

[35] Lee CY, Xie S, Gallagher P, Zhang Z, Tu Z. Deeply-supervised nets. In: Artificial intelligence and statistics. PMLR; 2015. p. 562-70.

[36] Venables WN, Ripley BD. Modern applied statistics with S-PLUS. Springer Science & Business Media; 2013.

[37] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 770-8.

[38] Devlin J, Chang MW, Lee K, Toutanova K. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:181004805. 2018.

[39] Brown T, Mann B, Ryder N, Subbiah M, Kaplan JD, Dhariwal P, et al. Language models are few-shot learners. Advances in neural information processing systems. 2020;33:1877-901.

[40] Taylor L, Nitschke G. Improving deep learning with generic data augmentation. In: 2018 IEEE symposium series on computational intelligence (SSCI). IEEE; 2018. p. 1542-7.

[41] Liu X, Zhang F, Hou Z, Mian L, Wang Z, Zhang J, et al. Self-supervised learning: Generative or contrastive. IEEE Transactions on Knowledge and Data Engineering. 2021;35(1):857-76.

[42] Pokle A, Tian J, Li Y, Risteski A. Contrasting the landscape of contrastive and non-contrastive learning. arXiv preprint arXiv:220315702. 2022.

[43] Lippe P. Contrastive Learning; 2023. Accessed: March 15, 2023. `https://lightning.ai/docs/pytorch/stable/notebooks/course_UvA-DL/13-contrastive-learning.html`.

[44] Hinton G, Vinyals O, Dean J. Distilling the knowledge in a neural network. arXiv preprint arXiv:150302531. 2015.

[45] Coates A, Ng A, Lee H. An analysis of single-layer networks in unsupervised feature learning. In: Proceedings of the fourteenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings; 2011. p. 215-23.

[46] Oord Avd, Li Y, Vinyals O. Representation learning with contrastive predictive coding. arXiv preprint arXiv:180703748. 2018.

[47] Wu Z, Xiong Y, Yu SX, Lin D. Unsupervised feature learning via non-parametric instance discrimination. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2018. p. 3733-42.

[48] Chen X, Fan H, Girshick R, He K. Improved baselines with momentum contrastive learning. arXiv preprint arXiv:200304297. 2020.

[49] Li F, Aoyama M, Shiraishi J, Abe H, Li Q, Suzuki K, et al. Radiologists' performance for differentiating benign from malignant lung nodules on high-resolution CT using computer-estimated likelihood of malignancy. American Journal of Roentgenology. 2004;183(5):1209-15.

[50] Doi K. Computer-aided diagnosis in medical imaging: historical review, current status and future potential. Computerized medical imaging and graphics. 2007;31(4-5):198-211.

[51] Caron M, Misra I, Mairal J, Goyal P, Bojanowski P, Joulin A. Unsupervised learning of visual features by contrasting cluster assignments. Advances in neural information processing systems. 2020;33:9912-24.

[52] Caron M, Bojanowski P, Joulin A, Douze M. Deep clustering for unsupervised learning of visual features. In: Proceedings of the European conference on computer vision (ECCV); 2018. p. 132-49.

[53] Gidaris S, Singh P, Komodakis N. Unsupervised representation learning by predicting image rotations. arXiv preprint arXiv:180307728. 2018.

[54] Caron M, Bojanowski P, Mairal J, Joulin A. Leveraging large-scale uncurated data for unsupervised pre-training of visual features. CoRR. 2019.

[55] Philipsen RH, Maduskar P, Hogeweg L, Melendez J, Sánchez CI, van Ginneken B. Localized energy-based normalization of medical images: application to chest radiography. IEEE transactions on medical imaging. 2015;34(9):1965-75.

[56] Castro E, Cardoso JS, Pereira JC. Elastic deformations for data augmentation in breast cancer mass detection. In: 2018 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI). IEEE; 2018. p. 230-4.

[57] Dorothy R, Joany R, Rathish RJ, Prabha SS, Rajendran S, Joseph S. Image enhancement by histogram equalization. International Journal of Nano Corrosion Science and Engineering. 2015;2(4):21-30.

[58] Sharma D. Intensity transformation using contrast limited adaptive histogram equalization. International Journal of Engineering Research. 2013;2(4):282-5.

[59] Gonzalez RC, Woods RE, Eddins S. Digital Image Processing Using MATLAB: Pearson Prentice Hall. Upper Saddle River, New Jersey. 2004.

[60] Xie H, Shan H, Cong W, Zhang X, Liu S, Ning R, et al. Dual network architecture for few-view CT-trained on ImageNet data and transferred for medical imaging. In: Developments in X-ray Tomography XII. vol. 11113. SPIE; 2019. p. 184-94.

[61] Van der Maaten L, Hinton G. Visualizing data using t-SNE. Journal of machine learning research. 2008;9(11).

[62] Rousseeuw PJ. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics. 1987;20:53-65.

Appendix A

SIFT

We provide some preliminary definitions below.

Definition A.1 (Convolution)**.** *Given descrete functions f and g, a convolution $*$ describes the amount of overlap of g as it is shifted across f.*

$$\big(f * g\big)[n] = \sum_{k=-\infty}^{\infty} f[k]g[n-k] \tag{A.1}$$

Definition A.2 (Difference of Gaussian)**.** *The difference of Gaussian filter is defined in* (A.2) *where G denotes the Gaussian function and k is a constant.*

$$DoG(x, y, \sigma) = I * G(k\sigma) - I * G(\sigma) \tag{A.2}$$

Definition A.3 (Gradient Magnitude and Orientation)**.** *Let f denote the input image. Let h_x and h_y denote Sobel filters.*

$$h_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \qquad h_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \tag{A.3}$$

The gradient magnitude is:

$$g = \sqrt{g_x^2 + g_y^2} \tag{A.4}$$

The gradient orientation is:

$$\theta = \arctan\left(\frac{g_y}{g_x}\right) \tag{A.5}$$

SIFT transforms an image into a set of interest points. The algorithm is described below.

1. **Detection of scale-space extrema** - Apply DoG with various values of k to detect extrema at multiple scales.
2. **Keypoint localisation** - Fit a quadratic function to DoG response of neighbouring pixels to refine extrema estimates to sub-pixel accuracy. Refined points are called keypoints.
3. **Orientation assignment** - For each keypoint, calculate the gradient magnitude and orientation of pixels in its neighbourhood. Each pixel votes for an orientation bin, weighted by its gradient magnitude. This creates a histogram of orientations and the keypoint is assigned the dominant orientation.
4. **Keypoint descriptor** - A 128-dimensional feature vector is used to describe each keypoint. Each keypoint describes the gradient magnitude and orientation of 16 local subregions to achieve scale and orientation invariance. Gradients are robust to intensity.

Appendix B

MedMNIST Dataset Details

B.1 Original Sizes

Dataset	Resolution (pixels)
Colon Pathology	224 × 224
Dermatology	600 × 450
Blood Cells[1]	200 × 200
Retina Fundus	1736 × 1824
Tissue	32 × 32
Retinal OCT	(384–1536) × (227–512)

Table B.1: Resolution of source images for different datasets from MedMNIST before being resized.

B.2 Sample Distribution

Class	# Samples		
	Train	Validation	Test
adipose	9,366	1,041	1,338
background	9,509	1,057	847
debris	10,360	1,152	339
lymphocytes	10,401	1,156	634
mucus	8,006	890	1,035
smooth muscle	12,182	1,354	592
normal colon mucosa	7,886	877	741
cancer-associated stroma	9,401	1,045	421
colorectal adenocarcinoma epithelium	12,885	1,432	1,233
$\sum$	89,996	10,004	7,180

Table B.2: Samples distribution per label for colon pathology.

[1] Centre cropped from 360 × 363

Class	# Samples		
	Train	Validation	Test
actinic keratoses and intraepithelial carcinoma	228	33	66
basal cell carcinoma	359	52	103
benign keratosis-like lesions	769	110	220
dermatofibroma	80	12	23
melanoma	779	111	223
melanocytic nevi	4,693	671	1,341
vascular lesions	99	14	29
$\sum$	7,007	1,003	2,005

Table B.3: Samples distribution per label for dermatology.

Class	# Samples		
	Train	Validation	Test
basophil	852	122	244
eosinophil	2,181	312	624
erythroblast	1,085	155	311
immature granulocytes	2,026	290	579
lymphocyte	849	122	243
monocyte	993	143	284
neutrophil	2,330	333	666
platelet	1,643	235	470
$\sum$	11,959	1,712	3,421

Table B.4: Samples distribution per label for blood cells.

Class	# Samples		
	Train	Validation	Test
A	486	54	174
B	128	12	46
C	206	28	92
D	194	20	68
E	66	6	20
$\sum$	1,080	120	400

Table B.5: Samples distribution per label for retina fundus. Classes are based off of ordinal regression.

Class	# Samples		
	Train	Validation	Test
Collecting Duct, Connecting Tubule	53,075	7,582	15,165
Distal Convoluted Tubule	7,814	1,117	2,233
Glomerular endothelial cells	5,866	838	1,677
Interstitial endothelial cells	15,406	2,201	4,402
Leukocytes	11,789	1,684	3,369
Podocytes	7,705	1,101	2,202
Proximal Tubule Segments	39,203	5,601	11,201
Thick Ascending Limb	24,608	3,516	7,031
$\sum$	170,866	24,440	48,880

Table B.6: Samples distribution per label for tissue.

Class	# Samples		
	Train	Validation	Test
Choroidal Neovascularization	33,484	3,721	250
Diabetic Macular Edema	10,213	1,135	250
Drusen	7,754	862	250
Normal	46,026	5,114	250
$\sum$	97,477	10,732	1,000

Table B.7: Samples distribution per label for Retinal OCT. Classes are based on ordinal regression.

Appendix C

Pretraining Epochs

Modality	Epochs
colon pathology	201
dermatology	2,000
blood cells	2,000
retina fundus	10,000
retinal OCT	200
tissue cells	200

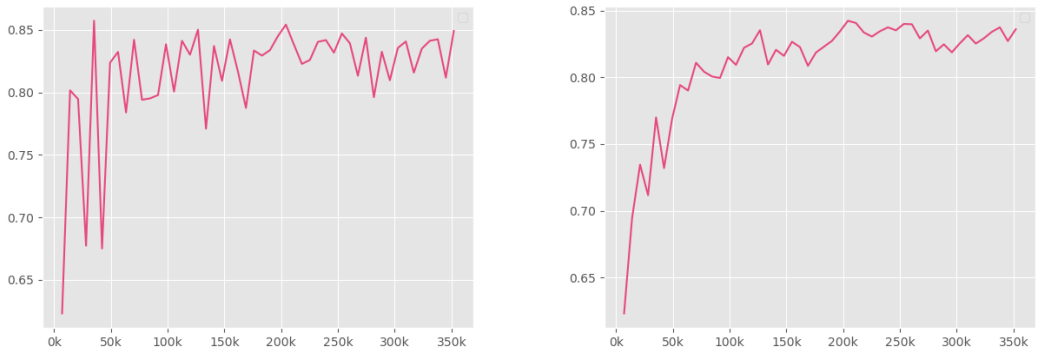
Table C.1: Number of epochs during pretraining across different medical imaging modalities. For larger datasets, lower epochs is used for practical training durations.

Appendix D

Graph Smoothing

We apply an exponential moving average (EMA), as defined in (D.1), to smooth out graphs when appropriate, for example, when displaying train and validation accuracy over time. The original graphs exhibit fluctuations due to mini-batch gradient descent. Figure D.1 compares graphs before and after applying EMA.

$$x^{(i)}_{\text{smoothed}} = (1-\alpha)x^{(i)} + \alpha x^{(i-1)} \quad 0 \leq \alpha \leq 1 \tag{D.1}$$



(a) original

(b) smoothed

Figure D.1: Example plots to showcase effect of applying EMA.