Big Data, Small Labels: Contrastive Learning for Medical Image Analysis

Luke Jenkinson

Supervised by: Paul Taylor and Watjana Lilaonitkul

Doctor of Philosophy

of

University College London.

Department of Computer Science

University College London

Declaration

I, Luke Jenkinson, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work.

Acknowledgements

Firstly, I would like to thank my supervisors, Waty and Paul, for their support and guidance throughout the PhD process, particularly through the COVID-19 pandemic and the many challenges that brought.

I'd also like to thank the many wonderful people who I have had the chance to work with and learn from including my CDT colleagues (particularly, Mark, Francisca, Santeri, Alistair, Toby, Adam, Ash, Anna, Mar, Morgan, Robert, Will) and my IHI colleagues (Boyu, Vincent, Asma, Chris, Becky, Patrick, Peter, Ghada, Jude, Bettina, Andre).

I'd like to thank my mum for her endless support and encouragement; my friends for their distraction from writing; and finally, I'd also like to thank my partner, Maddy.

Abstract

Deep neural networks have become the de facto standard for many computer vision tasks. Despite this, the uptake of these state-of-the-art methods to medical imaging tasks has been lacking. One possible reason for this is the scarcity of large, labelled medical image datasets for these models to train on. To combat this, numerous semi-supervised methods have been suggested for improving performance when faced with limited labelled training data. One such class of methods is contrastive learning: these methods aim to learn powerful features from unlabelled data, which can then be used, along with a small amount of labelled training data to produce higher performance than could be achieved with the labelled data alone.

This thesis examines two distinct contrastive methods, Contrastive Predictive Coding and SimCLR, across multiple dimensions. In the first part of this thesis, the ability of contrastive methods to increase performance on medical imaging tasks is validated, exploring how the size of the labelled dataset changes the performance of the downstream task compared with a powerful baseline. Additional work is undertaken to understand how this improvement in accuracy may affect the robustness of the model. In the second section of this thesis, design choices of the contrastive training protocol are examined to understand how to achieve the greatest performance. Some contrastive methods, most notably SimCLR, make heavy use of augmentation in their training protocol, and the impact of this has been under studied. This thesis examines the impact of both the type and the magnitude of these augmentations. Finally, a large study of the impact of unlabelled dataset on the downstream classification performance is presented, giving novel recommendations for how to improve performance on a wide variety of tasks.

Impact Statement

This thesis increases the utility and applicability of semi-supervised methods within the field of medical imaging, and more broadly across label limited but data rich problem sets. Three research questions were studied, with questions two and three building on the findings of the first:

1. Are semi-supervised approaches useful within the medical domain?
2. How should the methods be adapted to produce the best results?
3. Are there features of the data itself that can predict whether a semi-supervised method will work?

The first research question provides the foundation for the subsequent work. In the first few chapters, I highlight that the technical evaluation of these semi-supervised methods on datasets outside of general imaging datasets is lacking. There are two contributions from this work: a) it provides further evidence for the use of semi-supervised methods in medical imaging; b) it provides justification for the deeper experimentation of the subsequent research questions. This first chapter shows that, not only does Contrastive Predictive Coding have higher accuracy than a ResNet baseline under a low data regime, but that other important attributes for medical imaging are improved: model robustness to domain shift. In addition, section 4.3.2 of this chapter provides a technically novel solution for how to increase the robustness of the model to perturbation, showing that including relevant augmentations during unsupervised pre-training increases robustness to those perturbations during downstream inference.

From these foundational results, results chapters 2 and 3 (chapters 6 and 7) delve further into how the performance of these contrastive methods can be improved. Using SimCLR as a case study, chapter 6 studies how augmentation impacts these methods, finding that some of the results of [1] and [2] are overly reductive and this chapter provides practical guidance on optimising how augmentation should be applied. I argue that automated hyperparameter tuning should be used as best practice, rather than relying on general rules for what augmentations to use. In this

chapter, work is also conducted to evaluate often repeated, but never substantiated claims that contrastive learning produces representations that encode high level features that do not change under image augmentation. While often repeated, to the best of my knowledge, this is the first time that this claim has been validated.

The final results chapter presents a substantial study into how the datasets themselves impact performance, and the work challenges a number of commonly held beliefs. This chapter provides considerable expansion on the work of [3] [4] and [5], a set of other pieces of work that studied the effects of dataset on performance. This chapter extends several of their findings, and postulates that a significant problem within the self-supervised learning space is underfitting on datasets. Because self-supervised learning does not need the expensive label generation to produce good embeddings, the datasets that can be used with them are multiple orders of magnitude larger than those found in general deep learning. This work initially found the same results as [3]: that larger datasets did not produce better results for self-supervised learning. However, further investigation found that this was due to the networks underfitting at the larger dataset sizes. By following the suggestion of this chapter to introduce early stopping, one can increase performance. This thesis's suggestion to use early stopping within unsupervised training to increase performance on the downstream task is technically novel. In the discussion section of this chapter, I postulate that even the original SimCLR has underfit, and that performance gains could be achieved through training for longer. This has significant impact on how SSL should be approached in the future. I believe that greater performance can be gained through engineering challenges: training larger models, on faster hardware, for much longer, rather than through the invention of better models.

Nomenclature

Acronyms

AI: Artificial Intelligence

CXR: Chest x-ray

CT: Computerised Tomography

ECG: Electrocardiogram

EEG: Electroencephalogram

GPU: Graphics Processing Unit

ML: Machine Learning

MRI: Magnetic Resonance Imaging

OCT: Optical Coherence Tomography

Methods

CPC: Contrastive Predictive Coding [6]

SimCLR: A Simple Framework for Contrastive Learning of Visual Representations [1]

ResNet: Deep Residual Learning for Image Recognition [7]

Other Definitions

Epochs: A complete pass of the training set through the model during training.

Iteration: One mini-batch passing through the model during training.

Structure of the Thesis

This thesis has the following structure, which is designed to best guide the reader through the work completed as part of this PhD. Chapter 1 serves as an introduction to the problem space, highlighting the issues within the medical industry that I believe can be helped with artificial intelligence, and why I wished to study machine learning as a possible solution to them. From this Chapter 2 introduces concepts relating to learning from unlabelled data and evaluates the differing approaches that could be taken to the problem case presented in chapter 1.

Chapters 3 and 5 serve as methods chapters, introducing the two contrastive learning approaches that I have chosen to study, but also contain a small literature review of how these methods have been applied to medical tasks, with a strong preference for imaging tasks.

Chapters 4, 6 and 7 each serve as results chapters which describe the experimental setup and results. Additionally, these chapters also contain a small literature review of the work directly related to the experiments undertaken, highlighting gaps in the work, and the importance of conducting these experiments. Finally, each of these chapters contains a text box which contains a statement outlining how this chapter relates back to the aims of this thesis.

In addition to the main body of this work, two appendices are presented. Appendix A contains interesting results that did not fit with the story of the PhD, these may be referred to within the main body of work. Appendix B contains a stand alone chapter which is designed to help the reader understand the Noise Contrastive Estimator.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

From the Emergency Medical Hologram in Star Trek, to the medical droid in Star Wars: artificially intelligent doctors have been prominent in science fiction; however, with very few exceptions, they have not managed to make the jump from imagination to reality. Why is this? Artificial Intelligence (AI) investment has more than doubled year-on-year to $77.5 billion in 2021 [17], and great strides have been made in certain tasks: autonomous cars have driven billions of miles [18], and artificially intelligent voice assistants are widely available to consumers [19] [20]. Compared to this, the application of AI to medical tasks is underwhelming.

A large part of the reason for the growth of AI in certain sectors has been the huge labelled datasets that are freely available. Traditional machine learning datasets were of the order of tens of thousands of images (such as cifar-10 [21], cifar-100 [21] and MNIST [22]), however, in the past decade, these have been surpassed by much larger, richer datasets; most notably by the 1.2 million image ImageNet dataset [23] in 2011. However, these freely-available, large-scale datasets are far less prevalent within the medical imaging field due to ethical and data protection concerns. Because of this limitation, many of the high performing, data-hungry methodologies available in the literature are not able to be applied to the medical imaging field.

One possible solution to this problem is to use a large, unlabelled dataset from which to learn powerful representations. These representations can then be used alongside

a small, labelled dataset to produce far greater performance than could be achieved with the small labelled dataset alone. The main goal of this thesis is to understand whether one class of methods, contrastive learning, can effectively use these large unlabelled datasets to bypass the limitations of small medical datasets. If it can be successfully applied, how should it be implemented to increase the chance of getting the best performance.

1.1 Why do we want to use AI anyway?

To introduce this thesis, I start with *why?*: Why do we wish to create AI that is able to augment the ability of physicians? A number of reasons are presented below:

Medical scarcity: Physician shortages often make headlines in the United Kingdom [24], however, the UK is far from unique on this issue: many countries, including large healthcare spenders such as Germany [25] and the US [26], face the same issues. It is estimated that there is a shortage of 6·4 million physicians globally [27]. This lack of physicians leads to worse health outcomes and increased mortality and morbidity. While current AI methods cannot replace physicians completely, there is scope for AI methods to reduce the workload from each patient, thus allowing a healthcare worker to attend to a larger number of patients without compromising care. In addition, in situations where access to specialists is limited or non-existent, AI methods may be able to increase the competency of an individual physician to allow for much improved care in unfavourable conditions.

Reducing medical mistakes: Misdiagnosis and missed diagnoses are a significant problem which can lead to delayed treatment, thus increasing mortality and morbidity. [28] found 26% of Parkinson's sufferers were initially misdiagnosed, with 48% of these receiving treatment for a condition they did not have, and with 34% reporting a worsening of their health due to this mistake. Cancer Research UK found that the time interval between a patient presenting at a GP with cancer symptoms and receiving treatment was substantially longer for those whose diagnosis was initially missed [29]. AI methods could highlight areas of concern and request a second

opinion. For example, [30] [31] found that cancer detection rates from colonoscopy increased when an software based 'second observer' worked in partnership with the physician. Just as with the previous section, current methods are unlikely to replace physicians, but it is certainly possible that these methods could reduce medical errors [32].

Decreasing the delay at which high priority cases are seen: The speed at which a patient is seen will not have a uniform impact on every patient: for example, a wait of a week would not significantly impact the patient outcomes of someone with cataracts, however, that same week could massively reduce a patient's outcome in the case of an aggressive cancer. When a patient is referred to specialist treatment, a triage decision could be automatically made using AI, based upon the referring physician's request, highlighting the most severe cases for rapid review. Increasing the speed at which some patients are seen, even in a zero sum setting, could increase average patient outcomes.

Decreasing wasted time: An identified problem within medical imaging is as follows: based on a set of symptoms a patient has, a physician orders an imaging study to confirm or rule out the possible diagnosis. For a non-urgent test, the patient may have to wait a number of days for this. The patient then undergoes the imaging study, however, there is an issue with the image produced, the area of concern may not have been fully imaged, the patient may have moved slightly, leading to non-clinically useful images being produced. This mistake may only be picked up once the images are reviewed by the requesting physician, at which point another test must be ordered, restarting the whole process and wasting resources and increasing the time between symptoms and diagnosis. AI techniques may be able to help with this. Just as a smart phone camera can inform the user when the images taken are blurry, current AI methods can be trained to detect non clinically useful images, reducing the rate at which images would have to be retaken [33].

While artificial intelligence may be applicable to a very large number of problems within the medical field, in this work I have specifically examined the problem of medical imaging and how computer vision approaches may be able to solve this.

1.2 Computer Vision

In this thesis, high- and low-level features are referred to extensively: a theory of vision is used in which there are various levels of features within the image, with each successive level using the features of the previous section to build the higher level features. These features can be thought of as low and high frequency features respectively. Low level features refer to features that can be represented by a small number of pixels, for example an edge or a corner. These features are sometimes also referred to as high frequency features (or signals) due to their rapidly changing nature. In contrast to this, this thesis also refers to high level features. These features could be thought of as 'human level' features, features that one could refer to in words, for example a face or a lung. As with the low-level features, these are sometime referred to in terms of frequency. High level features are called low frequency features as they do not change much over the whole image. Generally, in the computer vision field, we are exploring techniques that are able to analyse these 'high level' features. For example, in self-driving cars, we are less concerned about whether there is an edge in a specific place, rather, we care about whether there is an obstacle in the road. Similarly, in the medical imaging domain we are generally interested in macro, human level, features; such as the presence of cancer or lesion. One analogy for how multi layered, neural networks work is through learning successively higher levels of features, with the first layers in a network learning the low-level features, and each layer after that combining these features to create higher levels of abstraction. In this thesis, Contrastive Predictive Coding and SimCLR are presented as methods to enforce that the network learns these high level features, rather than relying on low level features.

Computer vision has a long history: famously being set as a summer challenge by Seymour Papert in 1966 to design a system that could describe an image [34]. Traditionally, computer vision focused on constructing hand crafted features. These hand crafted features do not need datasets to create: they take advantage of the a priori knowledge instilled within them by the designer, for example an edge detector does not have to be learned, it can be manually implemented. Scale Invariant Feature Transform [35] (SIFT) and Speeded Up Robust Features [36] (SURF) are both traditional computer vision methods based upon matching local image features with a dictionary of known features.

While these methods found some success, hand crafting features is a complex task, with a separate design being taken for every imaging task. A different class of machine learning, deep learning, takes an alternate approach. These powerful models are able to learn features directly from the raw data, with no hand crafting needed. By taking an entirely data driven approach, the need for costly domain experts is greatly reduced. This type of machine learning has risen to prominence over the past decade due to its high performance compared to other computer vision approaches. Deep learning networks rely on updating a set of weights and biases which define a set of non-linear functions, to create estimates of underlying latent distributions. These methods were initially developed within the field of neuroscience as a way to model the behaviour of the brain, hence the name 'neural network'.

There have been two major advancements that have allowed deep learning to outperform other computer vision methods: firstly, the introduction of large, open, freely available datasets to train models on; and secondly, the development of Graphics Processing Unit (GPU) based acceleration to make use of the large quantities of data that these datasets gave researchers access to. These advancements together led to the rise in popularity of deep learning [37]. While a large number of the initial deep learning techniques had existed for decades, it was only with the production of these large datasets, along with the compute hardware to train models on them, that the real power of this class of machine learning could be achieved.

Despite the ability of deep learning to produce state of the art (SOTA) results, these methods are extremely data intensive; sometime requiring hundreds of millions of images to achieve this SOTA performance [3]. This is an acceptable trade off in some fields, where labelled data is in abundance, however, this is not true for medical imaging due to the vastly increased cost of collecting labelled data.

1.3 Learning from Small Datasets

In the previous section, it was noted that these deep learning methods are data hungry approaches. This is a known problem within the machine learning community and can lead to a number of problems if one does not have enough data to train the model on. When creating an artificially intelligent model to do a prediction task, it is imperative to train a model that performs well on data which has not been seen by the model previously. This is usually estimated by using a held out test set, which contain images that are not used within the training protocol. When data is limited and the training set is small, the difference in the performance metric between the test set and training set may be much larger than when training with a large training set. This is because, when learning from a small dataset, there are a number of issues that arise that impact the performance of the model on unseen data:

Overfitting: Using an over parameterised model to learn from limited data may cause overfitting. Overfitting occurs when a model learns a decision boundary that too closely matches the training data, at the expense of the ability to generalise. This causes a large gap between the performance metric on a test set and on the training set. Given an infinitely sized training set, overfitting would not be an issue as the training set performance would accurately approximate the test set performance, however, as the training set decreases in size, the approximation of the true underlying distribution of the data may deviate.

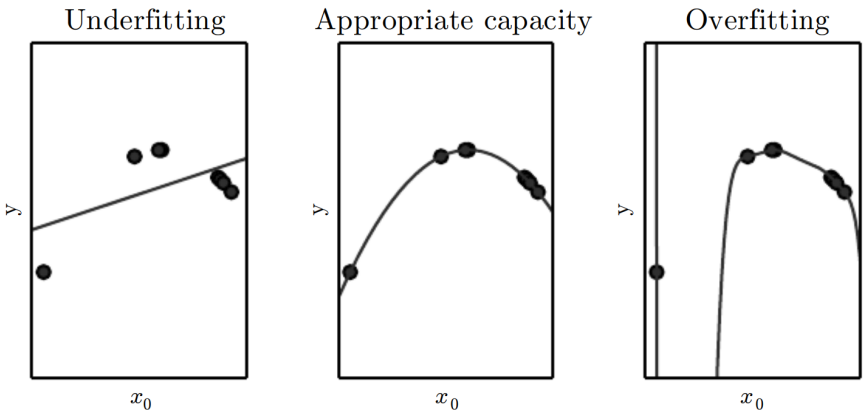

Figure 1.1: Example of underfitting, correctly parameterised, and overfitting on a dataset. Taken from [8].

Out of Distribution Data: When learning from data, the learned representations can only be as good as the data that they are learned from. When learning from limited data, it is possible that the distribution of data used does not accurately reflect the true underlying distribution of the data leading to lower performance on some subsets of the data. This is particularly noticeable for edge cases. Edge cases are "a problem or situation, [...] that only happens [...] in extreme situations" [38]. Within the context of medical imaging, this could be a pathology that sometimes presents differently, e.g serrated polyps compared to normal polyps, or the presentation found in paediatric patients compared with adults.

These highlighted issues are known problems within machine learning and so a number of mitigation strategies have been proposed to counter them:

Augmentation: For a powerful network to achieve good performance, it needs large amounts of data. In situations where large amounts of data are not present, one can 'create' new data by applying random augmentations [37] to the set of training data. This helps to prevent overfitting by increasing the model invarience to the random augmentations trained on. For example, a model trained to predict whether an image is a dog or a cat: the image still contains a dog even if the image has been reflected and rotated. Data augmentation is particularly effective for

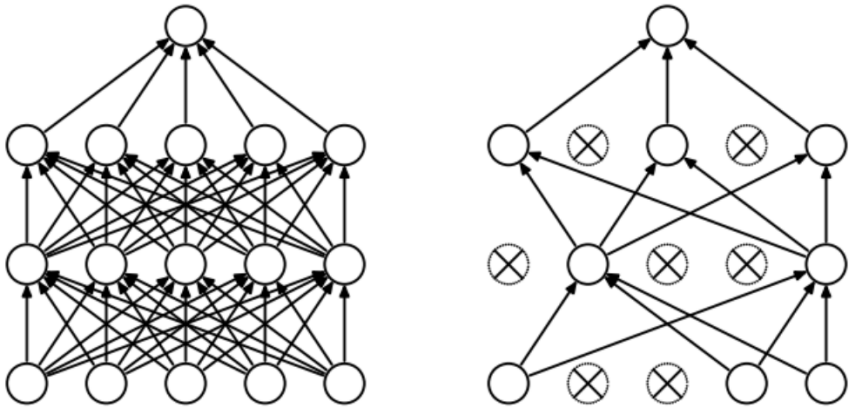

Figure 1.2: Dropout diagram, found in [9]. During training, a random subset of weights is set to 0, which forces the network not to rely on any one, or any one subset of nodes.

object classification tasks, particularly for natural images. Image augmentation will be studied in depth in chapter 6.

Dropout: Dropout [9] is a regularisation technique designed to prevent units from co-adapting too much. During training, the network is randomly 'thinned' (randomly setting weights to zero) to reduce the reliance on certain inputs. This thinning only happens during training. Figure 1.2 shows a visual representation of the method: on the left, a fully connected network in which each node in a layer connects to each node in the next layer. On the right, the network is 'thinned' by 'removing' some nodes. It is important to note that this dropping of nodes is done randomly during each training period. Thus, because the network cannot rely on any one, or any one set of nodes, a more resilient network is learned that is less prone to overfitting.

Early Stopping: Early stopping [37] is a technique designed to reduce the effects of overfitting. When training models with a large capacity on small datasets, over time, the training loss will continue to reduce, however, the performance on unseen data will start to decrease. Early stopping estimates this performance on unseen data by calculating the performance metric on a validation set, not used for training.

After the model's validation performance has not improved for a set number of iterations, known as the early stopping's patience, the method returns the model which has the lowest validation error. This should reduce the chance that the network has overfit on the small dataset.

Transfer Learning: To learn good representations, networks need to be exposed to large amounts of varying data, however, this is not always possible. Transfer learning [39] attempts to reduce the amount of data needed by firstly training a network on a large labelled dataset (which can be somewhat unrelated to the downstream task) and reusing the learned weights in a new network which can be fine-tuned to complete a new task. It is hoped that the features learned on the first task will transfer over to the second task, thus reducing the amount of labelled data needed to achieve good performance. Transfer learning has studied in further detail in chapter 4.

While these mitigation strategies do reduce some of the performance loss associated with small datasets, they do not solve the problem: ultimately, there is only so much information that one can extract from a limited sized dataset. Transfer learning does increase the amount of information available to the model, however, this relies on access to a second labelled dataset that has sufficient distribution overlap with the task that an implementer is trying to solve. This may not always be possible.

1.4 Problems with Collecting Data

While mitigation strategies exist for the issues caused by small datasets, they do not solve all of the problems. So, why can we not just collect more data? In some domains, it is relatively easy to create more labelled data: natural images can be labelled relatively cheaply by untrained people on platforms such as Amazon Mechanical Turk [40]. In contrast, medical images have to be labelled by highly

specialised physicians. In addition, ethical considerations must be made about releasing private medical records, to ensure that any data released cannot be traced back to the individual. The most costly part of data collection, especially for medical imaging is the collection of labels. There are a number of ways in which labels can be generated for use with a machine learning algorithm, however, they each come with drawbacks that may reduce their utility:

- **Manual labelling:** In this method, physicians are given medical images and are asked to give their diagnosis. This process generally gives quite good results, however, it is very costly due to the highly specialised nature of these experts. Additionally, there can also be disagreement between physicians (as would be the case in a real clinical setting), and a design decision must be made for how to deal with this. A standard approach [23] to solving the disagreement would be for multiple physicians to examine the same image, and use a majority voting system, however, this increases cost by multiple times. In addition to the increase in cost, majority voting does not ensure that the correct answer has been found, it just reduces simple errors.
- **Data Mining:** When a patient is undergoing treatment where an imaging study is conducted, this imaging study will (most often) be examined by a radiologist, who will write a report. These reports can then be data mined to attempt to find the diagnosis provided by the radiologist. While this is a far cheaper solution than manual labelling, it will decrease the accuracy of the labels, which will result in a machine learning algorithm being less accurate.
- **Outcome Based Labelling:** Doctors make mistakes. A user should, therefore, not treat labels provided by them as truth. A different type of labelling process works by matching images with their outcome in a post hoc way, this producing as close to ground truth as possible. This helps the labelling issues raised by manual labelling, and decreases cost. However, this method is limited to certain types of outcomes, and is not as fine grained for certain diseases. Therefore, this method may not be suitable for all tasks that we wish to apply machine learning to.

Due to all of these methods' inherent problems, because of either cost or inaccuracy, it is often easier to acquire large volumes of unlabelled data than a smaller quantity of labelled data. If methods could be developed that use large quantities of unlabelled data to reduce the amount of labelled data needed, machine learning could be applied to a larger number of problems.

One proposed way of doing this is semi-supervised learning. Under a semi-supervised paradigm, a representation is learned from a large, unlabelled dataset by conducting some unsupervised task. This representation is then used, along with a small, labelled dataset, to produce (hopefully) higher results than could be achieved with the small dataset alone.

1.5 Problem Summary

Due to the especially large cost of acquiring labelled data, the medical domain is well placed to be revolutionised by the recent advancements in semi-supervised learning. In this chapter, the following problem description is introduced:

- Medical imaging has a number of problems, such as medical scarcity and medical mistakes (section 1.1), that could hypothetically be solved or reduced by artificial intelligence.
- Computer vision techniques have been developed in the past, however, these methods have found limited success in being applied to medical tasks. These methods are typically data driven methods (most notably, deep learning) and thus require very large labelled sets of data to perform well.
- Techniques have been developed to reduce the amount of data needed to achieve high performance, however, these do not increase the performance to the level found with large datasets.
- While a huge amount of imaging studies take place, it is very costly to get gold standard labels for these images. This reduces the ability of state of the art methods to be applied to medical imaging tasks.

- If methods could be developed that take advantage of the relatively cheap, unlabelled data; the success of deep learning could be applied to more medical imaging tasks.

Based on this problem specification, I present the following work: chapter 2 introduces the concept of learning from unlabelled data, highlighting contrastive learning as a possible solution to the issues discussed here. Chapters 3 and 4 introduce and evaluate a method called Contrastive Predictive Coding across a number of medical imaging tasks. Chapter 5 introduces a second contrastive method: SimCLR. In chapter 6, the impact of augmentation strategy is tested to its limits within the context of medical imaging. Finally, in chapter 7, the impact of the unlabelled dataset on downstream classification performance is tested.

1.6 Datasets used

In this thesis, a number of imaging modalities have been used as example datasets to test the presented methods on, namely: a colonoscopy dataset, an Optical Coherence Tomography (OCT) dataset, and a dermatology photography dataset, in addition to some general imaging datasets similar to ImageNet [41]. In this section, the datasets are outlined, the pathologies are explored and the physics behind the imaging modalities is presented. Finally, for each of the sets of diseases, speculation on why semi-supervised approaches may be suited for these tasks is given.

1.6.1 Colonoscopy

Colonoscopy is an imaging study which allow a physician to examine a patient's bowel. This allows visual inspection of a problem area, or allows a physician to undertake a procedure such as biopsy or removal of a possible cancerous lesion. Colonoscopy is often used to diagnose colon cancers, or to identify pre-cancerous lesions, both of which are a leading cause of deaths in both males and females. A colonoscope traditionally consisted of a fibre optic cable which allowed a clinician to look inside of the colon. More recently, due to improvements in camera technology,

a small camera can be placed on the end of the flexible and controllable endoscope.

In this thesis, focus has been placed on the identification of polyps within colonoscopy images. Polyps are a precancerous, mushroom looking lesion that is often difficult to identify in colonoscopy screening procedures. As many as 1 in 4 adults over the age of 50 have bowel polyps [42] and the misidentification of polyps has been linked to higher mortality. Therefore, identification of polyps is important. The identification of polyps can be complicated by the variability in appearance, with some types (such as flat polyps) being notably more difficult to detect. In addition to polyps, in chapter 4, this thesis also attempts to classify a different pathology that can be identified with colonoscopy: colitis. Colitis is a chronic illness in which the walls of the bowls become inflamed, which can cause pain, discomfort and diarrhea. This thesis uses various grades of colitis, with the more advanced stages being easier to identify than earlier.

For this imaging modality, a randomly sampled subset (with equal numbers in each class) of the HyperKvasir dataset [10] was used. This dataset used frames from colonoscopy procedures collected as routine scans at a Norwegian hospital. Some images contain a green box in the bottom left-hand corner which is used by the radiographer to assist with the procedure, however, they are not in all images. To ensure that this does not have an impact on performance of the network, this area is blanked over in all images.

Colonoscopy procedures are well suited for use with semi-supervised networks due to the large volume of unlabelled data that the procedure produces. Each colonoscopy can take on the order of an hour to complete producing a large quantity of images (i.e each frame in the image can be treated as a separate image from which to learn representations from). If useful representations could be learned, then a small quantity of labelled data could be used to produce higher results than could be obtained with the labelled data alone, for limited extra cost.

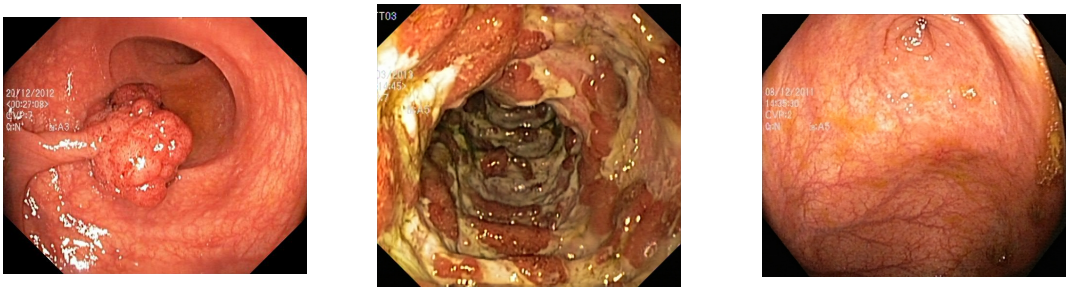

Figure 1.3: Colonoscopy example images, from left to right: Polyp, Colitis, Normal Cecum. These examples were chosen as extreme examples of the pathology, therefore, are far easier to distinguish than the typical images. These images were taken from the HyperKvasir dataset [10].

1.6.2 Optical Coherence Tomography

OCT is a three dimensional imaging modality used to take a tomography scan of the eye. This can then be used to diagnose diseases of the eye such as diabetic retinopathy or macular oedema. The OCT scan can image below the surface of the retina, allowing for earlier diagnosis of certain eye diseases. By allowing earlier diagnosis, the earlier that treatment can be started leading to better patient outcomes.

Certain eye pathologies create lesions below the surface of the eye, and so standard techniques of examining the eye, such as a Retinoscope, are insufficient for a full diagnosis. In addition to providing advantages over standard equipment such as a Retinoscope, OCT is able to provide much greater resolution than MRI at the expense of a more limited area of view. In addition, OCT scans are extremely quick to complete an imaging study, taking approximately six seconds to take a full scan compared to 10-30 minutes for methods like fluorescein tomography [43]. OCT is sometimes thought of as being similar to ultrasound scanning, using light rather than sound. In an OCT scan, a beam of light is shone at the back of the eye and the reflections of light are recorded. Interferometry is used to infer which beams of light bounced around the eye, which can then be subtracted from the image. The scanning laser is able to take a number of one dimensional scans of the eye (known as a-scans) at various depths to produce an imaging scan of a single slice of the eye (known as a b-scan). Multiple b scans can be taken of subsequent parts of the eye

to produce a three dimensional volumetric image. This is similar to how a CT scan takes images of slices of an object which can be taken multiple times on subsequent slices to produce a three dimensional representation. These images are used in conjunction with other imaging modalities to give a diagnosis.

In this thesis, an OCT dataset taken from [11] is used which contains the following classes:

- Choroidal Neovascularization is a continuation of a pathology known as age-related macular degeneration in which new blood vessels develop under the retina. This pathology can cause vision blurring and over time is able to damage the retina, leading to permanent sight loss. OCT scans are able to be used to identify the fluid filled areas beneath the surface of the retina, and therefore diagnose the condition which can then be treated with injections to the eye [44].

- Diabetes-related Macular Oedema is a complication of a pathology known as diabetic retinopathy. Diabetic retinopathy is caused when a patient has high blood sugar over a long period of time, this causes changes to the blood vessels at the back of the eye and bleeding can occur [45]. If left untreated this can further develop into Diabetes related macular oedema, a serious complication which affects the macula, and area of the eye that is responsible for "central vision" [46]. OCT can be used to measure the thickness of the macular, thus determining the extent of the damage caused by the disease.

- Drusen are small accumulations of lipids in the eye, that build up over time. While a small number of drusen are found in most people [47], as we age these deposits can grow in both size and number which may affect "central vision" [48]. As these deposits grow, they can damage the macula leading to partial vision loss. While more commonly identified through routine eye exams, they are also able to be identified through OCT scans.

- No pathology present: In addition to the three pathologies presented above, a class of images that contain no pathology is also included.

As with colonoscopy, OCT scans produce a large volume of unlabelled data that is able to be used for learning representations from. Each OCT scan produces a 3D volume, each slice of which can be treated as a separate image and can be used to train a network to produce representations. These representations along with a few examples of each pathology could produce better results than the few examples alone.

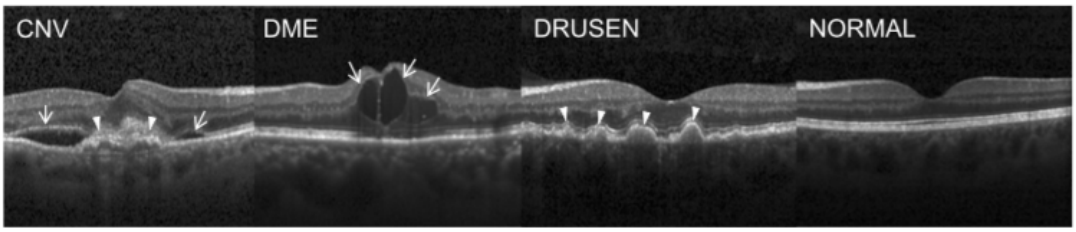

Figure 1.4: Example OCT images, from left to right:CNV, DME, Drusen, Normal (no pathology detected). Image taken from [11]. OCT scans produce 3D scans of the retina of the eye, these images consist of slices of this 3D representation which show the pathology (or not in the case of normal).

1.6.3 Dermatology photography

The final medical imaging dataset that is used in this thesis is a dermatology photography dataset consisting of photographs of skin abnormalities arranged in three classes: Benign lesions of the keratosis; Melanoma; and Melanocytic nevi. Dermatology photography can be used for either tracking the progression of an abnormality, or for telemedicine, allowing diagnosis of a disease without a specialist being present. This can increase the speed at which patients can be seen, with the patient only needing a single appointment to photograph the abnormality. The specialist may not even be in the same country as the patient being seen, this could be especially useful for accessing medical in remote regions of the world, increasing health outcomes and increasing health equality. The following pathologies are explored:

- Benign lesions of the keratosis: A generic class of non-cancerous lesions. They are grouped together due to their similar appearance. This class is included due to the difficulty in distinguishing between these and melanoma.

- Melanoma is a form of skin cancer that is capable of spreading to other parts of the body. Melanoma is thought to be caused by excessive exposure to UV radiation, which can cause damage to the DNA of skin cells. Melanoma can be extremely common, 1 in 14 men in Australia will develop melanoma at some point in their lives [49]. There are a number of sub types of melanomas that have differing appearance, which may affect diagnosis. As with keratosis, no specialist imaging device is needed to capture images of melanoma.

- Melanocytic nevi is the technical term for a skin mole [50]. Identification of this is especially important to be able to differentiate between a harmless skin condition and a condition that can possibly lead to death, such as melanoma. As with melanoma, this pathology may present in many different forms which can make it harder to differentiate between conditions, possibly leading to lower health outcomes.

Unlike the previous two datasets, in which a large number of images are generated during routine procedures that are able to be used for unsupervised training, dermatology photography does not have the same advantage. However, in this thesis, images of pathologies that are not part of the classes used for our training dataset have been used as part of the unlabelled dataset for optimising the encoder.

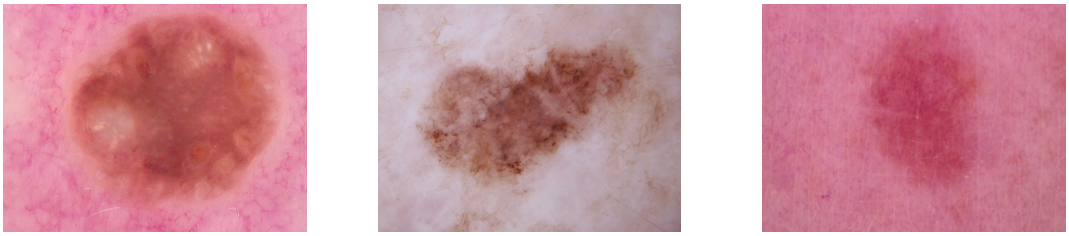

Figure 1.5: Example dermatology images, from left to right: Benign lesions of the keratosis; Melanoma; and Melanocytic nevi. These images are taken from the HAM10000 [12] dataset.

1.6.4 General imaging datasets

In addition to the medical imaging datasets used in this thesis, a number of general imaging datasets have been used. These datasets were chosen to provide domain

agnostic results across the work in chapters 6 and 7. It is hoped that the results found in these chapters, while especially relevant to medical imaging tasks, would also be relevant to any arbitrary task. In this thesis, two main general imaging datasets are used: subsets of the ImageNet dataset and the STL-10 dataset.

ImageNet: the ImageNet dataset is one of the most widely used imaging datasets, consisting of 1,281,167 training images across 1000 classes. Due to its large size, in this thesis, various sized subsets of the dataset are used to increase the speed at which training can take place, the exact process by which these subsets are generated are detailed in the individual chapters.

STL-10: The STL-10 dataset is a predefined subset of the ImageNet dataset commonly used for evaluation of unsupervised and semi-supervised networks. It consists of 100k unlabelled images used for learning representations from, and 13k labelled images arranged in 10 classes between training and test subsets, with 500 to 800 in each.

1.7 Aims of the thesis

The aim of this thesis is to study a set of semi-supervised, contrastive learning methods to understand (1) how they work, (2) their strengths and weaknesses and (3) to investigate how these methods could best be applied to the task of medical imaging. Medical imaging has unique challenges that have been described in the rest of this chapter, however, they often relate to the huge cost in acquiring labelled data to train the state-of-the-art models that are available in the literature. In this thesis, contrastive methods are proposed as possible techniques that could mitigate this challenge. In addition, work is undertaken to further investigate the cost-performance trade off that occurs when one changes from a standard supervised to semi-supervised approach. Based on the work presented here, chapter 8 gives a set of recommendations for how best to apply contrastive methods.

How chapter 4 relates to the aims of the thesis: Chapter 4 presents an eval-

uation of Contrastive Predictive Coding within the context of a medical imaging task. The literature [2] has shown that CPC can reduce the amount of labelled data needed for a network to achieve high performance, but it is yet to be seen whether this improvement will transfer to a medical imaging task. In addition, the effects of domain shift and transfer learning are investigated. The effect of these directly affect the cost, and therefore utility, of semi-supervised methods. The use of transfer learning can dramatically reduce the training times of large neural networks, thus enabling performance to be achieved that would be uneconomical with training from scratch. This relies on the distribution of features learned during pretraining to closely match the distribution of features found in the downstream task. By showing a high level of performance on a transfer learning task, this chapter validates one of the most important use cases of semi-supervised learning: training once on a large dataset, followed by cheap finetuning for task specific applications. Despite the initial promising results, this improvement in performance was not found to carry over to all tasks.

How chapter 6 relates to the aims of the thesis: Chapter 6 evaluates SimCLR as an alternative to CPC: while chapter 4 finds that CPC is able to improve abnormality detection performance on colonoscopy images, this improvement did not translate over to all modalities under study. SimCLR has the same theoretical benefits as CPC, with more positive results from the literature. This chapter begins to investigate aims 1 and 2 of this thesis: thereby increasing our understanding of the methods presented. This chapter evaluates the usage of augmentations in contrastive approaches, investigating how best to apply them to achieve the greatest performance. An additional investigation is conducted to show whether the internal, learned representations are truly invarient to augmentation as claimed (aim 1).

How chapter 7 relates to the aims of the thesis: Chapter 7 continues this secondary investigation into how best these contrastive methods should be applied, specifically investigating the dataset requirements for these methods. Outside the realms of academic research, datasets have to be created and this process is not cost free. It is therefore imperative that the wider community understands how

the design choices of this dataset will affect the performance of the subsequent network, to achieve the most cost optimal solution. Initially the size of the dataset is investigated, showing that merely increasing the size of the dataset is not enough to increase performance, this must be combined with an increase in training time (aim 1). To gain the best performance (aim 3), this thesis proposes using early stopping in combination with the longest training time as an implementer's budget will allow. This chapter also shows that the variety of data used is not important for the downstream classification performance of a SimCLR network, unlike what would be found in supervised learning, the learned features are more general.

Chapter 2

Learning from Unlabelled Data

In chapter 1, the issues with applying state of the art deep learning methods to medical imaging tasks were explored, concluding that - while AI systems could improve healthcare outcomes - it is often uneconomical to generate the large, labelled datasets needed to train the state of the art machine learning methods. However, also highlighted was the fact that unlabelled data is often orders of magnitude cheaper to collect. If it was possible to utilise this unlabelled dataset to increase the efficiency of the labelled data that is available, machine learning could be applied to more areas.

While some large scale labelled datasets exist, notably ImageNet [23], some unlabelled datasets have been released which are three orders of magnitude larger in scale. Commercial image datasets held privately by companies may be far larger than this, with Google photos holding 4 trillion images as of 2020 [51]. This shows the enormous scale at which unlabelled data can be collected. Not all of this data may be able to be used, but it does highlight the relative ease in which unlabelled data can be collected. This data could be utilised to pre-train a model to allow it to learn powerful representations, which could then be used to reduce the amount of labelled data needed to train high performing models.

This chapter summarises and evaluates approaches that could be taken to utilise this large quantity of unlabelled data. Initially, a naive approach is introduced: labelling

the unlabelled data. From there, a redefinition of the problem space is presented, one that can make use of the unlabelled data itself, however, these methods are shown to not be suitable for all tasks. Therefore, representation learning methods are presented before, finally, presenting contrastive learning. Contrastive learning is a subset of representation learning methods in which a contrastive loss is used. This directly optimises the encoder to place 'similar' images together in the latent space and 'dissimilar' images far apart.

2.1 Increasing the Number of Labels Available

Under a situation in which an implementer has a small amount of labelled data and a large amount of unlabelled data, the most natural approach would be to label all or some of the unlabelled dataset. Depending on the specifics of the task, different methods could be taken.

Manual labelling: The most conceptually simple approach could be simply to label the unlabelled data, thus creating a large labelled dataset. It is well established that larger labelled datasets produce better results [52], and it is likely that this approach would work if the goal is purely to increase performance of the model, with no other considerations. However, the main disadvantage of this approach is the cost. As presented in chapter 1: labelling data is costly. Fortunately, methods have been developed that allow for this unlabelled data to be leveraged without the high cost associated with labelling.

Pseudo-labelling: Pseudo-labelling [53] is one such method. Pseudo-labelling is a simple approach for improving performance when given small amounts of labelled data but large amounts of unlabelled data. In this method, rather than manually labelling the large unlabelled dataset, an automated approach is taken. Initially, a network is trained to classify the images on the small amount of labelled data. This trained network is then used to generate labels for the large unlabelled dataset, creating a set of pseudo-labels. This set of pseudo-labels is then used to train a second network which, it is hoped, will have a higher performance than the first network.

This approach's efficacy will depend entirely on the ability of the first network to correctly learn to classify the unlabelled dataset. If a poor classifier is trained, the pseudo-labels will be extremely noisy, leading to poor performance. In addition, pseudo-labelling can lead to low performance on edge cases, where the first classifier did not correctly label the data.

Active Learning: The previous two methods attempt to increase performance through increasing the number of labelled examples through brute force. Increasing the number of labelled examples increases the probability that a query is similar to an item already seen by the network, and therefore it is more likely to produce the correct answer, however, this is a crude process in which exponentially increasing the amount of data is only likely to linearly increase the performance of the network. This can be mitigated somewhat through collecting more data for sub-populations; however, this requires thorough investigation of the network, and for an implementer to have access to labels on which sub-populations each datapoint belongs to. Another approach is active learning: active learning attempts to increase the power of the network while labelling as few data points as possible [54]. Active learning attempts to find an optimal subset of a dataset, for which learning on from this subset would approximate the performance of learning from the full dataset, therefore, saving the cost of labelling the full dataset [55]. Active labelling is an active area of research, with some promising results, however it is beyond the scope of this thesis to study.

While these methods may seem naive, labelling the unlabelled data may be the most time and cost effective approach to take for some tasks, particularly those tasks in which the labelling task can be undertaken by non specialists. Relatively large, 100k image scale datasets can be created for only a few thousand US dollars in a short amount of time [40]. While this thesis will focus on areas in which this does not apply, (namely the medical domain, in which the costs and ethical considerations would be far greater), there will be a large number of problem areas in which this is the case. Before exploring whether semi-supervised learning could be right for

a task, it is worth being cognisant of whether the same result can be obtained by manually labelling the dataset.

2.2 Redefining the Problem

In situations where it is uneconomical to label more data, and automated labelling produces sub-par results, a different approach may be taken. It is not always necessary to develop a system that can classify an image, it may be sufficient to identify when an image contains an abnormality. This is known as out-of-distribution detection. For example, in chapter 1, "reducing medical mistakes" was highlighted as one area where AI could be used for improving healthcare outcomes. An AI could be trained to detect abnormalities for secondary review. Even if it cannot predict the specific abnormality, flagging the image for human review could be enough to improve health outcomes. Two different approaches for out-of-distribution detection using different base methods are presented below.

Generative Adversarial Networks: Generative Adversarial Networks (GANs) were first described in 2014 by deep learning pioneer Ian Goodfellow [56]. In this work Goodfellow et al propose a minimax game (a zero sum game, where one 'player's' loss is the other 'player's' gain) in which one player tries to generate images that the other player cannot discriminate from real data. "The generative model can be thought of as analogous to a team of counterfeiters trying to produce fake currency and to spend it without detection, while the discriminative model is analogous to the police, trying to detect the counterfeit currency" [56]. After this training, the discriminator part of the network is discarded, leaving just the generator. The trained generator can generate realistic images and is used in the next step of the pipeline to detect abnormal images.

One method for using a GAN for abnormality detection is the work by Schlegl et al [57]. This method uses a GAN to learn a latent space that represents the normal variability of non-abnormal data, from which, it can be determined whether a query datum lies within this latent space, thus, whether it is abnormal or not. This is

achieved through gradient descent on the similarity between generated data and the query data, using the discriminator to ensure the latent space is on the manifold. This allows the closest image in the latent space to be found. The difference between the query image and its closest match on the manifold is found, giving its abnormality score. If an image lies on the manifold, it will have a very low score, as the closest image will be very similar. However, if the query image is nothing like the manifold, as in the case of an abnormality, the difference between it and of any image on the manifold will be high, giving a high score.

Autoencoders: An alternate method is to use autoencoders. Autoencoders are models that attempt to learn to compress, and subsequently reconstruct, data without explicit labels: in the process learning a latent representation of the data. As with all semi-supervised methods, the autoencoder can be trained on the vast quantities of easily available unlabelled data. Autoencoders are typically thought of as having two sections: an encoder which projects the input down to a latent representation; and a decoder, which attempts to reconstruct the image from the latent embedding produced in the previous step. A loss based on the difference between the reconstructed image and the true image is used to optimised the two sub-networks.

In a similar way to the method introduced by [57], the reconstruction loss can be used as a method for detecting out of distribution images [58]. The autoencoder is able to reconstruct the distribution of images that it has seen before. When it encounters a query image that is dissimilar to the images that it has seen before, the ability of the network to reconstruct it is diminished, thus the reconstruction loss will be higher. A threshold can be applied to the loss to detect when an image is out-of-distribution.

While these methods may excel at detecting when a specific example does not fit within the distribution of the previously seen images; these methods are limited to one task – detecting out of distribution images. In many cases, merely being able to detect when a specific data point does not fit the training distribution is not

enough: further analysis is needed. For example, in the case of a triaging system that automatically flags anomalous OCT scans, one could imagine a case in which a large proportion of scans are anomalous (as generally a scan will only be performed if a physician feels there is something wrong). In that specific case, one would want a system that could not only flag scans that are anomalous, but can flag the subset which are likely to deteriorate rapidly, and therefore need to be treated faster than other diseases. In the next section, other methods are presented which allow for any arbitrary task to be performed.

2.3 Auxiliary Task Methods

In the previous two sections, two different conceptual frameworks for utilising unlabelled data were presented. Neither approach solves the issues presented in chapter 1 for all tasks. A third approach to this is semi-supervised learning. In semi-supervised learning, the unlabelled dataset is used to train an encoder to project the data into a more efficient space. The small amount of labelled data can then be used with these embeddings to produce higher performance at a lower labelled data cost. This section examines a subset which I term 'Auxiliary Task Methods' [1], that is, a set of methods that try to perform some upstream task to learn an embedding that is useful for some downstream task.

Autoencoders: In addition to the out-of-distribution approach, based upon reconstruction loss; autoencoders can also be used as semi-supervised learners. The network is trained in the same way as in section 2.2, however, after training the 'decoder' section is discarded. This leaves the 'encoder' portion of the network to project the input query image into the learned latent space (Figure 2.1 for a di-

[1] Throughout this PhD, I have used the term 'Auxiliary Task Method' to refer to the type of semi-supervised task in which an unrelated task is used to train an encoder to create useful embeddings. I highlight here that this differs from the other usage of the term auxiliary task within machine learning. This alternate usage refers to methods that have a second task that they are performing at the same time as the primary task, usually to improve performance of their primary task.

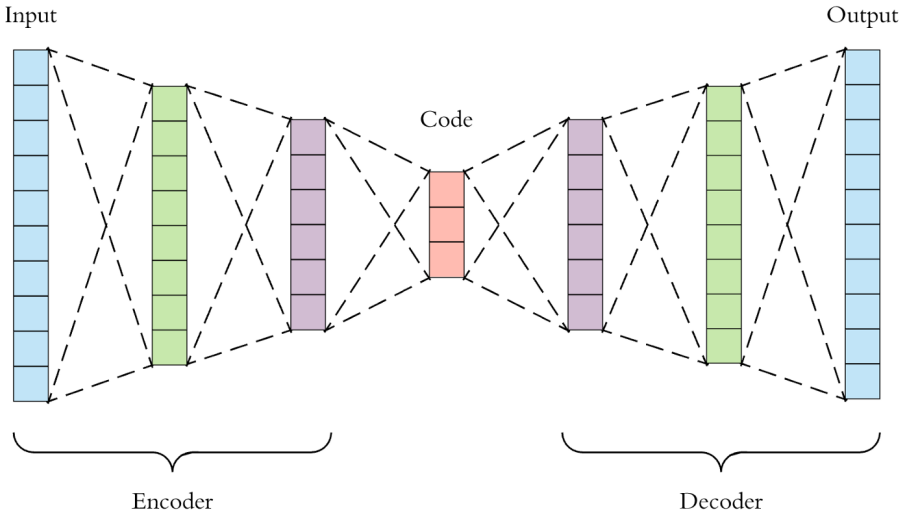

Figure 2.1: An autoencoder network: successive (usually convolutional) layers compress the input down to a latent space, from which, the network must attempt to reconstruct as much information as possible. Image taken from [13].

agram). This latent representation of the data can then be used by a secondary learning mechanism for performing some task, such as image classification.

Autoencoders suffer from a problem that can be seen reoccurring across section 2.3: the features that are optimal to be learned to complete the pretext task are not necessarily the same features that will be useful for the downstream task. For example, in a reconstruction loss the low level information that may not have much to do with the classification of the image is equally weighted to any other pixel value, one that may be very important to classification.

Rotation Prediction: Rotation prediction is a simplistic, auxiliary-task based semi-supervised learning method. In this method, an image transformed through a rotation, is fed to a network. The network must then predict which of the four cardinal directions the image has been rotated. While a simple task, this pretext task is still able to learn reasonable representations, for example [59] found that using a simple rotation prediction sub task was able to match the performance of the much more complex Contrastive Predictive Coding [6] pretraining, found later in the chapter.

While some success has been found; this methodology may not work for all tasks, such as tasks in which the features are rotation invariant. In addition, it is possible that the network learns to solve this task through learning features that are irrelevant to the downstream task. Imagine training a network on images of various dog breeds with the intent to use the embeddings to classify images into their respective breeds: however, the network could possibly learn to just predict the position of the sky as a proxy for rotation of the image. However, this feature would not be useful at all for predicting the breed of the dog. Thus the performance on the downstream task would be low.

Jigsaw: Noroozi and Favaro [14] propose using a 'jigsaw puzzle' as a pretext task for an unsupervised network. In this task, the network is fed shuffled patches of an image, and it is trained to predict the correct positions of the said patches. They have achieved 37.6% (top-1[2]) accuracy on ImageNet when only adding fully connected layers. A diagram of the method can be found in Figure 2.2. As with other auxiliary task methods, the ImageNet performance of this network is far below that of supervised networks. This is possibly because the network is able to learn features that are useful for the pretext task, but are less useful for the subsequent downstream task. For example, in general the downstream task will focus on high level (section 1.2), human level features, such as whether the image contains a cat or a dog, however, it is possible that the network has learned to solve the pretext task through features that are less relevant to the downstream task, such as colour distribution and continuity of edges.

[2]In this thesis, a number of references are made to "Top-x performance / accuracy", usually either top-1 or top-5 accuracy on the ImageNet challenge. Top-5 accuracy refers to the accuracy where the correct value appears as one of the 5 highest probability classes for a model, whereas top-1 accuracy refers to the normal accuracy of the model.

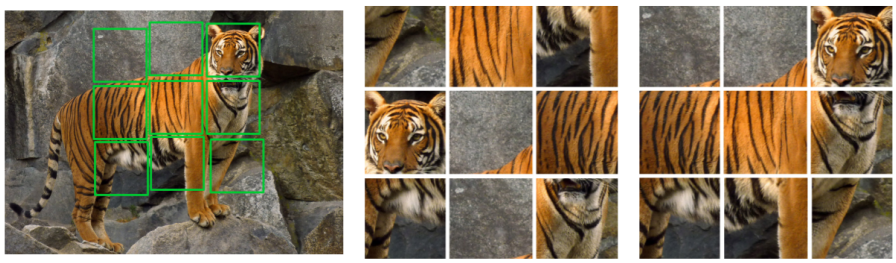

Figure 2.2: Figure found in [14]. A visual guide to the jigsaw pre-training method. In this method, patches are jumbled up, with a network predicting how to solve the 'jigsaw' that is given to it.

Context Prediction: Doersch [15], describe a similar method to jigsaw. Two images are fed to the network in parallel, a context image (A) and a query image (B). The network is trained to discern the relative position of B in relation to A. During training, a random patch of an image is taken (blue in Figure 2.3), along with one of eight neighbouring patches (red in Figure 2.3). This method relies on a double headed encoder with shared weights between the encoders to embed the two patches down to two latent encodings. From here a multi layered perceptron is used to predict the location of the query patch relative to the context patch. After unsupervised training, the encoder can be used to embed whole images down into the same latent space, and train a linear layer on these embeddings for prediction. This method achieved 45.7% Mean Average Precision on VOC-2007.

Context prediction suffers from the same issues as the jigsaw pretext task: the network may rely on features that are not useful for the downstream task to solve the pretext objective. The paper claims that this pretext task will force the network to learn a "a rich visual representation", however, this may not be the case, with the network being able to rely on the same set of non optimal features to solve the task. This issue was addressed in the paper: the authors included a gap between patches and randomly jittered each patch to attempt to reduce the impact of this trivial solution. However, they found that even with these mitigations the network was still able to find trivial solutions. As a second set of mitigations the authors explored using a colour projection and colour dropping, reporting the resuts of both.

Figure 2.3: Context prediction. Image from [15]. Here, a dual headed network encodes two image patches: a context patch (blue) and a query patch (red). The model is trained to predict the correct relative position of the query patch in relation to the context patch.

Despite the efforts to avoid trivial solutions, the network still performs substantially below what is able to be achieved with supervised networks.

Colourisation: Colourisation [60] is another proposed unsupervised methodology for learning to embed images. In this method, an autoencoder style network is used, which takes in an artificially greyscaled image, and attempts to reconstruct the coloured image. The latent embedding from the middle of the autoencoder can then be used as the embedding for the image. This method would not be a suitable method for use on medical images that are inherently greyscale , such as x-rays, CT scans, or OCT images. In addition, the features that are useful for the colour reconstruction task may not be helpful for some downstream tasks, for example tasks in which colour does not have an impact on class.

Inpainting: A method that builds upon the work found in Doersch et al [15] is the work presented in [61]. Rather than learning context through a single classification problem as presented in Doersch, inpainting presents a prediction problem with a much larger set of predicted values. In inpainting, a whole block of pixels are blanked over in an input image and an autoencoder style must predict the missing pixel intensities, resulting in a much more complex problem. This will hopefully lead to much better image representations learned by the encoder portion of the

network. Despite the optimism presented in the paper, ImageNet classification performance is still lower than can be found with supervised approaches. As with other methods presented in this chapter, the optimal weights for the encoder to solve the pretext task may not be the optimal weights to solve a downstream classification task. The objective function presented in [61] combines an l_2 loss of the pixel intensities along with an adversarial loss which improves the infill quality. I posit that while there is overlap between the features that are relevant for both tasks, they will be non-optimal. As with the autoencoder, minor differences in pixel intensity may make large differences to the l_2 loss, but fail to capture any high level detail that would be useful for a downstream task. In addition, while the adversarial loss is useful to produce better looking infilled images, this does not necessarily create better embedding spaces.

2.3.1 Discussion on Auxiliary tasks

Semi-supervised learning approaches have been touted as important areas for study due to the belief that learning features from large, unlabelled datasets will lead to better performance than learning from a relatively small, labelled dataset. The methods presented in this section have failed to outperform supervised approaches in their studies in the evaluation set up of the papers. The relative performance of the semi-supervised vs supervised will be dependent on the specific experimental conditions of the test being conducted; this experimental set up may not lead to an advantage being seen in the semi-supervised set up. A typical approach for evaluating semi-supervised methods involves training an encoder network on a set of unlabelled images, freezing the weights of the network, and evaluating the ability of the network to produce linearly separable classes. This evaluation methodology is different from how supervised approaches are evaluated, which may explain some of the performance drop.

While some success has been found in the literature [62] improving upon a supervised baseline; the low relative performance of semi-supervised approaches seen in

this chapter is disappointing. In this section, I have argued that this low relative performance is due to the networks learning low-level features that do not map across between the pretext and downstream tasks. It highlights that further work is needed to explore new methods for semi-supervised learning that could possibly outperform purely supervised approaches.

The goal of this chapter is to explore methods that can utilise the learned latent space to boost the performance of these deep learning methods above what would be possible with supervised approaches alone. If these methods are unable to outperform supervised baselines, they should be discarded as possible approaches for the task set out in chapter 1. From this argument, the next section introduces contrastive learning, a set of approaches that have found greater performance than traditional approaches, through trying to learn a latent space that more accurately embeds high level, human level features.

2.4 Contrastive Learning

In order to explore semi-supervised methods that could possibly outperform supervised methods, this section introduces contrastive learning, a set of related methods in which the latent space of an encoder network is directly optimised. Contrastive learning learns to place 'similar' elements together, and 'dissimilar' elements far apart. It does this without the use of explicit labels for what constitutes 'similar' and 'dissimilar' elements.

It can be argued that all machine learning methods learn to place similar elements together in their latent space, however, these methods typically require the use of explicit labels. For example, a network trained to classify images of cats and dogs will naturally create a latent space that separates cats and dogs, but only because it has been provided with these explicit labels. In contrast, contrastive learning is able to create this latent space without this set of labels, optimising the latent space on the images directly.

Contrastive methods also differ from the auxiliary task methods found in the previous section. The auxiliary task methods presented produce latent spaces that are useful for conducting the pretext task, which may or may not be useful for the downstream task. Contrastive approaches hope to learn a more general latent space that is useful for a large number of tasks, however, this is dependant on a number of assumptions. This is expanded upon later in this section.

Once the encoder is trained, it can be used in the same way as methods found in the previous section: a linear layer can be added to the encoder network and then finetuned on a task specific dataset. As with the methods presented in the previous section, it is hoped that the use of this unlabelled dataset will improve the downstream performance. This section presents a number of contrastive approaches:

2.4.1 Contrastive Predictive Coding (CPC)

This method is described in further detail in chapter 3.

Contrastive Predictive Coding [2] is a contrastive method to learn representations of patches of images which will then be able to be utilised in a second step, reducing the amount of data needed to reach a certain performance. The authors of [2] claim that using their modification of CPC is able to reduce the need for labelled data by 80%, replacing it with unlabelled data. CPC works by applying the contrastive loss function to patches within the same image. Under this construct, an encoder learns to place patches from the same image closer together in the latent space, while also pushing patches from random images further apart. By not directly training the model for one particular task, it is hoped that the embedding that the method learns will be applicable to a number of tasks, such as classification and segmentation.

Figure 2.4 shows the reported results from [2] showing that CPC can either achieve the same performance as a ResNet baseline using five times less data, or can have a much increased performance at the low data regimes. The performance gained on a general imaging task will not always transfer over to medical imaging tasks. Despite

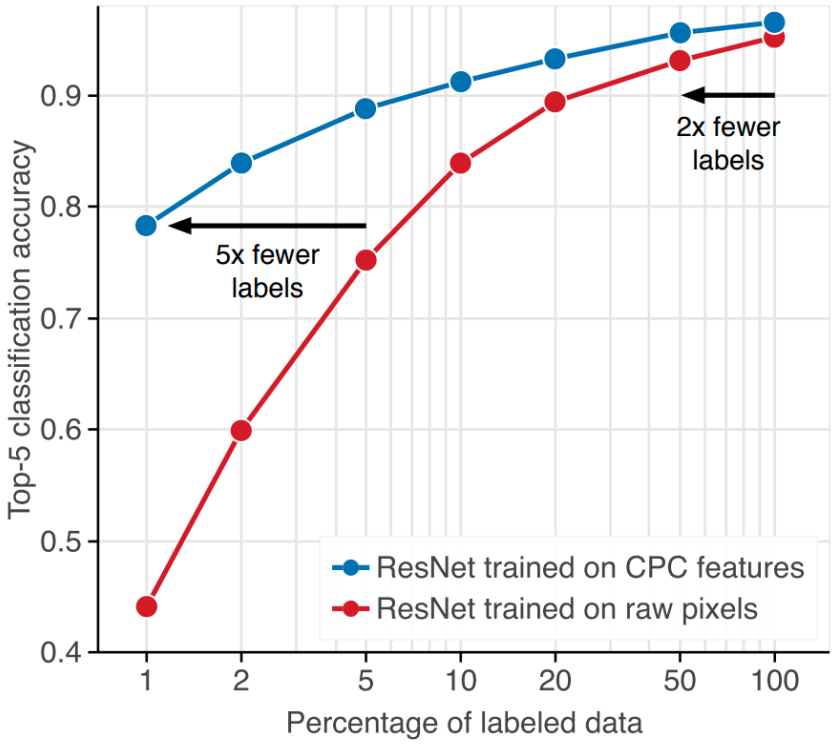

Figure 2.4: Reported [2] performance of Contrastive Predictive Coding on ImageNet compared with a ResNet baseline. Given small amounts of data, the network is able to achieve substantially increased accuracy. This graph also highlights how this can be re-framed: as achieving the same accuracy as another network, but using 5x less data.

this, given the potential reward, it appears to be a worthwhile area for investigation.

2.4.2 A Simplified method for Contrastive Learning Representations (SimCLR)

This method is described in further detail in chapter 5.

SimCLR is a similar contrastive method to Contrastive Predictive Coding, however, rather than learning an embedding network that is invarient to patch location; SimCLR attempts to train a encoder which is invarient to augmentation. In this method, a pair of images are both stochastically augmented, resulting in two augmented images. Each of these images are then embedded using the same encoder producing two embeddings. The loss function then optimised the encoder to make

the difference between the two vectors small in the case that they are augmented copies of the same image, and large if they are of different images. By optimising this objective function, an encoder is trained which should be invarient of augmentation and purely encodes the high level features.

2.4.3 Supervised Contrastive Learning

In all contrastive learning approaches, the ultimate aim is to produce embeddings in which two similar images are placed close together in the latent space, while at the same time, ensuring that two dissimilar images are far apart in the latent space. There are numerous ways in which 'similar' and 'dissimilar' can be defined, but for supervised contrastive learning, they are defined as either belonging to, or not belonging to a particular class respectively. There are a number of proposed loss functions for supervised contrastive learning: triplet [63, 64], and N-pairs loss [65].

Triplet loss: As with the semi-supervised methods for contrastive learning, the triplet loss attempts to learn an embedding in which similar points are placed closer together in the latent space and disimilar points are placed further apart. In some situations, this can be useful as a task in itself, for example, for face similarity networks such as Facenet [66] a network learns to produce embeddings in which different views of the same person produce similar embeddings. This network can then be used directly to identify users of a system. While some success for these methods has been found, there has been much less exploration for these methods for unsupervised pretraining. If one already has labels such as classes that are appropriate for training this loss function, then a more classical and highly performant loss function could be used directly.

SupCon: [67] propose a loss function which can, as opposed to the other supervised contrastive losses, lead to state of the art results that outperform a standard cross entropy loss function. The main notable change from prior work introduced by this piece, is the ability of the loss function to contrast between an arbitrary number of positives, as opposed to just one, as found normally. In most contrastive methods,

a single positive example is used which is contrasted against a large number of negatives; in SupCon, multiple positive examples are contrasted against the set of negatives. Self supervised contrastive loss functions typically take the form:

$$\sum_{i \in I} log \frac{exp(z_i z_j / T)}{\sum_{a \in A} exp(z_i z_a / T)} \tag{2.1}$$

This loss function minimises the difference between projections of 'similar' data (z_i and z_j), while maximising the difference between a projection z_i and a negative example vector z_a. The set A refers to the set of possible negative examples that a can be taken from, in most of the methods seen in this section, this would be a random image taken from elsewhere in the dataset where $i \neq a$. I refers to the set of images that the anchors (positives, i) come from. T is a temperature scaling parameter. Versions of this loss can be found in most semi-supervised contrastive approaches explored here. However, as its name suggests, this method uses labelled data to create its latent space, and therefore, based on the argument given earlier in the chapter that no more labelled data is accessible, this method can be rejected as one for study in this thesis.

2.4.4 Momentum Contrast (MoCo)

Momentum Contrast [68] (MoCo) is a somewhat similar method to SimCLR, in that it tries to embed two images that are transforms of each other close together with respect to a set of 'negative' [3] examples. The difference comes from how these negatives are given. In the case of SimCLR, the negatives are held within the mini-batch and the loss function attempts to minimise the distance between the 'positive' example and maximise the distance between all others in the mini-batch. In MoCo, a memory bank is used. With this, the set of negatives embeddings are not taken from the mini-batch themselves, rather they are in the form of a queue of previously

[3]Within the context of contrastive learning, the terms 'positive' and 'negative' examples are used: as mentioned in 2.4, contrastive learning aims to train an encoder that embeds 'similar' examples together, and 'dissimilar' examples far apart. In this thesis, 'similar' examples are termed 'positive' examples, and dissimilar as 'negative', due to the contrastive objective function framing the task as one in which the network must identify which of a set of elements correspond to a specific query example. Further detail on this can be found in appendix B.

seen examples. By separating the number of negative examples from the mini-batch size, MoCo is able to use much greater amounts of negative examples, leading to greater performance. The queue data structure allows for the algorithm to continuously update the set of negative examples using the latest X mini-batches of images.

The MoCo training protocol uses this queue of embeddings from which to act as the noise distribution sampling method. As a new batch of data is introduced, the encodings from this batch are enqueued, and the oldest is dequeued, ensuring the embeddings are relatively fresh. The use of this memory bank means that MoCo does not have to recompute the negative examples on each batch and allows for the set negative examples to be much larger than in SimCLR.

MOCOv2: The authors of MoCo released a short follow up note on the original MoCo method. They proposed using a number of the features of the SimCLR methodology to improve the performance of MoCo; namely the non-linear projection head, and the stronger augmentation. They found that introducing these methods lead to a 6.9 percentage point increase performance on ImageNet linear classification accuracy. Chapter 6 investigates the claim that stronger augmentation leads to higher performance.

2.4.5 Pretext Invariant Representation Learning (PIRL)

Pretext Invariant Representation Learning (PIRL) [16] is a closely related method to SimCLR. The PIRL method attempts to force an encoder to learn representations that are invariant to pretext augmentations, in their case the jigsaw transform (they also extend to the rotation transform). PIRL consists of an encoder that embeds two images to a latent space, one image is some arbitrary transformation of the other image. The encoder is optimised, such that the latent code of images and their transforms are close in the latent space. This is contrasted to a memory bank of embeddings taken from other images. The PIRL method does not require any one type of augmentation, they propose that it may work with any type. In [16], the authors show their method using the 'jigsaw' transformation. In this transform,

an image is split into sub images, and the order of these sub images is jumbled as can be seen in figure 2.5. The authors also generalise PIRL to use other pretext tasks, namely rotation, that is, forcing an encoder to form rotation invariant image representations, finding that this also produces useful representations.

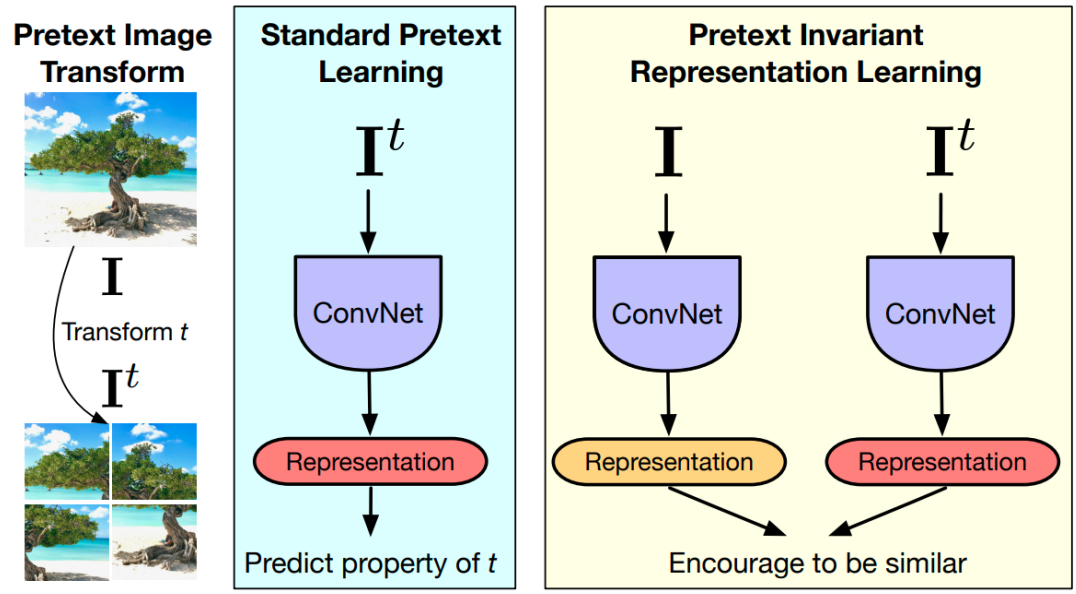

Figure 2.5: The PIRL training method, taken from [16]. In the yellow shaded box, the image shows how the same network is trained to produce representations that are similar for images that are transforms of each other.

Comparison to SimCLR: Both SimCLR and PIRL are methods in which the latent encodings produced by the training protocol are invarient to some kind of transform. PIRL uses a 'memory-bank' rather than explicit negative samples to contrast their examples to; this leads to a much larger amount of negative samples than SimCLR, at the expense of a more complex implementation.

2.4.6 AMDIM

Augmented Multiscale Deep InfoMax (AMDIM) [69] is a similar method to Contrastive Predictive Coding, in that the objective of both methods is to maximise the mutual information of a context with another 'view' of the same element. In

CPC, this context is a summary of 'past' representations of an image, with 'future' representations of the same image (see chapter 3 for more information). In AMDIM, the different views are independantly augmented derivative images of some base image. This work is based upon Deep InfoMax [70]; in which the mutual information between a global view of an image is maximised with a local view of the same image. Rather than maximising the mutual information between views of the same image, AMDIM attempts to maximise the information between two transformed images. They randomly choose from random resize, colour jitter, random flip, and random conversion to grayscale.

Comparison to SimCLR and CPC: AMDIM can be thought of as a halfway point between CPC and SimCLR, with similarities to both. CPC and AMDIM attempt to maximise mutual information between a context and a second view of the same image, however, their 'views' are different. The 'views' used by AMDIM are much similar to SimCLR, in which two augmentations of the same image are used.

2.4.7 Contrastive Multiview Coding

Contrastive Multiview Coding (CMC) [71] is another contrastive method the aim of which is to separate embeddings that represent related data, from embeddings taken from a noise distribution. CMC attempts to learn image embeddings that are invarient to different image channels such as: "luminance; chrominance; depth; and optical flow" [71]. In other words, CMC attempts to represent the information that is shared between these views of the same image, thus producing useful embeddings. As with most of the contrastive methods found in this section, this method attempts to force the network to learn high level features that will be useful for most downstream tasks, just differing in the aspect it is learning to be consistent between different 'views' of the same image.

2.5 Comparison to Other Neural Approaches

While a contrastive approach to semi-supervised learning has achieved good results, it is not without its disadvantages. Some approaches have been proposed to solve issues with contrastive approaches.

Bootstrap Your Own Latent: Boostrap Your Own Latent (BYOL) [72] is a related but different method to contrastive networks, which relies solely on positive pairs of images. This differs from the contrastive learning, in which pairs of positive examples are pushed together, and negative examples are pushed apart; in BYOL, this pushing apart does not happen. In BYOL, the authors state that two augmented views of the same image should "be predictive of each other". Given a view (transform $T1$) of an image I, the network should be able to predict how a secondary transform ($T2$) will affect the image, resulting in $T2(I)$. In BYOL, there are two networks: an online network and a target network. The online network is trained to predict the output of the target network, thus, training does not require negative examples, decreasing the complexity of training. During training, an image is augmented in two separate ways, one transformed image is then embedded using the online network, and one by a frozen target network. Both embeddings are then fed through a projection head to improve performance as with prior work. The embedding is then fed to a predictor network whose goal is to predict the output of the target network. Surprisingly, the embeddings produced by BYOL do not converge to a collapsed solution. Due to the lack of negative examples, the network could learn to output a constant embedding, thus minimising the loss; however, this does not happen. The authors suggest that such equilibria could be unstable.

2.6 Summary of Contrastive Methods

Table 2.1 gives a high level comparison between the key methods outlined in this chapter, highlighting the pretext task along with an indicative performance metric.

Table 2.1: A summary of the various methods presented in this chapter, giving their pretext task, loss function and an indicative performance. It is important to note that the papers varied in their evaluation methodology and so the results are not directly comparable between papers. For example, some of the papers reported Top-1 and some Top-5 accuracy on ImageNet; some used subsets of the full ImageNet, and some the full dataset; in addition, many types of networks were used, from AlexNet to ResNets. Therefore, for each of the methods, a supervised baseline is given, which should more accurately reflect the supervised performance one would expect for their experimental set up.

Method	**Date**	**Pretext task**	**Indicative Performance**	**Comment**
Rotation Prediction (Gidaris et al)	Mar 2018	Predicting the level of rotation of a query image from a set of predefined rotation levels.	CIFAR-10 accuracy – Supervised: 92.80% – Rotation prediction: 91.16%	
Jigsaw (Noroozi & Favaro)	Aug 2017	Solving jigsaw puzzles. Given a randomised set of patches, which is the correct ordering of those patches?	PASCAL VOC detection: – AlexNet style (supervised) = 78.2% – Jigsaw: 67.6%	

Method	Date	Pretext task	Indicative Performance	Comment
Context prediction (Doersch et al)	Jan 2016	Predicting the relative position of a query patch relative to a context patch.	VOC-2007 mAP: – VGG: 89.3% – Context prediction: 61.7%	No supervised result is given, the supervised performance is taken from the VGG paper. [73]
Colourisation (Larsson et al)	Aug 2017	Turning a greyscale image to a full colour image.	VOC 2007 mAP – VGG-16: 86.9% – Colourisation VGG-16: 77.2%	
Inpainting (Pathak et al)	Nov 2016	Predicting the content of a randomly selected patch of an image, based on the rest of the image.	Pascal VOC detection: – AlexNet style (supervised): 78.2% – Inpainting: 56.5%	

Method	Date	Pretext task	Indicative Performance	Comment
Contrastive Predictive Coding V1 (Oord et al)	Jan 2019	Differentiating patches surrounding a query patch from a set of patches that are randomly sampled from a large dataset.	ImageNet Top-1 accuracy. – AlexNet: 62.5% – CPC: 48.7%b	No supervised performance was given in CPCv1 paper. The supervised indicative result, the top-1 accuracy from the original AlexNet paper was used.
Contrastive Predictive Coding V2 (Hénaff et al)	Jul 2020	Differentiating patches surrounding a query patch from a set of patches that are randomly sampled from a large dataset.	ImageNet Top-5 accuracy, trained on 100% of the dataset. – "supervised baseline": 95.2% – CPCv2 96.5%	Much greater relative levels of performance were found when evaluating on smaller subsets of the labelled datasets.
SimCLR (Chen et al)	Jul 2020	Maximising the agreement between two independently	ImageNet Top-5 accuracy, trained on 10%	This performance was using models trained on limited subsets of the full

Method	Date	Pretext task	Indicative Performance	Comment
SimCLR (Cont.)		augmented copies of the same image and minimising the agreement between two random images.	of the full dataset. –Supervised baseline: 80.4% – SimCLR: 87.8%	ImageNet dataset which likely favoured the performance characteristics of the unsupervised method.
MoCo v1 (He et al)	Nov 2019	Maximising the agreement between two independently augmented copies of the same image and minimising the agreement between an image and the embeddings for other images taken from a memory bank.	ImageNet Top-1 accuracy: – Supervised 76.5% – MoCo: 77.3%	MoCo was trained on the Instagram 1 billion dataset for this task.

Method	**Date**	**Pretext task**	**Indicative Performance**	**Comment**
MoCo v2 (Chen et al)	Mar 2020	Maximising the agreement between two independently augmented copies of the same image and minimising the agreement between an image and the embeddings for other images taken from a memory bank.	ImageNet accuracy. Supervised baseline: 76.5% – MoCo v2: 71.1%	
Pretext Invariant Representation Learning (Misra & van der Maaten)	Dec 2019	Jigsaw task, see earlier.	ImageNet Top-1 accuracy: – PIRL: 57.8%	The paper uses the Jigsaw task, however, any image augmentation could be used. – No supervised performance was given.

Method	**Date**	**Pretext task**	**Indicative Performance**	**Comment**
Augmented Multiscale Deep InfoMax (Bachman et al)	Jul 2019	Maximise information between different views of the same object.	ImageNet accuracy: – ResNet50v2: 74.4% – AMDIM: 68.1%	
Contrastive Multiview Coding (Tian et al)	Dec 2020	Maximising the similarity between two views of the same scene. For example, a depth map and a standard picture.	Accuracy on STL-10: – “supervised”: 65.1% – CMC: 58.3-60.1% depending on setting	

Method	Date	Pretext task	Indicative Performance	Comment
Bootstrap your own latent (Grill et al)	Sep 2020	Predicting the embedding given by an out of date version of the current encoder.	Top-1 accuracy on imageNet using 10% of the available labels: – ResNet50: 56.4% – BYOL: 68.8%	A graph from the paper shows that when evaluating on the full ImageNet dataset, supervised performance is greater than that of BYOL. However, the values from that graph are not available.

2.7 Foundation Models

Since the main body of this work was completed, a new class of self supervised models has become more widespread: foundation models. Initially used mainly within the natural language processing community to describe a certain class of large language models [74], the term foundation model is now broadly used across the machine learning to refer to very large models that have been trained on internet scale data [4]. Foundational models are similar to the semi supervised methods presented earlier in the chapter: they gain their power by using large unlabelled dataset to learn features that are useful for a downstream task. The main difference between methods we would refer to as semi-supervised and foundational models are the scale. Foundational models are typically extremely large (many billions of parameters) and are trained on very large datasets. By training on these huge datasets, these foundational models are able to learn more generalised features that should be useful in a much larger number of areas.

Large Language Models: LLMs have become so synonymous with foundation modes that they are often used interchangeably. Large Language Models are a type of neural network that aim to create a model with understanding of language through pretraining on huge amounts of text data. Due to the commercial sensitivity of these models, training information about the state of the art models is less readily available than is ideal. Despite this, some open source models are available such as the llama series of models [78] [76] as well as the Mistral series of models [79]. These models are very similar to the semi supervised models presented in this chapter: They are trained on an unsupervised task (for large language models, this is usually an autoregressive task, i.e. predicting the next word in a block of text, however, this can also be trained to predict a masked word, such as BERT [80]). From there, either the embeddings can be used directly (as in some methods in this chapter), or

[4]There is no good definition for the amount of data that would be considered "very large", but to give the reader some examples: GPT-3 was trained on approx. 400 billion tokens [75], llama-3 was trained on 15 trillion tokens [76], and the LAION dataset contains 5 billion uncurated images [77].

they can be finetuned for other tasks (eg instruction tuned LLMs). It is important to note the scale of these models in comparison to the other models used in this chapter. For example, the training of the SimCLR model took approximately 1.5h using 128 TPU cores (approx. $300 for 1 training run to 100 epochs) whereas the Llama 3.1 model used 16000 GPUs concurrently (no cost is given for the training, however, the capital expenditure of the training cluster was likely to be in the 100s of millions of dollars).

While more common in natural language processing, foundational models can also be found in other domains. Most notably for this thesis are the computer vision foundation models:

Segment Anything: segmentation models are typically trained to segment using application specific datasets [81] [82]. Segment Anything [83] takes a different approach: by training a segmentation network on a suffficiently large and diverse dataset, a network is able to be trained that achieves good performance on a wide selection of tasks. Segment Anything is trained on 11 million images with an emphasis on collecting a wide range of images. The authors found that the zero-shot performance of the network was comparable to networks trained for that task.

DINO v2: Meta AI research released a model they term DINO v2 [84]. This model is a vision transformer trained on an extremely large (billion image scale), uncurated image dataset. It is also important to note the sheer scale of the training these models. The authors not that the full DINO v2 project consumed on the order of 4.8 million GPU hours. There are very few organisations that are able to produce work of this scale, and this is certainly far above the level available for use within a PhD.

2.8 Discussion of the Literature

In chapter 1, I introduced the problem specification that forms the basis for the work presented in this PhD thesis. While deep learning has revolutionised how we

interact with the world in many fields, the application of deep learning methods to the medical domain has been underwhelming. In chapter 1, I posit that this lack of application is due to the increased cost of creating large, labelled datasets in the medical domain in comparison with domains that do not need highly specialised labellers. I also highlight that there is a large cost differential between the collection of unlabelled and labelled datasets which can often be many orders of magnitude. If it were possible to use this unlabelled data to effectively increase the performance of deep learning methods with limited labelled data, the cost of applying these methods could be massively reduced.

Chapter 2 has identified a number of methods that could be used to reduce the cost of applying the current state of the art deep learning methods to medical imaging tasks.

Section 2.1 presents two methods in which the unlabelled data is 'labelled', with the naïve approach being to label the data manually. This approach is described as naïve, however, it could be the best approach for certain applications: it produces gold standard, human verified labels and in certain circumstances can produce a model that is faster to market (labels that can be given by non-specialised workers can be created from a large number of commercial suppliers [85], [86] in a few days), and with better performance than collecting an unlabelled dataset and trained using a semi-supervised approach. Its use will depend on the exact problem circumstances. The second approach to create a 'labelled' dataset is an approach known as pseudo-labelling [53], while much cheaper than manually labelling, the performance of this method will depend entirely on the performance of the initial network trained (that is, the network that is used to label the rest of the unlabelled data). For the rest of this thesis, I am working under the assumption that the cost differential of labelled and unlabelled data is such that no more labelled data can be generated. This may or may not be true of a reader's problem space, and therefore care should be taken before applying the work presented in this thesis.

Section 2.2 presents some alternative methods that could be used to solve a busi-

ness problem. Many problems within the medical domain can be represented as out of distribution detection: in many cases, it would be entirely appropriate for a Machine Learning based system to highlight cases in which there is an abnormality present which can be flagged for human review. These methods would not require any labelled data, and so would not have the issues associated with manual labelling (cost) or pseudo-labelling (lower performance). However, these methods would not be able to perform classification: which may, or may not, be important for the problem space someone is trying to solve.

Due to these limitations, section 2.3 introduces a set of pre-training methodologies that I term 'Auxiliary task methods'. This set of pre-training methods incorporates methods in which a pretext task is used to learn a set of weights for a model that is more favourable for performance of the downstream task than randomly initialising the weights. These weights can then either be frozen and used as a method to project data down to a latent space or finetuned on the downstream task. Unfortunately, these methods have not found a level of performance greater than that of supervised methods. This could be for a variety of factors, but I argue that this is most likely due to the knowledge required for succeeding at the pretext task not being the same knowledge that is required to succeed at the downstream task. For example, the network used for context prediction may place heavy emphasis on encoding the low level features at the edges of the image, as they are most relevant for predicting which patch is next to the context patch, however, this information is not that relevant for a downstream task such as predicting if an image is a cat or a dog. Similarly, for a method such as an autoencoder, slight variations in the hue of an image will be penalised at the same level as completely missing a section of the image, which could be extremely relevant for the downstream task. This lack of transfer between the upstream and downstream tasks likely led to lower performance than supervised training.

Some of the contrastive approaches introduced in section 2.4 attempt to solve the issue of the lack of co-linearity between the performances on the upstream and downstream tasks by attempting to force the networks to learn high level features that

are relevant for the downstream tasks. For example, Contrastive Predictive Coding (CPC) v2 introduced patch-based augmentations (that is, random augmentations of individual patches independent of the other patches). The authors claim that this change forces the network to learn the high level features due to it not being able to solve the problem using trivial solutions, such as continuity of lines or the colour distributions of the patches. SimCLR also follows this approach by learning representations that are consistent between images that have independently augmented copies of the same image, thus learning features that are unchanging in between these sets of augmentations. This does, however, assume that the high level features that this learns are the same (or at least partially overlapping) with the set of features that are useful for the downstream task. This would be dependent on the specific task. This learning of high level features varies in its definition across the various methodologies, but generally has lead to greater levels of performance than the auxiliary task methods, seen in the previous section.

2.9 Conclusion

Chapter 1 introduced the problem that training state of the art, deep learning models requires huge amounts of labelled data. This is extremely challenging in many domains, including the focus of this thesis: medical imaging. As discussed previously, labelling of data within the medical domain requires the knowledge of highly specialised, and therefore expensive, domain experts. The creation of a labelled medical image dataset could be orders of magnitude more expensive than a general imaging task. For deep learning to be accepted into industry, the cost of application must be substantially reduced.

In this chapter, a number of possible solutions were presented mainly focusing on semi-supervised methodologies. However, these methods often provide lower levels of performance than would be required for these methods to be applied into a production environment, and a traditional supervised approach may be more suitable.

In the second half of the chapter, contrastive learning was introduced as a set of

possible methods that have increased performance on some general imaging datasets and are strong candidates for solving our problem case. In the next two chapters, one of the methods highlighted in this chapter (Contrastive Predictive Coding) is evaluated in its performance on a medical imaging diagnosis dataset. These chapters lead on to further study in which a second method highlighted here, SimCLR, is evaluated for its potential on medical imaging tasks. Based on this survey of the literature, there are still unanswered questions as to whether these contrastive methods would be suitable for use with medical imaging (aim 3) and if so, how should they best be applied to achieve the best results (aims 1 and 2).

Contrastive learning has been chosen as an avenue of research due to the encouraging results found in CPCv2 and SimCLR. Semi-supervised learning has long been touted as a solution to the large cost associated with generating labels for a novel dataset, however, a large amount of the methods studied here failed to achieve higher levels of performance than a supervised baseline. CPCv2 claims that their contrastive pretraining methodology increases the performance over supervised training in situations with limited labelled training data; fitting with the problem specification felt acutely in the medical domain.

Chapter 3

Contrastive Predictive Coding Background

Chapters 1 and 2 outline the problem setting for how semi-supervised learning could provide benefits when applied to a medical imaging task; outlining the difficulty in acquiring labelled data, and how the contrastive learning framework can leverage the power of large, unlabelled datasets. In this chapter, one such method of contrastive learning, Contrastive Predictive Coding (CPC), is presented. Contrastive Predictive Coding is a representation learning protocol which learns latent embeddings of data that encodes high level information while discarding low level information. Prior work has shown that Contrastive Predictive Coding is able to improve predictive performance on ImageNet when given limited labelled data, along with large volumes of unlabelled data. However, work investigating whether this method could be used on medical imaging datasets is lacking. This chapter describes the design of the protocol, and its iterative improvement - CPCv2 - before exploring its usage in the literature and comparing and contrasting it to the Noise Contrastive Estimator.

3.1 Network Description

Contrastive Predictive Coding is a method for unsupervised learning, which can embed useful information into a low dimensional latent encoding. The CPC objective encodes contextual information into this latent encoding which allows the learned

representation to more efficiently represent the data. It can be used on any data which can be represented as a sequence and consists of three principles:

- **Contrastive:** the problem is framed as a contrastive learning problem , that is: the network must 'decide' which embedding matches a query embedding, out of a set of embeddings taken from a noise distribution. [6] proposes the InfoNCE loss function based upon the Noise Contrastive Estimator.
- **Predictive:** the embeddings are not contrasted directly, instead a 'context' vector is used to predict 'future' embeddings. These 'future' embeddings are subsequently contrasted with negative noise examples.
- **Coding:** The input data is represented as a latent code. Each datum is projected down to a low-dimensional, latent code using an encoder. As with most work within this domain, this generally takes the form of ResNet, however, any base encoder could be used.

The ultimate goal of the CPC protocol is to train an encoder to produce embeddings that encode high level features. The assumption behind the protocol is that the optimal features for the downstream task are these high level features, more discussion and investigation on the impact of this assumption can be found in chapter 6. The method relies on two sub-networks: an encoder, and an autoregressive predictor sub-network. The predictor sub-network attempts to predict the latent representation of parts of the data surrounding the current data. For example, in the case of images, the predictor attempts to predict the latent representation of the patches below the current patch; and in the case of speech, attempts to predict the latent representation of the word before and the word after.

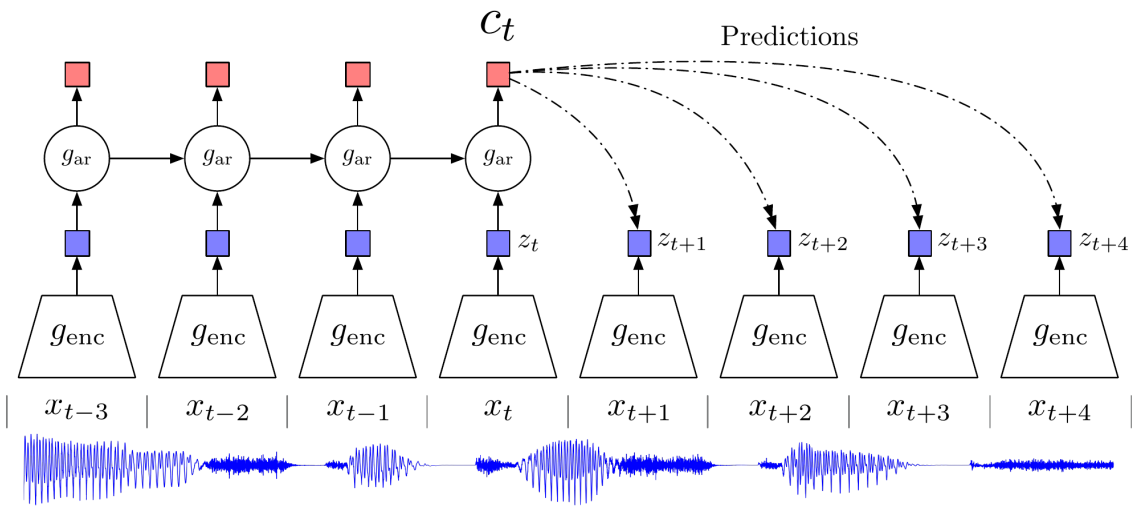

Figure 3.1: A diagram of the Contrastive Predictive Coding method. CPC is able to be used on a number of modalities (here, signals are shown), but the same structure remains. Sequential patches $\{x_t - 3,\ x_t - 2,\ x_t - 1,\ x_t\}$ are each encoded by g_{enc} down to a latent representation. Each of these latent encodings are fed to an autoregressive model (denoted g_{ar}) which summarises the data into a context vector (C_t). Non-linear projections are then taken from this context vector to give the 'future' predictions. These predictions are then individually contrasted with a set of 'noise' vectors.

.

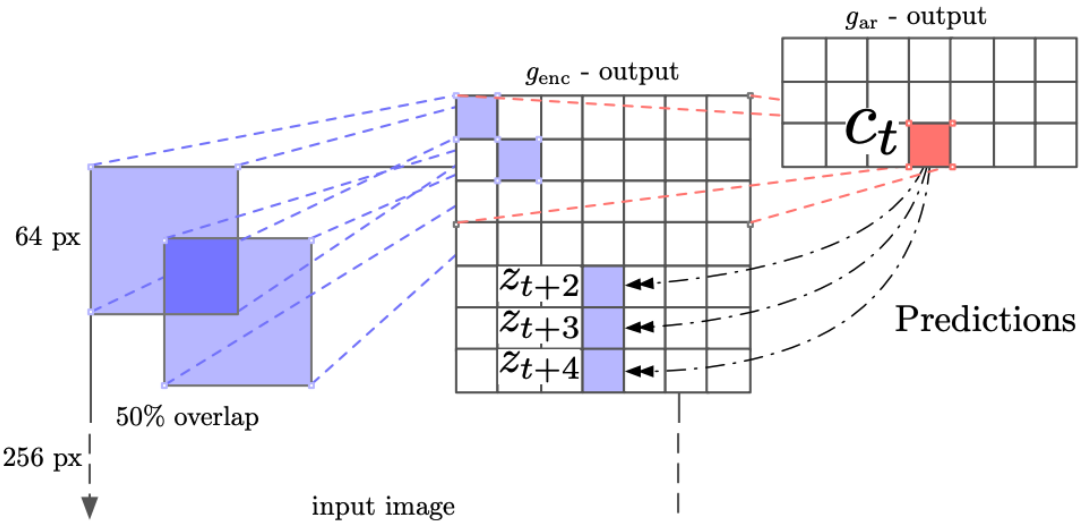

Figure 3.2: A diagram of how image patches are encoded taken from [6]. Partially overlapping image patches (left) are encoded down to a vector. After encoding, the output of the model (middle) is a 7x7x1024 tensor. A sequence of these are then summarised into a context vector C_t (right). From this context vector, predictions are made (z_{t+2}, z_{t+3}, z_{t+4}; note these are would be x_{t+1} from the figure 3.1). The contrastive loss is then applied to these predictions vs the true future embeddings.

3.2 Loss Function

The loss function attempts to learn useful features through two related goals: 1) make similar images close in the latent space, and 2) make dissimilar images far apart in the latent space. The inclusion of the second goal ensures that the encoder does not just learn a constant embedding, as would be the case if only the first goal existed [1]. The CPC protocol defines these 'similar images' to be image patches that are physically close together in the image. The loss must then 'push' patches that are dissimilar to be far apart in the latent space. The CPC objective defines dissimilar patches to be images taken from a noise distribution, defined as other images within the full dataset. Through this process, it is hoped that the encoder can learn embeddings that are useful no matter what task the encoder is used for, from classification to segmentation. This is despite no explicit training for these tasks.

Contrastive Predictive Coding optimises a contrastive loss function, InfoNCE: the network is optimised to identify a target $z_{i+k,j}$ from a set of randomly sampled feature vectors z_l. The probability given to each possible vector is calculated using a softmax, and evaluated using cross-entropy loss. Summing over all patches achieves:

$$L_{CPC} = -\sum_{i,j,k} log \frac{exp(\hat{z}^T_{i+k,j} z_{i+k,j})}{exp(\hat{z}^T_{i+k,j} z_{i+k,j}) + \sum_l exp(\hat{z}^T_{i+k,j} z_l)} \tag{3.1}$$

In this equation z_{i+kj} represents the encoding of the kth offset of patch i,j using the current encoder, with $\hat{z}_{i+k,j}$ representing the projection from the context vector. The rest of the equation is simply the categorical cross entropy of the softmax of this, with z_l being an encoding of a patch from elsewhere in the space of possible images. Taken as a whole, this is the probability that the network assigns to the predicted positive being the true positive. This loss function will optimise the network to produce embeddings that put similar patches together in the latent space

[1] BYOL seen in chapter 2 does not use negative samples, and yet does not produce constant embeddings: only hypotheses are given for why this could be the case and more investigation may be required.

and dissimilar images far apart in the latent space.

This loss function, introduced in [6], has been used both in its current form, and with modifications, in numerous subsequent works. [1, 68, 71] all use a modification of the InfoNCE loss function which incorporates l_2 normalisation of vectors and temperature scaling to improve performance. This modification is discussed in further detail in chapter 5.

3.3 Improving Performance: CPCv2

Contrastive Predictive Coding version 2 [2] (figure 3.3) was introduced as an improvement to the original CPC method. Henaff et al introduced a number of changes to the method, which in total improved the top-1 performance on 1% of the data from 23.1% to 52.7%. This is a marked improvement, however, it strays away from the standard method for comparing the performance of unsupervised methods: adding a linear layer to the output of the embedding. Therefore, the performance of CPCv2 cannot be directly compared to the other methods introduced in chapter 2, however, it is still important as it is one of the first methods to claim to outperform supervised learning.

The changes between CPCv1 and CPCv2 are summarised below:

- **Changes to the Patch Size:** In CPCv1 Oord et al used a 7x7 grid of patches; in CPCv2, the size of the individual patches was increased, thus the network was able to 'see' more of the image at once, meaning larger features could be encoded. When evaluated on an arbitrary dataset, this improved performance by 2%.
- **Model Capacity:** The width of the network at the site of latent embedding was increased from 1024 dimensions in CPCv1 to 4096 dimensions in CPCv2. When any type of unsupervised or semi-supervised learning task is performed,

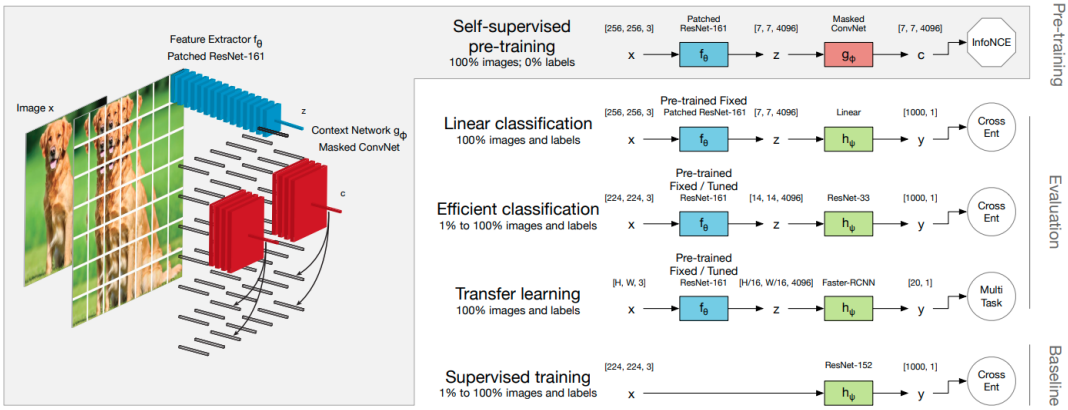

Figure 3.3: Visual representation of the CPC encoder training, along with how this encoding is used in the supervised phase of the method. This image shows not only the CPC encoder training, but also how the encoder can be utilised for efficient classification. This image has been taken from [2]

.

information is lost. With a good pretext task, the information lost is not that useful for the secondary task. By increasing the size of the latent embedding, more features are learned, therefore, less information is lost. In addition to network width, network depth is also increased. On an arbitrary task, these improved performance by 5%.

- **Patch-based Augmentation:** CPCv2 introduces patch-based augmentation. Rather than the whole image being transformed once, each patch that is created is independently augmented during sampling. This method was previously seen in context prediction [15] (section 2.3) and is applied for the same reason: without patch-based augmentation, it is possible for the network to learn to recognise low level feature across patches, rather than learning the high level contextual features. If the network is able to use these cues to solve the pretext task, the network could achieve higher levels of pretraining performance, which actually degrades downstream performance. Patch-based augmentation should help mitigate some of this risk by making it harder for the network to use these shortcuts. When evaluated on an arbitrary dataset, this improved performance by 4.5%.

- **Increasing the Degrees of Prediction:** In CPCv1, the goal of the CPC objective was to correctly identify which set of patches came from the patches proceeding the query image. CPCv2 extended this to force the objective to be able to predict the embeddings of the patches surrounding the query sequence, no matter which direction the patches are given. This includes top-to-bottom, bottom-to-top, left-to-right, and right-to-left. When evaluated on an arbitrary dataset, this improved performance by 2.5%

- **Change from Linear Layer to ResNet:** As explained before, the standard method for evaluation of unsupervised methods is to add a linear layer to the output of the network, this allows for evaluation of how linearly separable the classes have become, a pseudo evaluation of how well the unsupervised learning method has worked. However, this is unlikely to lead to the best performance. Hénaff et al changed the secondary network to a ResNet, which massively increased performance at the expense of comparability.

- **Layer Normalisation:** Hénaff et al found that the use of batch normalisation within the CPC protocol actually harmed performance. However, layer normalisation can be used in its place to increase performance. Hénaff et al hypothesised that this was due to the network taking advantage of the batch statistics to make its prediction, in essence: cheating. By changing to layer normalisation, and evaluating on an arbitrary dataset, this improved performance by 2%

As with CPCv1, while there is a large amount of interest in the method, there has been limited attempts to apply the method to datasets outside of general imaging datasets such as ImageNet [41]. When applied to other datasets, CPC would often report good, but not necessarily state of the art results: [87] [88] [89].

3.4 Comparison to Related Methods

Contrastive Predictive Coding builds on a number of works. The contrastive loss function, InfoNCE, was based upon the work of Gutmann in Noise Contrastive Estimation [90].

Noise Contrastive Estimation: Noise Contrastive Estimation (NCE) is a estimation principle to model the underlying distribution of a set of data. This modelling is accomplished through distinguishing between the real data within the set and artificial noise. This loss function is conceptualised as a supervised learning problem: training a network to complete a binary classification problem, classifying real data from noise. Given a set of real data: $\{X_1, X_2, X_3, X_4, ...X_N\}$ and a set of noise $\{Y_1, Y_2, Y_3, Y_4, ...Y_N\}$, NCE optimises the objective function:

$$J_T(\theta) = \frac{1}{2T} \sum_t ln[h(\mathbf{x}_t; \theta)] - ln[1 - h(\mathbf{y}_t; \theta)] \tag{3.2}$$

Where:

$$h(\mathbf{u}; \theta) = \frac{1}{1 + exp(-G(u; \theta))} \tag{3.3}$$

$$G(\mathbf{u}; \theta) = lnp_m(\mathbf{u}; \theta) - lnp_n(\mathbf{u}) \tag{3.4}$$

The objective is to maximise the difference between a positive item and noise, and minimise the distance between noise and noise. In equation 3.2, $h(\mathbf{u}; \theta)$ is the sigmoid function of the difference between the probability that the datum comes from and the noise distribution (eq 3.3). In the equations above, theta (θ) is the model parameters; with $h(\mathbf{x}_t; \theta)$ being the model with parameters θ and input x_t.

Gutmann et al note that the closer the noise distribution is to the data distribution, the better the model will be. It is important to point out that the representations produced by the NCE are not normalised: the model must learn to produce normalised vectors by itself. Optimisation of the objective will result in a statistical model of the data which can then be used in a further task.

In contrast to a lot of previous unsupervised methods such as PCA [91], the Noise Contrastive Estimator method conceptualises the unsupervised task as a supervised learning problem. From this building block, the InfoNCE loss function described in section 3.2 was built. This extended the work from a binary classification problem to a contrastive problem, in which the network has to identify the correct embedding from a set of possible 'noise'. The "noise model" used in CPC is images taken from elsewhere in the full dataset, as opposed to any artificially constructed dataset. This allows the noise model to be close to the true distribution, which should improve performance.

3.5 Use in Literature

Semi-supervised learning is often proposed as a possible solution to a lack of labelled data in many fields. Contrastive Predictive Coding should help with data acquisition problems that are acutely felt in the medical imaging domain. Despite this, there is limited work looking to utilise CPC to improve results. This section attempts to describe and evaluate the work that has been conducted.

Histology image interpretation has had by far the most interest in using CPC [2]. [92] applied a two stage process to detection and localisation of breast cancer in histological data. They combined two different methods, CPC with multi-instance learning. As with normal CPC, an encoder is trained on an unlabelled dataset of histopathelogical images, learning an embedding of the data. This embedding is then used in the secondary task: multi instance learning. Multi instance learning (MIL) is a type of weakly-supervised learning in which a classifier learns to classify a 'bag' of data points as either 0 or 1, with a bag labelled as 1 if it contains at least one instance of the feature to be detected, and 0 if it contains no instances. This allows both classification as well as localisation with no segmentation data. Segmentation of medical images is a time consuming and expensive process, particularly for medical images. While the method works for weak segmentation with no direct labels, it usually suffers from very weak performance in comparison to a normal

[2] There does not appear to be any principled reason for this.

supervised method. [92] reported a 62.6% ± 11.6% accuracy on a binary classification task when just using MIL. However, when the authors trained the MIL on a CPC trained embedding mechanism, performance was increased to 90.6% ± 2.88%.

In addition to image data, 3D adaptations of the CPC method have gained interest due to 3D data being more prevalent in medical imaging than in other domains. Modalities such as MRI, CT, and OCT all provide 3D scans, and methods that can natively handle this data could possibly lead to better performance. [93] presented a 3D adaptation of CPC along with other adapted self-supervised methods, and evaluated on three distinct 3D datasets. They found that training on 3D data directly performed better than using a model trained on 2D data and used on 3D. [94] evaluated 3D CPC along with other methods, and a performance increasing "task related CPC" with 3D CPC performing on par with 3D jigsaw. [95] also found that 3D CPC outperformed most other comparable baselines.

As outlined previously, CPC is able to be applied to many different data types, including signal data. One type of signal data that has received a large amount of attention is electrocardiogram (ECG) classification. [96] applied various methods found in computer vision to the interpretation of ECG data, they found that CPC performed well; using CPC encodings and a linear layer was only 0.8% below supervised performance despite a significantly less complex learning model. [97] also used an image based method for their analysis of CPC on ECG data. CPC was outperformed by more popular methods such as SimCLR and MoCo but performed well, as with the previous method. [98] explored ECG abnormality classification among other tasks, and found that CPC performed the worst out of tested methods, and significantly worse (71.6% to 98.4% accuracy) than the supervised baseline. [99] explored classifying electroencephalogram (EEG) data with various contrastive self-supervised learning methods including CPC, finding that CPC performed worse than their supervised baseline and 5 out of 6 of the other methods tested.

The large majority of the work completed on CPC has been focused on these small areas of interest. For CPC to gain more utility, its evaluation needs to be taken on

a greater amount of datasets across multiple imaging domains.

3.6 Direction of Future Work

Contrastive representation learning provides a possible framework that could alleviate some of the issues around data access for medical image analysis using AI. Contrastive Predictive Coding is one such method that has found success on general (ImageNet) datasets. Despite this theoretical utility there is a lack of replication on datasets outside of general imaging datasets, and those replications that do exist mainly focus on a limited number of domains (histology, ECG, and 3D adaptation have a lot of attention). In addition to the lack of replication, the replications that do exist does not show the same level of improvement that the original paper [2] showed.

Overall, Contrastive Predictive Coding [6] introduces an interesting semi-supervised methodology with theoretical use cases within the medical imaging field. Oord, Hénaff, and others have provided evidence for the utility within general imaging tasks, however, the evaluative work on medical imaging task is limited. To address this limitation, I present the following chapter: an evaluation of Contrastive Predictive Coding on a novel polyp detection dataset. In addition, work is conducted to examine how well the representations handle adversarial attack and domain shift. In the next chapter, work is conducted to examine the effect of minor perturbations of the input data on the ability of the network to produce good results. I use the term 'adversarial attack' in this section, however, this will measure the ability of the networks to withstand any such minor variation in input data, such as sensor noise and artifacting. By increasing the body of work evaluating CPC, it is hoped that this will add to the evidence for when CPC could be of use.

Chapter 4

An Evaluation Of Contrastive Predictive Coding For Medical Image Analysis

Abstract

Numerous semi-supervised methods, such as Contrastive Predictive Coding (CPC), have been suggested for improving performance when faced with limited labelled training data. This chapter tests whether the performance of a ResNet trained on CPC embeddings is better for polyp detection in colonoscopy images than a ResNet trained on the pure pixels. It shows that a ResNet trained on CPC embeddings has higher classification performance (+11.2%) when given limited data. The resistance of the two approaches to perturbation of the images is evaluated, showing that ResNets trained on CPC embeddings could be more susceptible to random perturbation than ResNets trained on the pixels and presents a technically novel mitigation to this issue. In addition, an evaluation of how well these models handle domain shift shows a statistically significant improvement in AUC when learning from CPC embeddings. Due to the success of applying the CPC methodology to a polyp detection task, the work is extended to a larger sample of medical imaging tasks, finding that CPC improves performance on some, but not all datasets.

4.1 Introduction

Contrastive Predictive Coding (CPC) [6] [2] is a representation learning method to establish powerful image features without the need for labelled data. The original CPC paper (CPCv1) proposed a universal self-supervised representation learning approach that can embed shared contextual information across space between high-dimensional signals while discarding less relevant local low-level information and noise. Intuitively, CPC focuses on learning the context of the features within the image, by learning representations that relate high-dimensional signals from one area in the image with representations in other areas of the image. This representation can then be used by a second network to perform a classification task. In this research, embeddings are obtained from training a CPC network on medical images to enhance the performance of a classifier trained on small datasets. In this work, this second phase of learning, using the small dataset to learn from the representation learned through CPC, is termed 'the downstream task'. Existing work shows that the advantage obtained from using CPC varies with the size of the dataset available for the downstream task, and that CPC will be most beneficial where that dataset is small. However, some published work [6] is based on comparisons where an under-powered baseline is used. Therefore, this work explores whether the advantage remains when an equally powerful network is used for the comparison.

A further question for such representation learning methods is whether the use of the method makes the network more or less robust to challenges known to affect reliability and generalisability: this work investigates both domain shift and adversarial attack. It is possible that representation learning techniques could make the downstream learning less susceptible to such shifts, if the embeddings learned in the initial task are more generalisable. Equally it is possible that the two-stage pipeline builds in more dependencies, making the approach more susceptible to such shifts, showing a large difference in the robustness of the two approaches.

In this chapter, the Contrastive Predictive Coding (CPC) framework is evaluated for identifying polyps in colonoscopy images. Cancer of the colon is one of the leading

cancers in both men and women [100]. Some studies [101] [102] have indicated that having a second observer at a diagnostic procedure can increase the rate of detection of these cancers by up to 50%. It is possible that AI can act as that observer, increasing detection rates [30] [31]. If this could be applied to clinical practice, mortality from missed lesions could be reduced. In addition to colonoscopy images, this chapter studies the effect of learning from CPC embeddings across three additional datasets: an extended version of the polyp detection dataset; an Optical Coherence Tomography scan dataset; and a dataset consisting of dermatology images.

It is hoped that this chapter can facilitate improvements in the state of the art when performing machine learning on limited data.

4.2 Methodology

In 2018, [6] introduced the representation learning framework called Contrastive Predictive Coding (CPC). The framework proposed a universal self-supervised representation learning approach that can embed shared contextual information across space between high-dimensional signals while discarding less relevant local low-level information and noise. This representation can then be used by a second network to perform a classification task. Figure 4.1 shows a visual representation of the CPC framework, and how the CPC trained encoder can be used to train a second ResNet. Part (a) of the diagram shows how a ResNet-50 is trained using the autoregressive (AR) network along with the InfoNCE loss function. Parts (b)(i) and (b)(ii) show how the CPC encoder can be used for the training of a ResNet. Figure 4.1 additionally shows how the labelled data is fed directly to the ResNet for training the baseline models

Encoder Training: Every image is split into a 7x7 grid of overlapping patches, each of which are randomly augmented using channel dropout, rotation, shear, elastic transform, colour, and jitter. These patches are then projected to an embedding using a ResNet-50 [7] with the final layers removed after the flatten layer, resulting in a 1024 vector as the embedding for each patch. A sequence of these patches

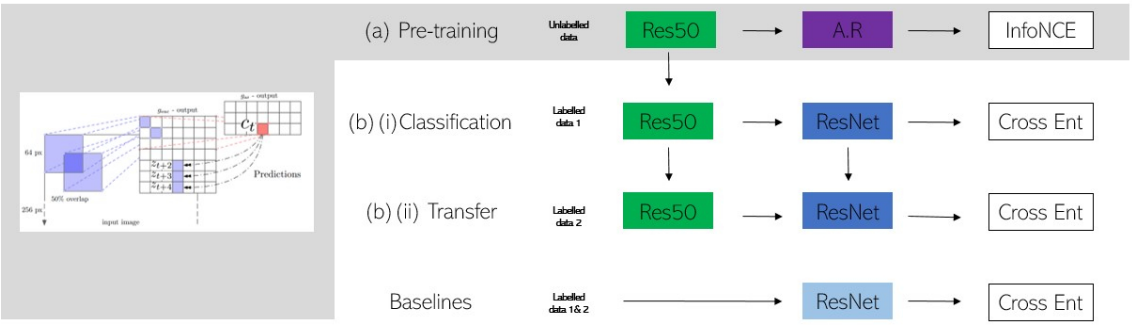

Figure 4.1: Visual representation of the CPC encoder training, along with how this embedding is used in the supervised phase of the method. This image shows not only the CPC encoder training, but also how the encoder can be utilised for classification. Adapted from [6].

.

(length $\in$ {1,2,3}) is fed to an autoregressive model (GRU [103]) used to predict the three 'future' embeddings of the sequence. The loss function then contrasts these predictions with embeddings taken randomly from the full dataset using the InfoNCE loss function, optimising the encoder to learn low-frequency features. The CPC encoder is trained for 60k iterations [1], with a batch size of 16, and optimised using the ADAM optimiser [104] and a learning rate of 2e-4. This training is one epoch of 60k images. While this was smaller than was found in [6], this was multiple days of training time. Throughout this thesis, trade offs had to be made due to the computational cost of training these unsupervised methods. This is discussed in the limitations section of chapter 8. To train the encoder, the unlabelled images from the HyperKvasir dataset [10] are used.

Classifier Training: After training the encoder, all sections are discarded except the ResNet encoder. For each image in the labelled training set, the image is split into 7x7 grid of patches and embedded using the pre-trained encoder, resulting in a 7x7x1024 tensor for each image. A ResNet-11 (with reduced pooling size due to the small input) is then trained to classify the images based on the embeddings. The ADAM optimiser [104] is used with a learning rate of 5e-4, and early stopping [37]

[1] "Iterations" refer to the number of batches of data that the model is trained for. This is used consistently throughout the thesis.

with a patience of 50, up to a maximum of 1000 epochs.

Implementation: All implementation is in Keras [105] 2.2.4 , with the autoregressive section of the network forked and adapted from [106]. Albumentations is used [107] for augmentation. All datasets are open source and do not require ethical approval.

4.2.1 Statistical Tests

H_0: The two distributions have the same mean.

H_1: The distributions have different means.

A statistical test is needed to examine whether CPC performs better than ResNet, given a set amount of data. Multiple networks are trained and evaluated on either CPC embeddings or on the pure pixels. This gives a distribution of accuracies for each level of data, however, a way to compare these distributions is needed. There are a number of possible statistical tests that could be used, these are examined below:

T-test: One standard way of comparing two distributions is the t-test. The t-test is a statistic to estimate whether there is a difference in the means between two sample distributions. It assumes that the data is normality distributed. Due to this assumption, the t-test has not been chosen for statistical analysis for this chapter.

Wilcoxon signed-rank test: An alternative method would be to use a Wilcoxon signed-rank test. The Wilcoxon signed-rank test is a non-parametric replacement for the t-test in cases when the data samples do not approximate normality. However, the problem with this method is that it can only tell you whether the ranks of the differences are statistically significantly different, rather than the means of the two groups as would be the case if performing a t-test. For this reason, a choice was made to also use the bootstrap hypothesis testing method, described below, to test whether the means of the groups are statistically significantly different.

Bootstrap Hypothesis Testing: The goal of this experiment is to test whether the mean accuracy of the ResNet and the mean accuracy of the CPC ResNet is significantly different. To perform a statistical test, an assumption is made that the results come from the same underlying distribution, this will be the null hypothesis. A probability density function is then created for the difference in means, assuming that both sets of data come from the same underlying distribution. From this, the P-value for the actual sample can be calculated.

The algorithm is set out in pseudo-code below:

Data: x = distribution 1; y = distribution 2; z = x $\bigcup$ y
for $i < 1000000$ **do**
    set i = randomly sample from z, len(x) times
    set j = randomly sample from z, len(y) times
    difference in means = mean(set i) - mean(set j)
    store(difference in means)
    i++
end
PDF = the stored distribution of means

4.3 Experiments and Results

Three experiments are conducted to evaluate the CPC embedding mechanism: firstly, a study into whether the results found in [2] can be replicated on a medical imaging task. The endcoder is then evaluated to examine how robust these embeddings are to perturbation. Finally, examination of how well the embeddings translate to domain shifted image datasets under a data efficient transfer learning paradigm.

4.3.1 Learning from CPC embeddings

This section examines whether a ResNet trained on CPC embeddings will perform statistically significantly differently than a ResNet trained on pure pixels. The performance of a ResNet trained on CPC embeddings is compared to the same set of ResNets [2] trained on pure pixels for direct comparability.

Dataset: The first dataset used consists of 2000 colonoscopy images split into two classes from the HyperKvasir [10] dataset, collected as routine scans at a Norwegian hospital. Images containing either a polyp or no detected abnormality are used. Some images contain a green box in the bottom left hand corner which is used by the radiographer to assist with the procedure, however, they are not in all images. To ensure that this does not have an impact on performance of the network, this area is blanked over in all images.

Experimental Design: This experiment examines whether CPC can be used to improve predictive performance of a ResNet for polyp detection, when given limited labelled data (see figure 4.1). It compares the CPC framework to a ResNet trained on pure pixels. Two sizes of ResNet are used: ResNet-11 and ResNet-50. Examination is conducted on different subsets of the full training data to assess whether the learned CPC embeddings lead to performance improvement. The dataset is divided using a 80:20 train:test split; the training set is then randomly sampled 20 times to create 20 separate training and validation permutations. From these permutations an undersampled dataset is used for training. Samples of sizes 1%, 2%, 5%, 10% 20%, 50%, 100% of the training set are used. Classifier models are trained using the protocol set out in 4.2. For each subset, a test is conducted to examine whether using the CPC embeddings leads to any change in predictive performance.

[2] An argument can be made that this would not be a fair comparison as the models that have been pre-trained have benefited from additional training (the pre-training itself). While I acknowledge this argument, in this thesis I have followed the principal models should be held constant in their downstream training length and size, with the point of comparison being the initialisation weights.

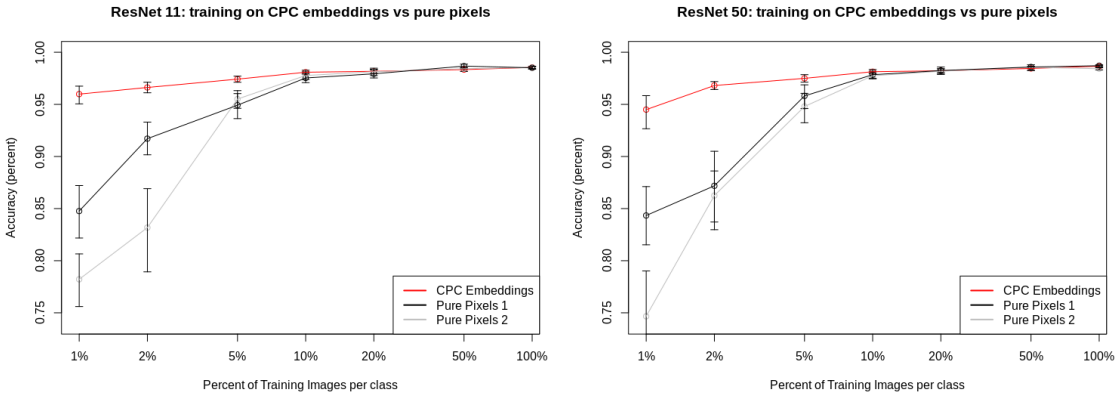

Figure 4.2: Mean classification accuracy of a ResNet trained on the pure pixels (shown in black) and a ResNet trained on the learned CPC embeddings (shown in red). Left shows a ResNet-11 and right shows a ResNet-50. A baseline with standard hyperparameters is shown in grey in both images.

Results: Figure 4.2 shows the performance of ResNets trained on pure pixels and CPC embeddings at different amounts of labelled data. There is statistically significant performance gains from the CPC embedding at: 1%, 2%, and 5% of the training data. As the amount of labelled data increases, the incremental gain diminishes until there is no difference between training with the CPC embeddings and pure pixels. This finding is consistent between both ResNet11 and ResNet50.

4.3.2 Robustness to Perturbation

Deep neural networks are susceptible to minor additions of noise massively changing the output [108]; good models should be robust to such challenges. In this section, a study on how these 'attacks' affect the performance of ResNet trained on either pure pixels or on CPC embeddings is conducted. In addition to investigating how these attacks impact downstream classification performance, this section also provides a technically novel mitigation strategy that helps reduce the impact of these attacks.

Perturbation Dataset: Three different augmentations are used that I conjecture will have an impact on the detection of a polyp, based upon work completed by [109] and using default augmentation ranges from [107]:

- **Brightness:** The brightness of the image will be randomly varied. This will have a limit of 0.2.
- **Noise:** Random Gaussian noise will be added. This will have a random variance between 10.0 and 50.0.
- **Blur:** The image will be blurred using a randomly sized filter up to a size of 7 pixels.

All images were generated ex ante, ensuring that all networks are tested on exactly the same test images. Ten derivative images are created for each image in the test set, giving a total of 4000 images for each 'adversarial' test set.

Experimental Design: The effect of these 'attacks' is examined on three sets of networks taken from the previous experiment (using the ResNet11s): one set of models trained on 1% of the training data; one set at 5% of the data; and one set at 100% of the data. For each of the perturbations, for each of the image amounts, for each of the types of ResNet, the performance of the network for polyp detection is evaluated on an augmented test set. Note, that no further training has been conducted. These results are given in table 4.1. A second experiment is then conducted with a technically novel mitigation is included. The same experimental setup is then repeated, however using an second encoder is trained using the protocol set out in 4.2, however, includes the perturbations in its training protocol to help mitigate its impact.

Table 4.1: Mean accuracy of ResNets either trained on CPC embeddings (Embs) or pure pixels when given test set with random perturbations (shown under "Normal Augmentation"). The second set of experimental results, showing the results from the technically novel mitigation being applied, is shown under "With Mitigation". **Bold** indicates significance.

		Normal Augmentation			With Mitigation	
	Augmentation	Pure Pixels	CPC Embs 1	P-value 1	CPC Embs 2	P-value 2
1%	Brightness	0.7123	0.6718	**0.0305**	0.8579	**<1e-4**
	Noise	0.6045	0.5181	**0.0163**	0.5217	**1.75e-2**
	Blur	0.8361	0.7173	**2.34e-3**	0.8044	0.113
5%	Brightness	0.7795	0.7256	**6.85e-3**	0.935	**<1e-4**
	Noise	0.5679	0.5513	0.310	0.54975	0.292
	Blur	0.9356	0.773	**<1e-4**	0.84	**3.30e-4**
100%	Brightness	0.9068	0.7429	**<1e-4**	0.9288	0.146
	Noise	0.7745	0.5676	**3.21e-3**	0.5731	**3.56e-3**
	Blur	0.9709	0.7854	**<1e-4**	0.8704	**<1e-4**

Results: Table 4.1 shows the mean performance of the selection of ResNets when undergoing 'attack' from various image perturbations. It can clearly be seen that in the case of all three 'attacks' that the ResNets trained on the CPC embeddings are more susceptible. In the second experiment, when the proposed, technically novel mitigation is included in the training protocol of CPC, this increases mean performance in 8/9 cases, however, in 4/9 cases, the ResNet trained on pure pixels still attains higher performance.

4.3.3 Domain Adaptation

Domain adaptation is an open challenge for machine learning [110], and one that is particularly relevant for medical imaging. The images captured by different scanners can vastly differ based on scanner parameters and brand, and the pathology may look completely different depending on population. This section evaluates how networks trained using CPC or pixels respond to this domain shift.

Datasets: This section tests the models on two datasets not used for training. The datasets are detailed in table 4.2.

Experimental Design: The models trained in section 4.3.1 are finetuned , then evaluated on datasets not included in the HyperKVASIR datasets to test ability of both sets of ResNets to adapt to a new domain. This includes data taken on different scanners and on different populations. For this set of tests, a subset of two

Table 4.2: Data description of the test sets used in section 4.3.3.

Dataset	Number of Images (polyps/non-polyps)	Notes
Child (a) [111]	200/800	Images of polyps in children, uses the same scanner brand as the training set
Child (b) [111]	100/300	Images of polyps in children and taken on a different scanner to the training set.

models are selected: the set of 20 models trained on 1% and 100%, and for comparison ResNets trained from scratch (i.e a new model trained on CPC embeddings of the new training set). Two model subsets were chosen, one at either extreme of dataset sizes to show that the results hold for different dataset sizes, while balancing against the increase in train time of finetuning all models across the subsets used in 4.3.1. The same training protocol described in section 4.3.1 is used. Their performance is then evaluated on the test sets of these datasets. For each of the models, eight images of both polyp and non-polyp images are randomly sampled from the relevant training set for finetuning [3]. In this experiment AUC was used as the evaluation metric rather than accuracy used in the rest of the chapter. The Child (a) and (b) datasets have unbalanced test sets and therefore reporting the accuracy figure could be misleading to the reader. For this reason, the evaluation criteria has been changed to AUC, a metric that is less susceptible to class imbalance.

Results: In 5/6 cases found in table 4.3, learning from Contrastive Predictive Coding embeddings results in statistically significantly higher performance than learning directly from the pixels. It should be noted that the pure pixel models trained on 100% of the data could not relearn the domain shifted features and achieved an AUC of approx 0.5. This outlier result has not been explored to understand why using the model trained on the pure pixels on 100% of the data failed to learn. One possible reason for this could be dead neurons: the process in which neurons with a ReLU activation layer stop contributing to the output of a model, therefore limiting its ability to learn. As this has not been studied any further, this is purely speculation.

[3] Throughout this thesis, finetuning refers to initialising a network with previously learned weights and training from this starting point.

Table 4.3: Classification AUC of ResNets trained on pure pixels or on CPC embeddings when tested on various testing sets when under domain shift. **Bold** indicates significance.

	From Scratch			1%			100%		
Dataset	Pure Pixels	CPC Embs.	P-val	Pure Pixels	CPC Embs.	P-val	Pure Pixels	CPC Embs.	P-val
Child (a) [111]	0.7549	0.7747	**1.67e-3**	0.7253	0.7720	**5.52e-3**	0.4990	0.7506	**<1e-4**
Child (b) [111]	0.7895	0.7868	0.141	0.6870	0.7652	**0.0378**	0.5333	0.7762	**<1e-4**

4.4 Extension to Other Datasets

On the basis of the encouraging results results found in the previous section, the application of CPC to medical imaging tasks has been extended to three more datasets. In this section, the performance of CPC on an extended colonoscopy dataset, OCT and dermatology images are explored. Initially, the ability of the networks to learn useful embeddings is examined. The experiment described in 4.3.1 is repeated on these extended datasets.

4.4.1 Datasets

This section uses three medical imaging datasets across a range of imaging modalities and pathologies, to ensure its general applicability. With all datasets, a random subset of the relevant datasets is chosen which is then split into training and testing sets. The training set is further split into a training and validation set, the latter being used for early stopping [37] a technique to prevent overfitting on training data. It works by monitoring a validation set that is not part of the training set, and stops training when the validation loss does not decrease for a set number of epochs, the value of which is called the "patience".

Colonoscopy: A dataset of 3000 colonoscopy images from the HyperKvasir [10] dataset is split into three classes. Images are used containing: a polyp; no detected abnormality; and one class combining the different grades of colitis into a single

class. The HyperKVASIR dataset also contains an unlabelled dataset of gastro intestinal tract endoscopy images which were used for training the CPC encoder. This dataset is an extended version of the one used in the previous experiment, with the addition of images containing colitis.

Optical Coherence Tomography: Retinal Optical Coherence Tomography (OCT) is an imaging modality used to take three dimensional scans of the retina. These scans can then be interpreted by an physician to identify a number of retinal pathologies such as macular degeneration and diabetic macular oedema. In the UK, OCT scans may be interpreted by non-physicians [112], therefore automatic interpretation could help with reducing missed diagnoses. A subset of [11] dataset is used. The subset consists of four classes of images: choroidal neovascularization ("abnormal blood vessels grow[ing] into the retina and leak fluid" [113]); diabetes-related macular edema ("a complication of diabetes caused by fluid accumulation in the macula that can affect the fovea" [114]); drusen ("deposits of cellular debris that accumulate under the retina" [115]); and images where no pathology is present. 8000 randomly selected images split between four classes are used for supervised training and the full dataset is used for encoder training.

Dermatology: Skin cancer is an increasingly common form of cancer that affects between 2 and 3 million people worldwide each year [116]. Diagnosis relies on interpretation of photographic images of the lesion. A subset of the HAM10000 [12] dataset is used utilising 3000 images split into three classes: Benign lesions of the keratosis (a non cancerous lesion); Melanoma; and Melanocytic nevi (pigmented moles [117]). The full dataset is used for unsupervised training.

4.4.2 Learning Useful Features

The CPC encoder is not explicitly trained to produce useful encodings for any one task, and therefore, there is no guarantee that the features learned will be any more useful than a random projection. An evaluation is conducted into whether the CPC encoder has learned useful features for a medical imaging classification task across

three datasets.

Experimental Design: Two sets of linear layers are trained on each dataset: one set starting from frozen CPC embeddings; and one starting from the frozen embeddings produced by a randomly initialised encoder. Note that while the networks are frozen, the linear layers are not, allowing them to learn if the features are useful for the downstream task. The layers are trained on a randomly selected 1% subset of the full datasets. The layers are trained for a maximum of 1000 epochs, using early stopping with a patience of 50 to prevent overfitting. The ADAM optimiser with a learning rate of 5e-4 is used.

Statistical Analysis: 20 experimental repeats are conducted, with seeded randomisation for reproducibility, comparing the distribution of the mean estimated using the bootstrap statistical test [118]. 1000000 samples with replacement are used to estimate the probability distribution.

Results: Table 4.4 outlines the mean performance of both sets of linear layers. In all cases, the linear layers trained on CPC embeddings have a statistically significant improvement over training on a random projection. This result indicates that the CPC pre-training can learn features that are useful for classification without any explicit training for this task. Note: random chance is 33.3% for the colon and dermatology dataset, and 25% for the OCT dataset. While statistically significant, the results for OCT is poor.

Table 4.4: Linear layers trained on either CPC embeddings or on embeddings produced by a randomly initialised encoder. Networks are trained using a randomly selected subset of the full dataset, consisting of 1% of the images. **Bold** indicates significant result.

Dataset	CPC Mean Accuracy	Random Mean Accuracy	P-value
Colon	0.7681	0.3348	**<1e-4**
OCT	0.2726	0.2496	**1.87e-4**
Dermatology	0.6355	0.3384	**<1e-4**

4.4.3 Classification from CPC embeddings

In this section, an investigation is presented into whether training a ResNet on CPC embeddings will achieve a statistically significantly different performance than a ResNet trained directly on the image pixels. The experiment is conducted on the three datasets outlined in section 4.4.1.

Experimental Design: The performance of ResNet-11s trained directly on the images is compared to the performance of ResNet-11s trained on CPC embeddings of those images. This follows the same model training design set out in section 4.2 for both encoder training and classifier training. A pooling size of two is used within the second ResNet for learning from CPC embeddings (compared with eight for the pure pixels) due to its smaller input size. For each dataset, the performance is evaluated in {1%, 2%, 5%, 10%, 20%, 50%, 100%} sized subsets of the full dataset. As with the previous experiment: the networks are trained for a maximum of 1000 epochs, using early stopping with a patience of 50 and the ADAM optimiser with a learning rate of 5e-4.

Statistical Analysis: 20 repeats are conducted with the same randomised seeds as in the previous experiment, reporting the mean and 95% confidence intervals.

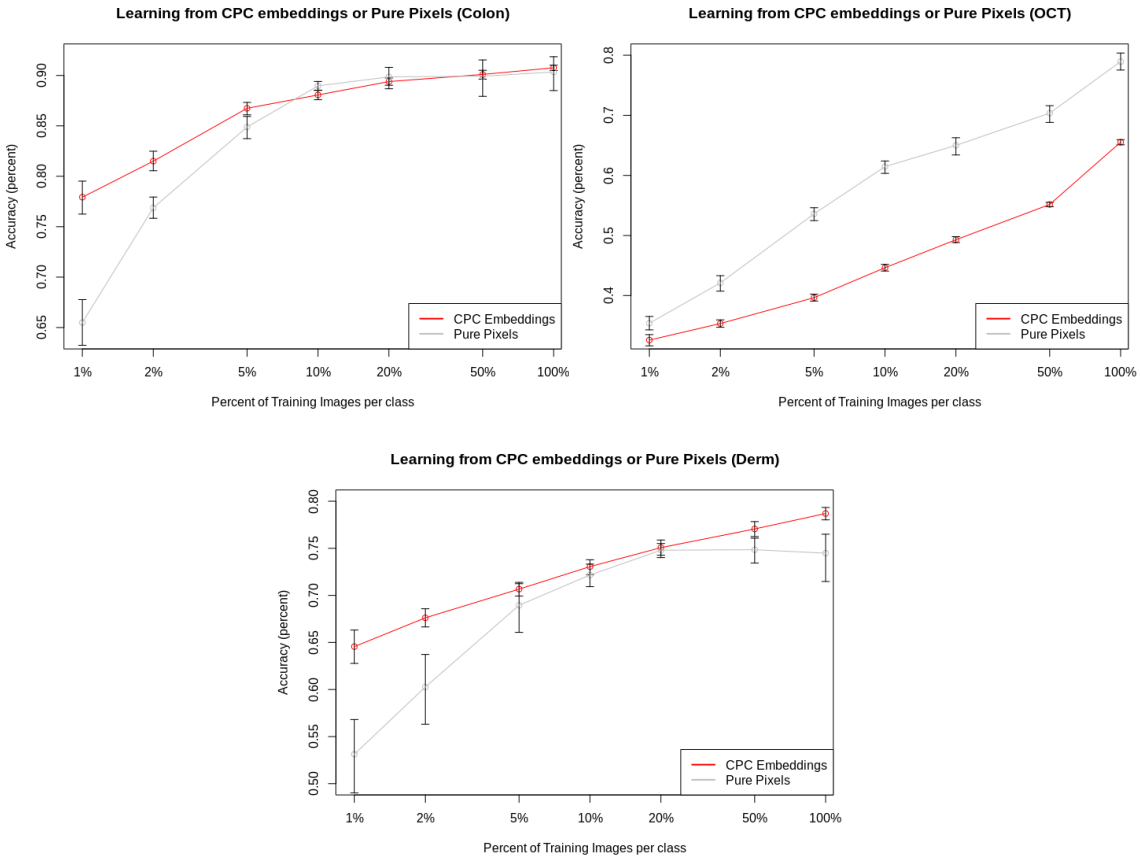

Figure 4.3: ResNets trained on either the pure pixels or on the CPC embeddings. Showing: Colonoscopy, OCT, Dermatology. Non-overlapping bars indicate significance.

Results: Figure 4.3 compares the performance of two sets of ResNets across three datasets. There are marginal performance gains from learning from CPC embeddings on two of the datasets, colonoscopy and dermatology particularly when trained on limited data. However, the ResNet trained on pure pixels of the OCT dataset outperforms the ResNet trained on CPC embeddings. Hence, learning from CPC embeddings may result in higher performance than learning directly from the pixels. However, it is likely that it does not work for all limited labelled data tasks.

4.5 Discussion

This Chapter has presented evidence for the application of Contrastive Predictive Coding to improving the performance of medical image classification when given limited labelled training data. Experiment 1 (section 4.3.1) showed that learning from CPC embeddings can increase accuracy of polyp detection by 11.2% when given limited labels example. Despite this, experiment 2 (section 4.3.2) found that models trained on CPC embeddings are less robust to perturbation than models trained directly on the images. Experiment 3 (section 4.3.3) showed that, despite the results of experiment 2 (section 4.3.2), the models trained on CPC embeddings achieved higher AUC on a transfer learning task than the model trained on the pixels directly. Based on the positive results of section 4.3, further experimentation was conducted to examine whether these positive results would extend to further datasets (section 4.4). This section found that while CPC improved performance on colonoscopy and dermatology images, it failed to increase performance for classification of OCT scans.

Different imaging modalities have varying levels of suitability for use with Contrastive Predictive Coding (and semi-supervised learning more generally): Colonoscopy produce a large number of images that can be used as an unlabelled dataset, with each procedure producing between 30 and 45 minutes of video from which, each frame can be extracted to produce an unlabelled training set. This differentiates colonoscopy from other medical imaging tasks, in which, one patient usually produces one image: such as in the case of chest x-ray. This could be one explanation for why there has been limited work exploring CPC on medical imaging tasks, and those that exist focus on limited modalities such as histopathology images [92] [119] [120] (Histopathology images are usually very high resolution images, which are able to be split into many sub images to create the unlabelled dataset). Despite these theoretical arguments, CPC performed poorly on OCT scans and well on dermatology photographs, therefore, the suitability of a dataset for use with CPC is dictated by far more than just dataset availability. Based on this result, chapter 7 investigates how the dataset affects performance of a semi-supervised method.

Ensuring the safety and reliability of AI systems is of paramount importance, particularly for safety critical applications such as those in the medical field. It is imperative that we study the robustness of these systems to factors that may dramatically affect their performance. This chapter has shown that, despite the networks trained on CPC embeddings performing better in accuracy when given limited data, they are far less robust to perturbation than the networks trained directly on the pixels. This finding could have a large impact on the utility of this method for safety critical applications. In experiment 2 (section 4.3.2), a basic mitigation strategy is proposed to help increase the robustness of the model to perturbation, however, this fails to increase the performance to that of the model trained directly on the pixels. While beyond the scope of this work, further investigation is invited into possible mitigation methods that could increase the model performance to equal to, or greater than, that achieved by learning from pure pixels. A second limitation of this experiment is that synthetic corruption has been used, which does not take into account the likelihood of these appearing in a natural setting. This may go some way to explaining the difference in result between experiments 2 (section 4.3.2) and 3 (section 4.3.3).

Experiment 3 (section 4.3.3) found that learning from CPC embeddings produced models that were more robust to domain shift than learning directly from the pixels. This suggests that the representations learned by the CPC encoder training are more generalisable than those learned directly from the pixels under a supervised training regime. If this can be shown to be true across a larger number of datasets, this paper, along with these other, would show that the assertions from [6] that CPC learns higher level features to be true. Section 7.11 details an experiment in which the generalisability of CPC investigated, finding that the features learned by CPC are general features.

This chapter gives two principles for when CPC embeddings may be most appropriate for increasing performance: Experiment 1 (section 4.3.1) of the work shows that as the amount of data increases, the performance of the two network types converge, showing that any benefit of CPC is only felt at extremely low data sizes.

Experiment 2 (section 4.3.2) shows that CPC is less robust to perturbations of the test data. Based on the proposed mitigation, further work should be undertaken to study the effects of the chosen set of augmentations on the downstream performance.

Limitations: Despite the contributions of this work, there are a number of limitations that should be highlighted to the reader. Firstly, due to the computational complexity of training semi-supervised models, the CPC models used throughout this chapter have been trained for 60k iterations which was equivalent to 1 epoch of training. This likely lead to a sub-optimally trained model. Additionally, with the exception of the work in 4.3.1, the downstream ResNet in all other experiments was a ResNet-11, much smaller than the ResNet-50 used in [2]. In chapter 8, I hypothesise on how these changes may impact on interpretation of the results, however, this is just speculation. Care must be taken with these results as it is possible that these results do not extend to larger network sizes and training lengths.

In addition to the limitations outlined above, section 4.3.2 has outlined that ResNet-11s trained on CPC embeddings are more susceptible to perturbation that ResNets trained on pure pixels. This goes against the conventional narrative that they are less likely to overfit. Despite this counter intuitive finding, no attempt has been made to qualify why this has happened.

4.6 Conclusion

Contributions: Most AI studies of colonoscopy reported in literature [121] [122] [123] [124] demonstrate that where extensive training datasets are available, networks can achieve remarkable performance. However, accessing high-quality, labelled datasets remains challenging. This chapter provides evidence for how a self-supervised framework can be leveraged to enhance supervised performance on small labelled datasets. The contributions of this chapter can be summarised as follows:

- Evidence is presented that using the proposed framework with CPC pre-training, prediction performance can be increased by over 10 percentage points

on unseen datasets when trained with limited labelled data.

- Prior work shows that deep learning can be very susceptible to adversarial attacks [108]. This work demonstrates that ResNets trained with CPC are more susceptible to perturbation than ResNets trained directly on pixel data.

- This work shows that ResNets trained on CPC embeddings perform better, or no worse than, ResNets trained on the pure pixels under a data-limited transfer learning scenario for the task of polyp detection.

- Finally, this work is extended to three other medical imaging datasets, showing mixed results: confirming that CPC works on some, but not all, datasets.

This chapter has shown that ResNets trained on CPC embeddings can achieve higher polyp classification performance than ResNets trained directly on the images. Despite this, this work also identifies a possible weakness of the CPC method and proposes a mitigation for this. Additionally, it shows that learning from CPC embeddings performs statistically significantly better than learning from pure pixels under domain shift. Based on these encouraging results, the final section of this work extends the evaluation to three additional datasets, finding that learning from CPC embeddings improves performance on the extended colonoscopy and dermatology datasets, but not the OCT dataset, leading to the conclusion that CPC is not a universal method for improving performance on small datasets.

Link to the aims: This thesis aims to investigate semi-supervised learning and to understand how best to apply this set of methods to obtain the highest performance. This chapter has shown that, under very low labelled data regimes, Contrastive Predictive Coding (CPC) can achieve higher classification accuracy on a medical imaging task. It most importantly shows that, as the amount of labelled data increases, the relative advantage of the semi-supervised method diminishes. This shows the set of circumstances in which semi-supervised learning is likely to be the most useful: in situations where the cost of acquisition of extra labelled data is extremely cost prohibitive, such as in the case of rare diseases. It also shows that it is likely to have less utility in the case where the cost of acquisition of more labelled data is relatively cheap, such as when labelled data can be collected from non-specialists, eg AWS's Mechanical Turk [40].

Chapter 5

SimCLR Background

In addition to Contrastive Predictive Coding, there are numerous other contrastive learning protocols for unsupervised representation learning. SimCLR [1] is one such method. As with CPC, the SimCLR protocol trains an encoder in an unsupervised fashion which can then be used to perform a supervised task, hopefully reducing the number of labelled examples one needs to achieve an acceptable performance. SimCLR uses a contrastive loss to maximise the agreement between the latent encodings of two observations of the same image with different augmentations. This forces the network to produce augmentation invariant embeddings; with the idea that this will produce better semi-supervised performance. SimCLR is a conceptually simple framework for contrastive learning compared to related methods such as MoCo and CPC. Despite its simplistic nature, SimCLR has been shown to produce high performing results on multiple datasets [1] [3].

5.1 Network Description

SimCLR is a contrastive method (see chapter 2) which attempts to learn features that are invariant to a number of random augmentations. It does this through maximising the agreement between different 'views' of the same image in the encoder latent space. A diagram is presented in Figure 5.1: here, an image is transformed by two random transformations, with each then projected into a latent space using an encoder (in this work, a ResNet). This representation is then further transformed

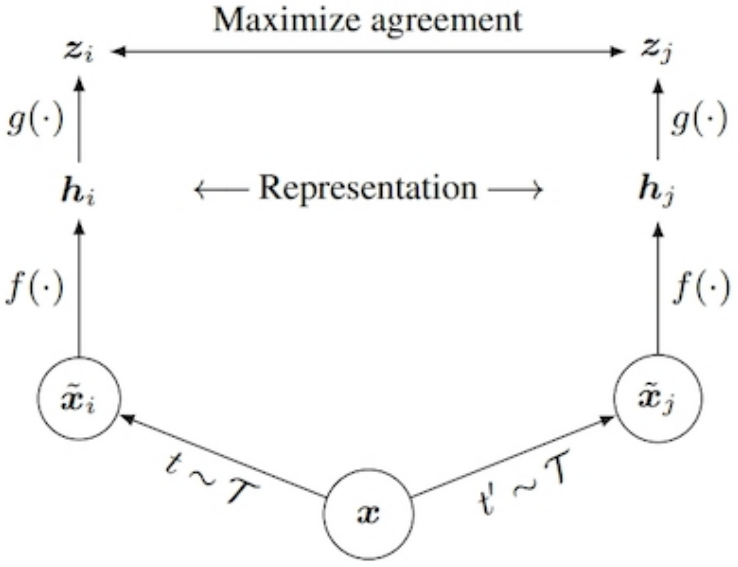

Figure 5.1: A diagrammatic representation of the SimCLR network. x is the input image, $\bar{x}_i$ is the transformed image, $f(.)$ is the encoder network, $g(.)$ is the projection head network, z_i is the final representation.

using a non-linear projection head.

For each image, x, in the dataset; two sets of of transforms t and t' are applied separately to the image. This produces two randomly transformed versions of x: producing $\tilde{x_i}$ and $\tilde{x_j}$. Each of these transformed images are projected from image space to latent space using an encoder, $f(.)$ (after training is complete, this is the representation that will be used in the downstream task). A non-linear projection head, $g(.)$ is applied to these representations, forming z_i and z_j. A contrastive loss is then applied to the set of projected representations, maximising the agreement between the two z vectors.

SimCLR is a concatenation of a number of proposed components of other semi-supervised techniques, distilled down into a simple to implement framework which does not require specialised architectures (unlike CPC) nor a memory bank (unlike MoCo). The authors note that "almost all individual components of [their] framework have appeared in previous work, [...] the superiority of [their] framework is not explained by a single design choice, but by their composition". These components

are detailed below:

- **Data Augmentation:** Data augmentation takes a large role within the SimCLR framework. As with a large number of contrastive approaches, SimCLR attempts to minimise the distance in latent space between two 'views' of the same image: here taking the form of two stochastic augmentations of the same image. This differs from Ye et al [125] in that both views are augmented, rather than in Ye's case which are minimising the distance between an unaugmented and augmented version of an image. In SimCLR, the authors claim that the use of random crop and colour distortion is "crucial to achieve good performance", however, this assertion is likely to only hold for object centric image tasks, such as ImageNet.

- **Encoder:** Ultimately, the goal of SimCLR is to learn useful representations that may then be used in some 'downstream task'. Under this and related frameworks, the representations take the form of latent 'codes', i.e a vector representing the image. To produce this code, a mechanism to project the image from image space to embedding space is needed. SimCLR trains a ResNet encoder to act as this encoder, however, there is no principled reason that this cannot be any arbitrary neural network.

- **Projection Head:** Rather than applying the contrastive loss function directly to the output of the embedding, Chen et al find that using a non-linear projection head increases performance by 10% over no head, and by 3% over using a linear projection head. This approach has been taken by prior work [6], but the basis for doing so is just empirical. Chen et al hypothesise that this is due to the head allowing the encoder to keep some features that are not useful for solving the contrastive task, but are useful for the downstream task.

- **Contrastive Loss Function:** SimCLR optimises a contrastive loss function which seeks to maximise the agreement between the two embeddings of two views of the same image and maximises the disagreement between all other examples in the batch. This loss is termed NT-Xent and is studied in further detail in the section below.

5.2 Loss Function

SimCLR uses a contrastive loss function similar to the InfoNCE loss function introduced in the Contrastive Predictive Coding chapters, chapters 3 and 4. This loss function minimises the difference between embeddings of similar pieces of data (in this case, similar is defined as being two different transforms of the same image), and maximise the difference between dissimilar pieces of data (dissimilar being defined as any pair of images that are not stochastic transforms of each other).

$$l_{i,j} = -log\frac{exp(sim(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} 1_{[k\neq i]} exp(sim(z_i, z_k)/\tau)} \tag{5.1}$$

In this equation, the objective is to 'identify' the pair of images that correspond to the same image transformed with the two separate transforms. This loss function is a simple contrastive loss function, similar to the loss function found in Contrastive Predictive Coding (chapters 3 and 4), however, with 2 main changes: 1) a temperature scaling parameter; 2) the similarity metric is changed to be the cosine similarity. The temperature scaling parameter has been introduced as a performance enhancing change, this has also been found in numerous other works [71] [126] [16]. Rather than using the dot product ($z_i^T z_j$) of two vectors as the similarity metric - seen in CPC - SimCLR opts to normalise this vector and use the cosine similarity metric ($sim(z_i, z_j) = z_i^T z_j/|z_i||z_j|$). This is despite Noise Contrastive Estimation explicitly not requiring the vector to be normalised, and instead requiring the encoder to learn an embedding function that produces normalised vectors [90].

5.3 Improving Network Performance

In [1], Chen et al make a number of conjectures of how to set up SimCLR to gain the best performance out of the framework. They state the following:

- **Use large batches:** Using larger batch sizes was consistently shown to increase the performance of the network, with the largest jump in performance when the network is trained for fewer epochs. Because SimCLR does not use a

memory bank, its batch sizes are much smaller than can be found in other work such as MoCo [68]. This, however, does seem to have a limit. Further study showed that the increase in performance peaked at approximately a batch size of 8192. Additional study is needed to ensure this holds for other network sizes and datasets. This direct sampling of the negative examples leads to higher memory requirements than would be needed with a memory bank.

- **Use a non-linear projection head:** Applying the contrastive loss on a non-linear projection head rather than directly on the encoder output is shown to improve performance. Both SimCLR and Contrastive Predictive Coding apply the contrastive loss to a projection of the embeddings rather than the embeddings directly. Three possibilities were investigated, applying the contrastive loss directly to the representations, applying it to a linear projection of the representations, or applying to a non-linear projection of the representations. A non-linear projection was empirically found to give the best representations.

- **Choice of transforms matter:** A number of different transforms were tested in the SimCLR paper. No single augmentation was found to be good enough to learn high quality representations, however, augmentations could be composed together to make a more difficult task, which would then lead to better representations. The authors found that random-crop composed with colour-distortion and random blur produced the best results, however, I hypothesise that this will be dataset specific, particularly in the case of non general imaging datasets.

- **Size of network matters:** Increases in both depth and width were found to lead to higher performance. This is also consistent with what was found with Contrastive Predictive Coding. While the same finding is true of supervised learning, the authors found that the gap between supervised and linear layer on unsupervised network narrowed as the size of the network grew. They claim that this indicates that SimCLR benefits more from the increase in size than in the supervised approach.

5.4 Use in Literature

Unlike with Contrastive Predictive Coding (CPC) in chapters 3 and 4, SimCLR has been well utilised in a number of areas. In this section, the literature is examined to explore how, and to what problems, SimCLR has been applied. A large number of works use SimCLR as a powerful baseline to their proposed model rather than directly evaluating the method:

A large study [127] was conducted examining SimCLR on both chest x-rays and dermatology images. A number of investigations were undertaken: Azizi et al concurred with [128] that training the unsupervised model on ImageNet, then training the model again on an unlabelled domain specific dataset before finally training on a labelled dataset, produces the best results. In Azizi et al's testing of SimCLR, they claimed that SimCLR lead to higher performance than their ResNet baseline for both the dermatology dataset and the chest x-ray dataset however, the level of improvement seen on the chest x-ray dataset is very small, with a 0.0046 increase in AUC when just using an ImageNet unsupervised dataset and a 0.0104 increase in AUC when using the transfer learning approach outlined before. While not directly comparable because of differing metrics, this is a far smaller relative improvement than was apparent with the dermatology dataset, indicating that the SimCLR method has differing efficacy with different datasets, as was found with CPC in chapter 4. [129] also found that SimCLR worked well on dermatology data, but found that other contrastive methods like MoCo performed even better.

[130] found that SimCLR did not perform as well as either a supervised baseline nor supervised contrastive learning when evaluated on CIFAR 10 and 100 datasets. This result was consistent when evaluated on brain MRI images: SimCLR performed worse than their proposed method and also a supervised baseline. It is not clear whether the authors finetuned the model or these are the results of a linear evaluation layer. [131] also explored SimCLR as a baseline on MRI images, finding that SimCLR outperformed MoCo [68] in most cases. In contrast, [132] found that SimCLR only performed slightly better than a much simpler rotation predic-

tion pre-training task. [133] also found that SimCLR performed only on par with other methods tested. [134] showed SimCLR outperforming even their own proposed model (without a time based component).

A number of papers that have evaluated methods have found that other contrastive methods have performed better that SimCLR, most notably: MoCo [68]. As stated previously, [129] found that MoCo worked better on dermatology images. In [135] SimCLR performed almost 15 percentage points worse than MoCo on a histology dataset and worse than the supervised baseline that MoCo was able to either match or beat. In [97] SimCLR performed very poorly when evaluated using a frozen network and linear layer, but comparatively well when finetuned, this was one paper which showed an higher performance for CPC than for SimCLR.

Some papers have attempted to analyse the impact of testing the network on a different domain to the dataset originally trained on. This kind of analysis is important when wanting to apply a method to a medical imaging task, due to the vastly different appearances of images when changing parameters such as MRI scanner settings. In addition, the acquisition of medical images is far harder and costlier than general images, leading to much more varied datasets being used. In [136], the authors found that a transfer approach, where a network trained on ImageNet, then on the domain dataset worked better than training solely on just the domain dataset. [137] also examined various methods, including SimCLR on transfer learning from a general imaging task to a medical imaging task. They agree with [136] that ImageNet to domain specific produces the best results, possibly due to the network creating more robust features from the larger, but undrelated, dataset.

SimCLR has also been adapted to be used on other types of problems outside of image classification: [138] adapted SimCLR to embed electronic health records, marginally improving critical care outcome prediction; [139] adapted SimCLR to detect out of distribution skin lesions, reducing the need for even labelled data.

5.5 Comparison to Related Methods

As was covered in section 5.1, SimCLR did not introduce any new ideas: its contribution to the literature was the composition of these ideas together to produce a method that was greater than the sum of its parts. As such, SimCLR takes heavily from previous methods which have been analysed here.

Contrastive Predictive Coding: The contrastive Predictive Coding protocol was developed to force an encoder to learn high-level features while discarding high frequency information such as noise and texture. It does this by optimising an encoder to produce embeddings that are predictive of other patches in the same image, while ensuring that they are far apart from any other image in the dataset. Fundamentally, the idea is that features that are consistent across pieces of an image will be high level features which will be useful for prediction. In contrast, SimCLR also hopes to learn high level features, however, their approach relies on the belief that high level features will be consistent across transforms.

Due to their similar philosophy, SimCLR and CPC do share similar loss functions with the SimCLR loss being based upon the loss from CPC. One of the major differences with the loss function is the normalisation of the feature vectors found in the SimCLR loss. Normalisation of the feature vectors is not required, and is specifically excluded from Noise Contrastive Estimation (from which the InfoNCE loss is based), however, there is empirical evidence that higher performance can be achieved through normalisation [1].

CPC and SimCLR also have different 'views' of the image. CPC embeds an image, through embedding overlapping patches into a latent space, thus producing a 2D array of vectors for each image. This means that each feature vector can only encode smaller features than could be available across the full image. SimCLR embeds the full image down to a latent space, thus being able to embed any sized feature that could be invarient to the transforms. This is one possible reason for the better performance found in the literature, as a large number of imaging datasets consist

of 'object centric' data, with a single object in the image with a background (i.e similar to ImageNet). However, medical imaging datasets may not conform to this paradigm and, therefore, performance may not be higher on these datasets.

Momentum Contrast: Both MoCo and SimCLR work upon similar principles: they contrast a set of negative examples with a singular positive example thus learning an embedding which places similar data together in the latent space. However, how these negative examples (the "noise" data found in InfoNCE) are sampled differs between the two methods. SimCLR directly samples negative examples, whereas MoCo uses a memory bank of previously used negative examples. This memory bank technique can also be found in PIRL [16] and InstDisc [140]. By directly sampling the negative examples, it allows the network to learn from negative examples using the current encoder, rather than an out of date one as would be found with MoCo. This comes at the expense of increased computational complexity.

As was covered in chapter 2, MoCo-v2 introduced a number of the proposed improvements from SimCLR: using a non-linear projection head and using stronger augmentations. As with SimCLR, it was found empirically that both of these additions increased the power of the network, leading to greater performance. As discussed in chapter 3, "stronger augmentations" in the form of patch-based augmentation was also found to increase performance of contrastive predictive coding. It is certainly possible that these could be universal improvements to the contrastive approach.

5.6 Areas of investigation

[1] introduced SimCLR, a collection of non-novel components bundled into a novel package, partially building on the work introduced by Contrastive Predictive Coding (chapters 3 and 4). Both methods rely on the embedding of similar images together in the latent space and dissimilar images apart, however, there are differences in the networks' architecture, which affects how each of the networks 'see' the image.

In addition, there is a difference between what each method classes as similar or dissimilar images. In the CPC method, similar patches are classed as patches that come from the same image, while in SimCLR, similar images are seen as the different transforms of the same image. This may go some way to explaining differences in the application of the two methods, with SimCLR being replicated a far greater number of times.

While there was some mixed results found in the literature, generally the SimCLR method performed favourably to related methods. In addition, unlike CPC, there has been a large number of pieces of work applying SimCLR. These two factors are likely related. Due to the large number of replications of the SimCLR method, it is not necessary to perform the same type of experiment conduced in chapter 4: testing whether SimCLR can boost the performance over the performance of some baseline, since this would not contribute much to the general discussion surrounding SimCLR.

Instead, the next two chapters (chapters 6 and 7) have conducted studies examining how certain aspects of design choices affects the performance of SimCLR. In chapter 4, experimentation was conduced to examine how the size of the labelled dataset affects the downstream classification performance in addition to experiments on domain shift and adversarial attack using augmentation. In chapter 6, the network builds on the work of Hénaff, evaluating how the addition of patch-based augmentation affects downstream classification performance. From this starting point, the impact of augmentation on downstream performance is evaluated. In chapter 7, the work on dataset analysis (chapter 4) is continued: evaluating how the characteristics of the unlabelled dataset affect downstream classification performance.

Chapter 6

Understanding Data Augmentation for Contrastive Models

Abstract

SimCLR is a method that relies heavily on the augmentations used during unsupervised training. In this chapter, a number of related experiments are conducted to explore the impact that the choice of augmentation protocol has on the downstream task performance of contrastive methods using ResNet-11s. Initially, the type and magnitude of augmentation are investigated. This initial section finds that the performance of the network with various augmentations will depend on the exact task, therefore, hyperparameter tuning should be undertaken for each task and new dataset. This chapter then investigates the claim that stronger augmentation leads to greater performance: finding that this conjecture holds for colour based augmentations. However, it also finds that performance of the crop augmentation stays steady, before rapidly dropping off, therefore heavy augmentation actually harms performance. Further study finds that heavy additive noise does not increase performance above limited noise; and more rotation does not increase performance. In addition, this chapter evaluates the claim that SimCLR produces representations that are invariant to augmentation. Under the experimental design in section 6.6,

SimCLR produces representations that are more invariant than a supervised training protocol. This experiment is then extended to conclude that augmentation during the supervised downstream task is still necessary to achieve the best performance.

6.1 Introduction

SimCLR has been shown to produce state of the art results when applied to datasets with relatively large unlabelled datasets along with small labelled datasets. This method is heavily reliant on the composition of multiple, randomly applied augmentations, forcing the encoder to learn representations that are invariant/predictive of other augmented copies of the same image. Current work has suggested that a limited sample of total possible augmentations is all that is needed to produce good results, however, it is likely that this not to be true for all tasks; particularly non object-centric datasets such as medical imaging.

In this chapter, a number of experiments are conducted to investigate how the choice of augmentation strategy impacts the downstream performance of the SimCLR protocol. Initially, the work of chapter 4 is built on: investigating whether the suggestion of [2], to introduce patch-based augmentation to Contrastive Predictive Coding in order to increase performance, transfers to medical images. Secondly, the augmentation strategy proposed by SimCLR is investigated, specifically, evaluating the choice of data augmentation composition on medical imaging tasks. Then, another claim of SimCLR is evaluated: do high levels of augmentation always produce increased performance across different augmentations? Finally, the conjecture that SimCLR learns representations that are invariant to augmentation is assessed.

6.2 Background

Data augmentation is an important part of the current deep learning orthodoxy. Supervised learning has utilised augmentation for increasing the generalisability of deep neural networks. As the size of deep learning models has grown from 6.8 million parameters in GoogLeNet [141] to modern works such as the 174.6 billion parame-

ter GPT-3 [75], the ability of these networks to overfit on even very large datasets has increased. By applying random - but realistic - augmentations to input data, the network becomes more robust to variation within unseen data, and produces a model that is less likely to have overfit on the input dataset.

In addition to supervised methods, the contrastive methods found in chapter 2 also utilise augmentation extensively. [1] showed the importance of choice of augmentation: Chen et al found that no single augmentation produced good representations, however, multiple augmentations composed together did produce good representations. [142] improved the performance of the Momentum Contrast method [68] by increasing the power of the augmentations used during training. [143] investigated a subset of data augmentations including {cropping, rotation, colour distortion, grey scaling} finding that including all four augmentations together produced the greatest result for diabetic retinopathy detection. Colour distortion and greyscaling was found to have the largest singular effect, showing that creating colour invariant feature representations are just as important as in general imaging (i.e ImageNet) tasks. This is likely not to be the case in all medical imaging tasks. For example, dermatology image diagnosis can rely on colour. Despite this theoretical limitation, [127] used colour augmentation for medical imaging and found good results. No clear advice on which augmentations should work well on medical imaging tasks has been produced: therefore further work is needed.

6.3 Impact of Patch-based Augmentation for Contrastive Predictive Coding

Chapter 4 showed how Contrastive Predictive Coding could be used to improve the performance of a ResNet when learning from limited labels. As part of CPCv2, patch-based augmentation was introduced as a method to improve the downstream classification performance of the protocol, with [2] showing that it improved performance by 4.5% on an arbitrary dataset. In this section, this work is extended to a

medical imaging setting. The effect of this design choice may have a different impact upon medical imaging than more object-centric datasets. In medical datasets, features such as colour may have a far more important role in classification than in datasets such as ImageNet. This limitation of the patch-based augmentation was alluded to in [2], but no study was conducted.

6.3.1 Experimental Design

This experiment takes a similar form as the experiments found in chapter 4: for each of the datasets under study (section 4.4.1), two encoder networks are trained, one with, and one without patch-based augmentations. The training protocol from section 4.2 is followed with the only change being the addition of patch based augmentation. These encoders are trained for 60k iterations on each dataset. From these learned embeddings, ResNets are trained for image classification. The ResNet's ability to learn from each type of encoding is evaluated in {1%, 2%, 5%, 10%, 20%, 50%, 100%} sized subsets of the full dataset. For each encoder, 20 ResNets are trained on the embeddings produced by the respective CPC encoder. The ResNet networks are trained for a maximum of 1000 epochs, using early stopping with a patience of 50 and the ADAM optimiser with a learning rate of 5e-4. The mean value is reported with 95% confidence intervals.

6.3.2 Results

Figure 6.1 shows the impact of introducing patch-based augmentation across three medical imaging datasets (colonoscopy images, Optical Coherence Tomography (OCT) scans, and dermatology photographs) at varying sized subsets of the labelled dataset. [2] states that introducing patch-based augmentation increases performance. The results of this section contradict this: they show that introducing patch-based augmentation reduces performance across all subsets in the OCT and dermatology datasets, and decreases performance on the colonoscopy dataset in the larger sized subsets (but does increase performance at the 1% subset). It is possible that the set of augmentations that produced the greatest effect for a dataset such as ImageNet, are not

the same augmentations that will produce the greatest effect in every task. Based on these results, the next section examines the composition of various augmentations using SimCLR on medical datasets.

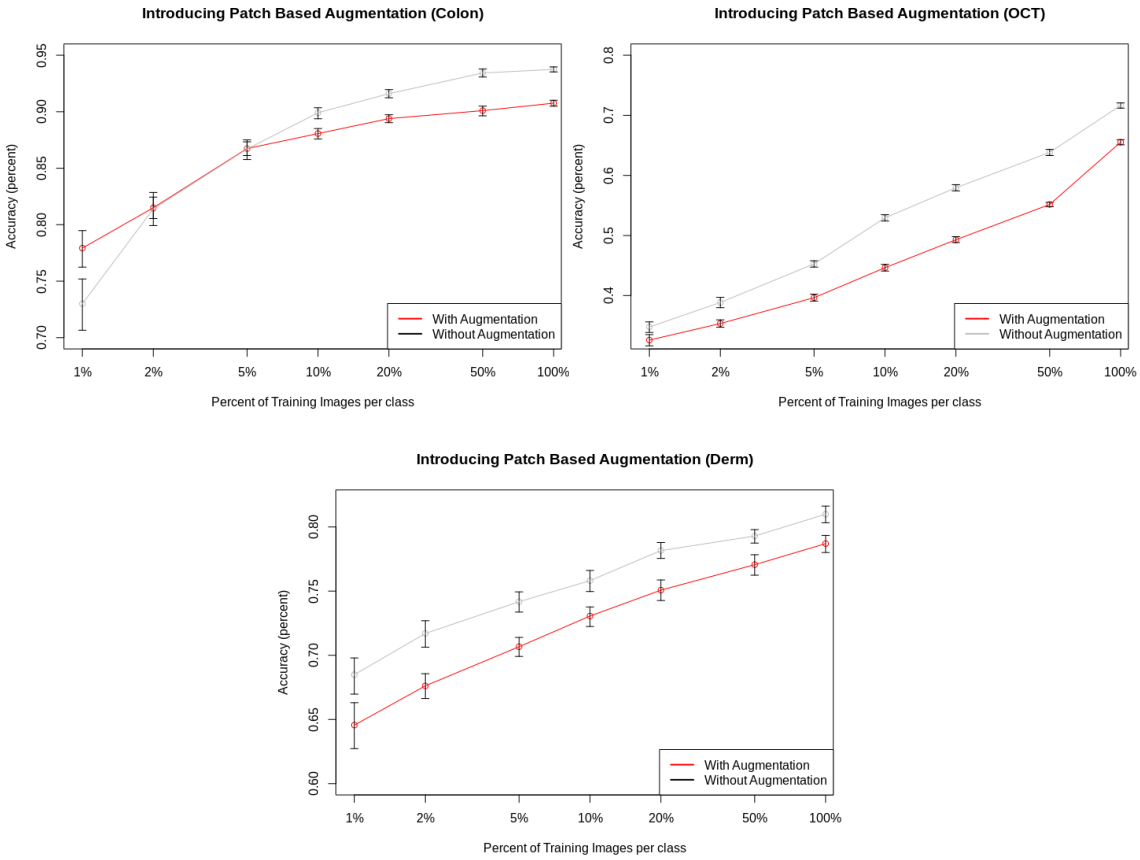

Figure 6.1: Impact of introducing patch-based augmentation to the CPC training protocol across three medical imaging tasks: Colonoscopy, OCT, Dermatology. Non overlapping bars show significance.

6.4 Composition of Augmentations for SimCLR on Medical Images

Chen et al [1] report that the optimal set of augmentations for use with the SimCLR protocol are: random crop, colour distortions, and random Gaussian blur. Medical images are likely to have a different set of requirements for augmentation than object-centric datasets such as ImageNet. In this experiment, an evaluation of possible

augmentation compositions is conducted, based on [1]. This section extends the work completed in [143].

6.4.1 Experimental Design

In this experiment, an evaluation of the augmentation strategy used in the SimCLR protocol is presented. Across two medical datasets, the combined performance of various augmentations is investigated, the augmentations chosen are: {random crop, colour distortion, noise, rotation }. For each pair of augmentations five SimCLR encoders (ResNet-11s) are trained, then the utility of this encoder is evaluated by adding a linear layer and freezing the encoder, the classification from this set up will be known as the linear predictive accuracy. This linear layer is then optimised using a labelled dataset outlined in the dataset section below. The SimCLR networks are trained for 100 epochs using a learning rate of 0.001, a temperature parameter of 0.1 and a batch size of 128. Due to the GPU memory requirements of having large batch sizes, this is less than was used in [1]. This change is discussed at length in the limitations of compute section in chapter 8. The network is implemented in Tensorflow [144] and trained on a single GPU. Fine-tuning of the model is not undertaken, to ensure that any result is purely the product of the SimCLR methodology.

Datasets: The first dataset used in this experiment is the HAM10000 dataset [12] first seen in chapter 4. The full dataset of 10k images is used to train the SimCLR encoder. For the labelled dataset, the same random subset used in chapter 4 is used, re-sampled to give a 50:50 training:testing split. In addition to the dermatology dataset, the OCT dataset also used in chapter 4 is also reused here. To train the SimCLR encoder, 10k images are randomly selected from the full OCT dataset. A different random selection is used for each of the five repeats of this experiment. As with the dermatology dataset, the same labelled subset is reused from chapter 4, re-sampled to a 50:50 training:testing split. No additional insight would be gained from the addition of the third medical imaging datasets found in chapter 4, and therefore to save on compute, this has not been included.

	Rotation	Translation	Colour	Noise	Average
Rotation	0.7947	0.8503	0.7523	0.8028	0.8
Translation	0.8504	0.8512	0.7563	0.8533	0.8278
Colour	0.7687	0.7525	0.7519	0.7512	0.7561
Noise	0.8056	0.8507	0.7584	0.7795	0.7985

Table 6.1: Linear predictive accuracy on the dermatology dataset of various compositions of augmentations.

6.4.2 Results

Tables 6.1 and 6.2 show the mean performance of a linear layer trained on encoders with various combinations of augmentations on dermatology and OCT data respectively. Contrary to the work of Chen [1], colour distortion was not found to lead to the greatest performance. On the dermatology dataset, it actually lead to a decrease in performance when applied with another augmentation, compared with just applying the other augmentation by itself.

However, the most noteworthy point is how the augmentations together are not consistent between datasets. For example, on the dermatology dataset, the performance Translation + Rotation performs on par with both Translation + Noise, and Translation by itself. However, on the OCT dataset Translation + Rotation performs subpar compared to the other Translation combinations, actually harming performance.

	Rotation	Translation	Colour	Noise	Average
Rotation	0.6818	0.7008	0.6807	0.6792	0.6856
Translation	0.7008	0.7840	0.7886	0.7800	0.7633
Colour	0.6718	0.7726	0.5922	0.5461	0.6457
Noise	0.6927	0.7733	0.5556	0.6053	0.6567

Table 6.2: Linear predictive accuracy on the OCT dataset of various compositions of augmentations.

6.5 Limits of Augmentation

[1] claims that stronger augmentation leads to greater performance, however, they only provide empirical results for colour augmentations. This claim is hard to believe, as the stronger an augmentation is, the further from the distribution it is hoping to emulate it will become (for example in the case of noise, as the amount of noise increases to an unreasonable level, an image will just appear as noise, not representing a realistic depiction of what the image is meant to depict). If this augmentation strategy becomes too extreme, the performance of the network will degrade. However, the inverse is also true, adding augmentation when training generally improves performance. Therefore, if there is insufficient augmentation, the performance is likely to be non-optimal. In this experiment, the claims made by [1] are evaluated across the four sets of augmentation strategies used in 6.4. In this and the subsequent experiments from this chapter, non medical datasets were used. This was done for consistency between this chapter and chapter 7 and also for more broad appeal outside of the medical imaging community. This choice is discussed in more length in the limitations section of chapter 8.

6.5.1 Experimental Design

This section empirically evaluates the claims made by [1] that stronger augmentation leads to greater performance. For each of the augmentation strategies under evaluation (random crop, colour distortion, rotation, and additive noise), five encoders are trained at various augmentation magnitudes. The same training protocol

as section 6.4 is followed, with the only change being the selection and magnitude of augmentation. The magnitude of augmentation is varied between:

- **Random Crop:** Varied between a maximum of 90% and 10% of full image size in 10% increments.
- **Colour distortion:** 0.1 and 0.9 in 0.1 increments.
- **Rotation:** 0.2 and 1 * π radians maximum rotation angle in 0.2 increments.
- **Additive noise:** Additive Gaussian noise with a standard deviation of between 0.01 and 0.09 in increments of 0.01.

Dataset: The STL-10 dataset [145] is used due to its common usage within the semi-supervised literature, thus allowing easy comparison. For the unlabelled dataset, multiple random subsets of the ImageNet dataset are used to increase the generalisabilty of this results.

Model Training: These models are trained using the same training protocol as 6.4. The SimCLR networks are trained for 100 epochs (100 epochs is the default training time from [1]) using a learning rate of 0.001, a temperature parameter of 0.1 and a batch size of 128. During supervised training, the models are also trained for 100 epochs, a learning rate of 0.001 and a batch size of 128. As with most training in this thesis, this has been limited by the computational requirements of SimCLR; this is discussed in depth in the limitations section in chapter 8.

6.5.2 Results

Figure 6.2 shows the results of varying the augmentation across multiple values and multiple augmentations. The results found here concur with the work of Chen [1] for colour distortion, showing that stronger colour augmentation leads to higher downstream performance. Despite this, there is no increase in performance for either additive noise or for random rotation. And, most notably, the inverse is true for random crop: initially there is no change in performance as the augmentation level is increased, however, towards the higher levels of augmentation, the downstream

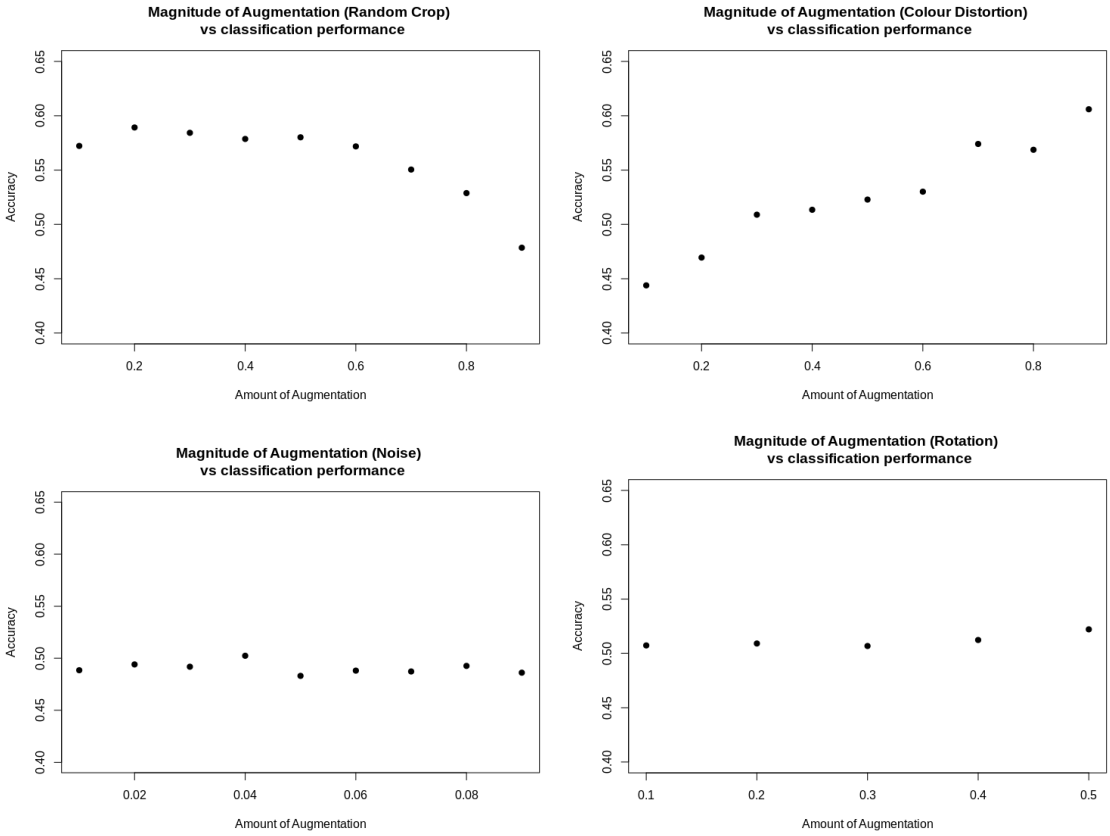

Figure 6.2: Network performance against augmentation amount for the four augmentations investigated in experiment 1, from left-to-right, top-to-bottom: Random Crop, Colour Distortion, Additive Noise, Random Rotation

performance drops off substantially. In addition to the result shown in figure 6.2 for the additive noise, a testing regime with much higher levels of noise can be found in appendix 1. This result also showed no increase in performance with greater augmentation.

6.6 Creating Invariant Representations

Some of the literature suggests that SimCLR produces good results by creating representations that are invariant to image augmentation [1]. This is only somewhat true, due to the inclusion of a non linear projection layer rather than applying the contrastive loss directly to the output of the encoder. A level of invariance could also be found in a network trained on a supervised task, due to the inclusion of augmentations. In this experiment, the exact level of invariance to augmentation is

quantified and compared to the 'invariance' found when using a traditional classifier.

6.6.1 Experimental Design

In this experiment, the invariance of the output of a SimCLR encoder (i.e a ResNet) is compared with the same encoder portion trained in a supervised manner using the same set of augmentations as the SimCLR protocol. To compare the 'invariance' of the two networks the following experiment is conducted: for each image in the dataset, 100 randomly augmented versions will be created in addition to the unaugmented version. The cosine similarity between the latent representations of the unaugmented image and each of the augmented images is calculated, thus producing a set of invariance scores for each image. The mean value of this set is stored for each image. This gives a distribution of mean invariance scores, for both the SimCLR encoder and the supervised encoder, which can then be compared.

Network Training: For both the SimCLR and supervised encoder, a ResNet-11 is used. Both sets of networks are trained on the training split of the STL-10 dataset, to ensure direct comparability. Both sets of networks are trained for 100 epochs with a batch size of 128. For the SimCLR training, a learning rate of 0.001, and a temperature parameter of 0.1 is used. For the supervised network, a learning rate of 0.001 is used. Both networks are implemented in Tensorflow [144] and trained on a single GPU. The same selection of augmentations are used during both training protocols which consist of {Random flip, random translation, random zoom, and colour distortions}.

6.6.2 Results

Figure 6.3 shows the two distributions of mean invariance to augmentation. SimCLR reports a higher invariance to augmentation than the supervised protocol with a mean invariance of 0.8170 compared with a mean invariance of 0.7430 ($P \leq$ 1e-4). This result validates the claim made by [1] that SimCLR produces invariance to augmentation. Despite this, a higher invariance does not *necessarily* lead to better

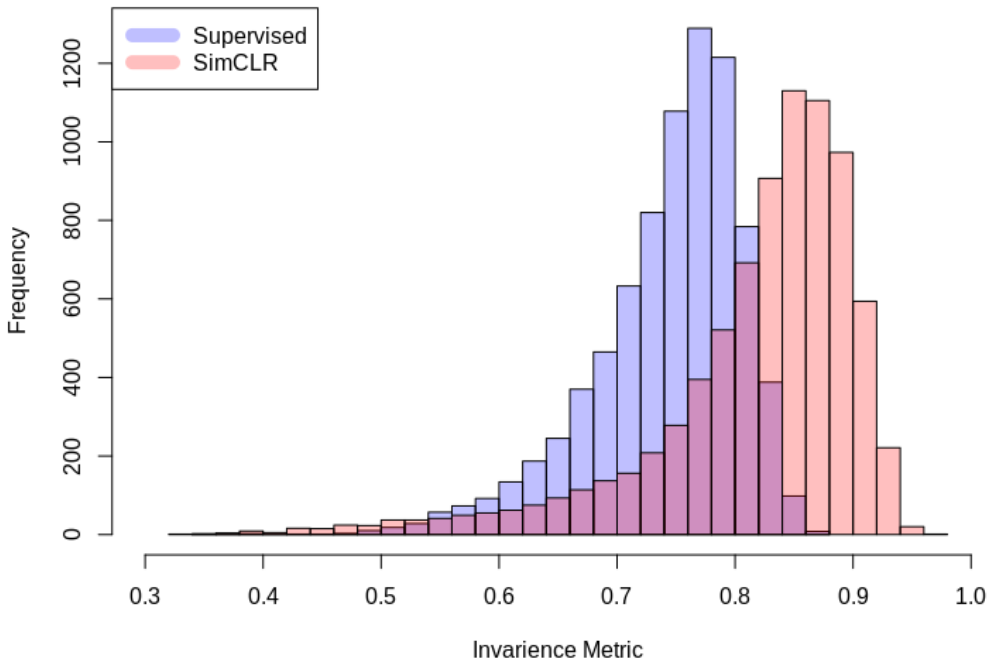

Figure 6.3: Invariance to augmentation, distribution of results from the experiment. Dark pink indicates overlapping distributions. SimCLR is more invariant to transform than the supervised baseline.

performance. In the next section, this result is further explored by examining the impact of this invariance on the use of augmentation during downstream classification.

6.7 Is supervised augmentation needed with SimCLR?

As stated previously, it is often said that SimCLR encodes information by learning to encode a dataset such that the embeddings of a datum is invariant to image augmentation. Under this assumption, data augmentation during supervised training would have limited or negligible impact on downstream classification performance. In this section, the work of the section 6.6 is expanded upon and given practical implications: is supervised augmentation during the downstream task needed to achieve good results?

6.7.1 Experimental Design

Ten SimCLR encoders are trained on subsets of the ImageNet dataset, using the same protocol described in section 6.4. Initially, these networks are frozen, a linear layer is added and, this linear layer is trained (for 100 epochs with the ADAM optimiser and using a learning rate of 0.001) with and without data augmentation in the supervised portion. This experimental setup is repeated, but with the networks not being frozen. This repeat was conducted due to the latter being a more realistic scenario for any implementer, therefore giving them more useful information. The results from the networks, both with and without the supervised augmentation are then statistically analysed to confirm whether there is a difference between the two sets of results. As with previous experiments, the STL-10 dataset is used for the supervised section and randomly chosen subsets of the ImageNet dataset for the unsupervised training section.

Statistical Tests: A t-test is performed between the two sets of linear layers to evaluate whether the two distributions are distinct. Because two measurements are made (frozen and non-frozen encoders), a Bonferroni corrected p-value of $\frac{0.05}{2} = 0.025$ is used.

6.7.2 Results

For the networks with the frozen encoder, the network trained with augmentation achieved a mean accuracy of 0.6874 compared with a performance of 0.6722 for the network trained without augmentation ($P < 1e\text{-}4$). For the networks where fine tuning was used, the network performance with augmentation was 0.7141 compared with a mean network performance of 0.6173 ($P = 0.01375$). Based upon these results, the conclusion can be made that augmentation during supervised training is not needed if using a frozen encoder network as there is no increase in performance from doing so, this is likely due to the encoded invariance to augmentation. However, when finetuning the network (as would be the case if this method was put into practice), augmentation of the supervised section increases performance. It is worth noting that it is likely that the non-frozen network without augmentation likely

overfit on the training dataset causing the low performance.

6.8 Discussion

This chapter has studied the impact of augmentation within the contrastive learning framework, investigating across two distinct methods, and through both the upstream and downstream tasks. Based on this work, the following novel results are presented:

- Patch-based augmentation was not found to increase performance on all three of the medical datasets studied in section 6.3. This disagrees with the work found in CPCv2 [2]. Hyperparameter tuning of the augmentation should be applied to achieve the greatest performance.
- The augmentations that gave the largest improvement on a medical imaging task (section 6.4), did not align with those proposed by [1]. In addition, there was differences in performance between datasets.
- In contrast to the work of Chen [1], section 6.5 finds that stronger augmentation does not always lead to higher performance, and in some cases, such as random crop, actually leads to lower performance.
- Section 6.6 examined the invariance to augmentation of SimCLR, finding that SimCLR was more invariant than a supervised network of the same type.

Selection and amount of augmentation: Sections 6.3 - 6.5 examined claims made about the type and amount of augmentation that is necessary to achieve good performance on a contrastive task. Initially starting with the claim that patch-based augmentation increased the downstream performance of the CPC protocol, section 6.3 finds that the performance is actually degraded by the inclusion of patch-based augmentation under that experimental set up. It should be noted that the augmentations used were not tuned. Based on this result, the augmentations used in SimCLR (a method that is far more reliant on augmentation than CPC) are evaluated. Section 6.4 finds that the the augmentations that had the greatest level of performance is inconsistent between the literature on ImageNet and the results

found on a medical imaging task. In addition, the performance between medical imaging tasks differed. To achieve the greatest possible performance, hyperparameter tuning of the augmentations should be conducted. It is beyond the scope of this chapter to examine *why* each dataset performed differently to different augmentations. However, one does not need to know *why* the network performs differently to be able to mitigate its effects through hyperparameter turning.

In addition to the type of augmentation, the amount of augmentation was also investigated (section 6.5); exploring a claim made by [1] and [142] that stronger augmentation produced greater downstream classification performance. I dispute this finding, with the work of section 6.5 showing that stronger augmentation did not in fact produce better results, and in one case actually harmed performance. Similar to the examination of augmentation types in Section 6.4, the significance of this work lies in its potential to be mitigated. The work suggests that the amount of augmentation should be hyperparameter tuned, thus increasing the chance of gaining the best possible performance.

Invariant Representations: Chen et al [1] also claimed that the network learns embeddings that are invariant to augmentation, and to do so, the network must learn high level features. Section 6.6 examines the level of invariance, finding that SimCLR produces more invariant embeddings than a supervised baseline. Section 6.7 extends this to examine whether this invariance leads to a network that does not require augmentation during the downstream task, finding that augmentation of the input data does not increase the linear separability of the embeddings. However, using augmentation during finetuning of the model does increase performance. This finding builds on the work of [1] by quantitatively validating the claim of embedding invariance, but showing that this invariance is not absolute: further gains can still be achieved from augmentation during supervised training.

Limitations: As with the work in chapter 4, this work is limited by the tradeoffs that were made to combat the computational cost of training contrastive networks. Throughout this chapter, a ResNet-11 was used as the network being trained for

SimCLR, rather than the ResNet-50 found in [1]. In addition, tradeoffs were made regarding image size and batch size: this chapter uses an image size of 96x96 compared to a typical 256x256 and a batch size of 128 compared to the baseline batch size of 256. It is likely that these changes had an impact on the results of this chapter and care must be taken when interpreting them. In chapter 8, I speculate on how these changes impact the results of this chapter and also give greater context on why these changes were made.

6.9 Conclusion

This chapter has investigated the effect of augmentation on the performance of contrastive learning methodologies, giving two overarching recommendations for gaining the best performance. 1) Hyperparameter tuning should be conducted on the augmentations used for training the encoder. This should be conducted on not only the type of augmentation, but also its magnitude. 2) SimCLR does produce more invariant embeddings to augmentation than a supervised baseline. Despite this, to gain the best performance, augmentation during the downstream task is still necessary.

Link to the aims: This thesis aims to investigate semi-supervised learning and to understand how best to apply this set of methods to obtain the highest performance. A large component of many contrastive learning approaches is their inclusion of heavy augmentation; this is especially true with SimCLR where the network learns to match embeddings of stochastic transforms of a single image. This chapter has investigated the impact of the type and amount of augmentation on the downstream performance of the SimCLR protocol. The chapter finds that there are no general rules for either the size or amount of augmentation, despite the suggestion of Chen et al [1]. I therefore suggest that the approach to get the best performance from SimCLR is to treat the type and magnitude of augmentation as a hyperparameter to be tuned. Additionally, this chapter investigates the encoder's invariance to augmentation and its impact on performance. Importantly, this section finds that data augmentation during supervised finetuning still increases performance, and therefore is a vital step in the full deployment process.

Chapter 7

Understanding Datasets for Contrastive Learning

Abstract

Dataset design can have a large impact on network performance, however, there have been limited efforts to quantify this. This chapter attempts to close this gap in the literature by examining a number of dataset characteristics which should have an impact on downstream classification performance and assess the effect of changing them in a resource constrained setting. This is accomplished through two related sections: firstly, an examination of how dataset size impacts the downstream performance of a network; secondly, an examination into how dataset composition affects downstream performance. Initially, this work shows that increasing the number of images in a dataset while keeping the number of epochs constant size leads to greater performance. An ablation study is performed that finds that most of the performance increase comes from the increase in the number of iterations rather than increasing the number of unique images. This section also shows that when training on large datasets, it is likely that previous work has underfit, however, this issue can be mitigated through the addition of an early stopping mechanism. The usage of early stopping within unsupervised pre-training is a technically novel contribution of this thesis. In the second section, a number of experiments are conducted that find that there is no statistically significant correlation between the 'overlap' of the

labelled and unlabelled dataset and the downstream performance. This is consistent across metrics and experiments.

7.1 Introduction

Design choice for a machine learning solution makes the difference between a system that cannot learn and a system that produces state of the art results. The vast majority of the literature focuses on design of the network, however, this overlooks a very important feature to optimise: the training dataset. When starting to solve a problem, resources are not unlimited, therefore, one has to choose how best to allocate these resources to produce the optimal result. This chapter contains two separate, but related, questions which will study how one can optimise the dataset used in a deep learning problem to increase the chance of gaining a high performing network.

Sections 7.3 - 7.6 of this chapter examines the impact of dataset size and how this affects downstream classification performance. In sections 7.8 - 7.11 two hypotheses of dataset latent variables that may affect the performance of a model are tested: the semantic variation of a dataset and the distribution overlap between a self-supervised task and the downstream task. This work is intended to guide design choice for the creation of new datasets: most freely available datasets cannot be used for commercial purposes and thus, new datasets must be created. This work informs the cost-benefit trade off of collecting more unlabelled data to train on.

7.2 Background

Study of the impact of datasets on network performance is often overlooked with research often simply examining whether the features learned on one dataset will transfer well over to a new dataset, for example, my own work in chapter 4. This often takes the form of: “will ImageNet features transfer to another dataset?”. In this chapter, a more in-depth study is conducted across two macro features that are

believed to have an impact on downstream performance: dataset size and dataset content.

Dataset Size: [3] studied the use of extremely large, uncurated datasets for unsupervised pre-training of very large models. Their applicable findings were that: despite being uncurated, and therefore having no guarantee of distribution overlap between the downstream task and pre-training task, the authors showed that the network could perform just as well as pre-training on ImageNet. They also showed that when keeping the number of iterations the same and increasing the number of unique images, the model performance plateaued after a certain amount, and that any increase in the number of unique images did not lead to an increase in performance. This second finding is consistent with the findings of [4], in which a higher resolution study was conducted, finding that there is no improvement in downstream performance when using more than 5% of ImageNet for pretraining and limited improvement when using more than 1%. However, there are also studies that refute this finding: [5] found that downstream performance kept increasing as the dataset size increased, this continued up to their maximum size of ≈100M images. This inconsistency highlights that further study is needed to be able to understand the variable that affects the downstream performance of a network when trained on an unlabelled dataset.

Dataset Content: A number of possible latent characteristics of the dataset used for pre-training of the encoder network have been suggested as being important for downstream performance, most notably that the distribution of the pretext dataset should match the distribution for the downstream task. [146] reports that "the SSL techniques [they] studied all suffered when the unlabelled data came from different classes than the labelled data". However, [97] argues that SSL features are much more resilient to class imbalance than supervised learning because the features learned by the network are not biased by class label. In situations where there is a large batch size, there is limited incentive for learning features that will be beneficial for both common (and uncommon) classes. The authors posit that SSL does not have this issue, and instead features that are common between classes are highly

prized. As with the literature on dataset size, consensus has not been reached and should be further examined.

7.3 Does changing the size of the unlabelled dataset alter linear classification accuracy of SimCLR?

Some prior work, e.g [3], has suggested that increasing the size of the unlabelled dataset used in semi-supervised learning increases performance. In this section, these claims are externally validated on new datasets and the semi-supervised method used in this section.

7.3.1 Experimental Design

This experiment aims to validate the claims that increasing the size of the unlabelled dataset will increase linear classification performance. To do this, a number of SimCLR encoders (ResNet-11s) are trained on varying sized datasets and their linear classification performance is reported. Five repeats are conducted.

Datasets: There are two datasets types needed to explore this problem: a size varying unlabelled dataset; and a static labelled dataset. To generate the unlabelled datasets, random subsets of the ImageNet-1M dataset are used of size X where $X \in \{100, 200, 500, 1k, 2k, 5k, 10k, 20k, 50k, 100k\}$. The images are centre cropped and downsampled to be 96x96 pixels to ensure they are the same size as the labelled dataset. For the static labelled dataset,the STL-10 dataset is used. The STL-10 dataset was specifically designed for unsupervised learning and consists of 13000 images across 10 classes, split into a 5k:8k train:test split. The full dataset is used for evaluation.

Training the networks: The SimCLR networks are trained for 100 epochs on each of the variously sized datasets outlined above. A learning rate of 0.001, a temperature parameter of 0.1 and a batch size of 128 is used. The network is implemented in Tensorflow [144] and trained on a single GPU. To evaluate the

performance of the SimCLR embeddings, a linear layer is added to the output of the network. The linear layer is then trained for 100 epochs on the frozen SimCLR embeddings. A learning rate of 0.001 is used. As with the work in the previous chapter, fine-tuning of the model is not undertaken, to ensure that any result is purely the product of the SimCLR encoder training methodology.

7.3.2 Results

The linear classification performance of an encoder trained on varying sized subsets of the dataset can be found in figure 7.1. As the number of training images increases, so does the performance. This result concurs with the result of [5]. This result is further studied in an ablation study in section 7.4.

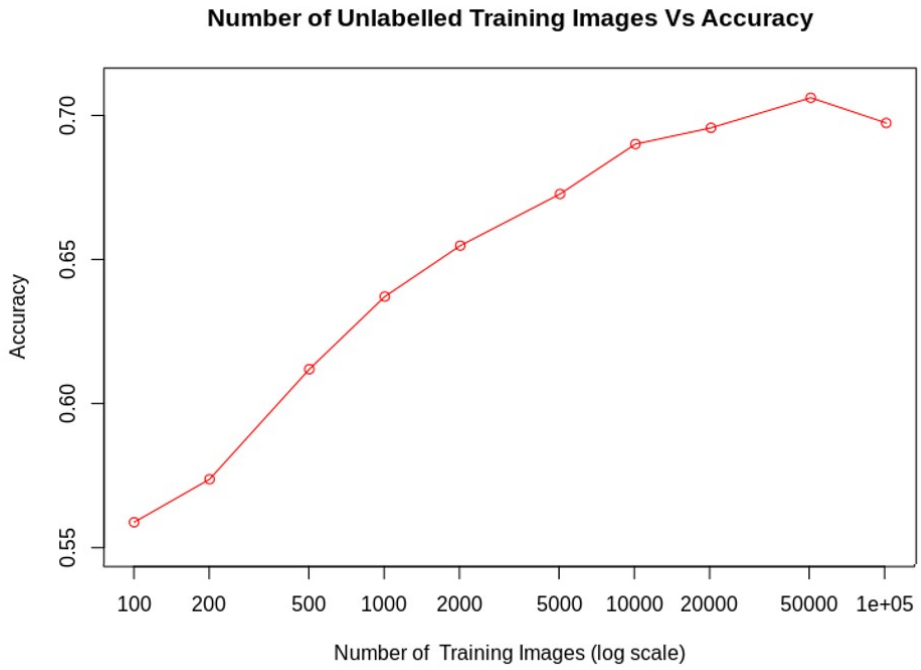

Figure 7.1: Size of unlabelled dataset used for encoder pre-training vs the linear classification performance of the STL-10 dataset using that encoder. The x-axis is approximately logarithmically spaced between 100 and 100000 images.

7.4 Ablation of increasing dataset size

When increasing the size of the unlabelled dataset as in the prior section, two metrics are increased: **(a)** the number of unique images are increased, and **(b)** the number of iterations are increased, where iterations is the total number of training batches the network has. Here, the experimental setup of [3] is followed to distinguish between the two effects.

7.4.1 Experimental Design

In this section, two experiments are conducted; in each one, changing only one of the two latent metrics. In experiment one, the number of images will be kept static, while the number of iterations changed; and in experiment two, the number of iterations will remain static, while the number of unique images will be varied.

Training the network: The same training protocol found in section 7.3 is followed with the following modifications: for our sub-experiment one, the number of images will be kept static at 1k with the number of iterations varying approximately logarithmically spaced. The runs equivalent to 100, 200, 500 images are excluded from this experiment due to the number of iterations being less than the number of total images. For sub-experiment two, the number of iterations is kept static at 100K. This is equivalent to 1k epochs at 100 unique images and 1 epoch at 100k images. The dataset descriptions can be found in 7.3.1.

7.4.2 Results

Figure 7.2 displays the results from this section. The image on the left, shows that as the number of iterations increases, the linear classification performance also increases. In the image on the right, an unexpected result can be seen: if the number of iterations are kept constant at 100k, the performance of the network stays static at approximately 63.5%.

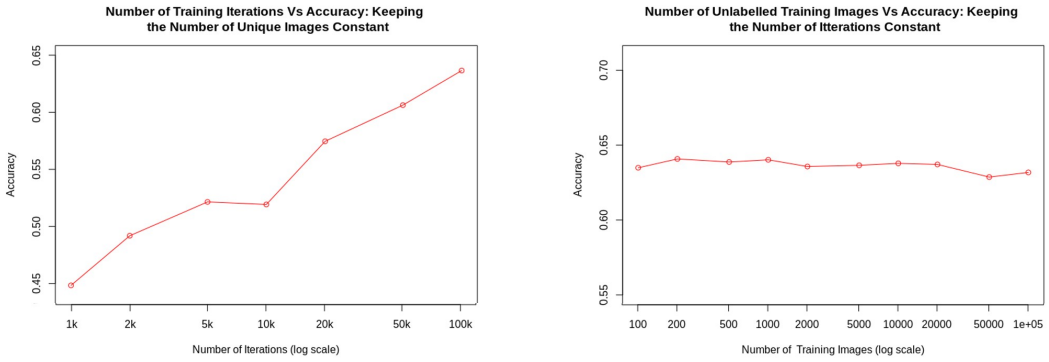

Figure 7.2: **(Left)** Number of iterations for encoder pre-training vs the linear classification performance of the STL-10 dataset using that encoder keeping the number of images static at 1k. **(Right)** Unique images in the unlabelled dataset used for encoder pre-training vs the linear classification performance of the STL-10 dataset using that encoder

7.5 Further Exploration of Number of Unique Images

Section 7.4, showed that when the number of iterations was kept constant, the performance of the network remained static. Since this result was unexpected based on the literature, in this section, further exploration of this result is conducted. In the previous section, the number of training iterations is kept constant at 100k. To ensure that this result is more robust, this experiment is repeated across a number of different amounts of constant iterations.

7.5.1 Experimental Design

The same experimental design as section 7.4 is followed, keeping the number of iterations constant. A set of networks are trained using N images for X iterations where $N \in \{100, 200, 500, 1000, 2000, 5000, 10000, 20000, 50000, 100000\}$ and $X \in \{100\text{k}, 200\text{k}, 500\text{k}\}$. The networks are trained using the same training protocol as 7.4, just varying the total number of iterations. The results are then plotted, separating the number of iterations by colour. The dataset descriptions can be found in 7.3.1.

7.5.2 Results

Figure 7.3 shows that the network has underfit to the larger number of unique images in the previous experiment. As the number of iterations the network is trained for is increased, the network is able to achieve higher performance before plateauing. While the very low end of images are starting to show signs of overfitting (100 images achieves lower average performance when increasing above 100k iterations, and 200 images has started to show a drop in performance when above 500k iterations), no signs of overfitting for ≥ 500 images can be seen. I postulate that higher performance can be achieved for all subsets greater than or equal to 500 images by increasing the number of iterations trained for. This result further reinforces the result found in the previous section: increasing the number of unique images in a dataset does not inherently increase the performance of the network, it must be accompanied with an increase in the number of iterations.

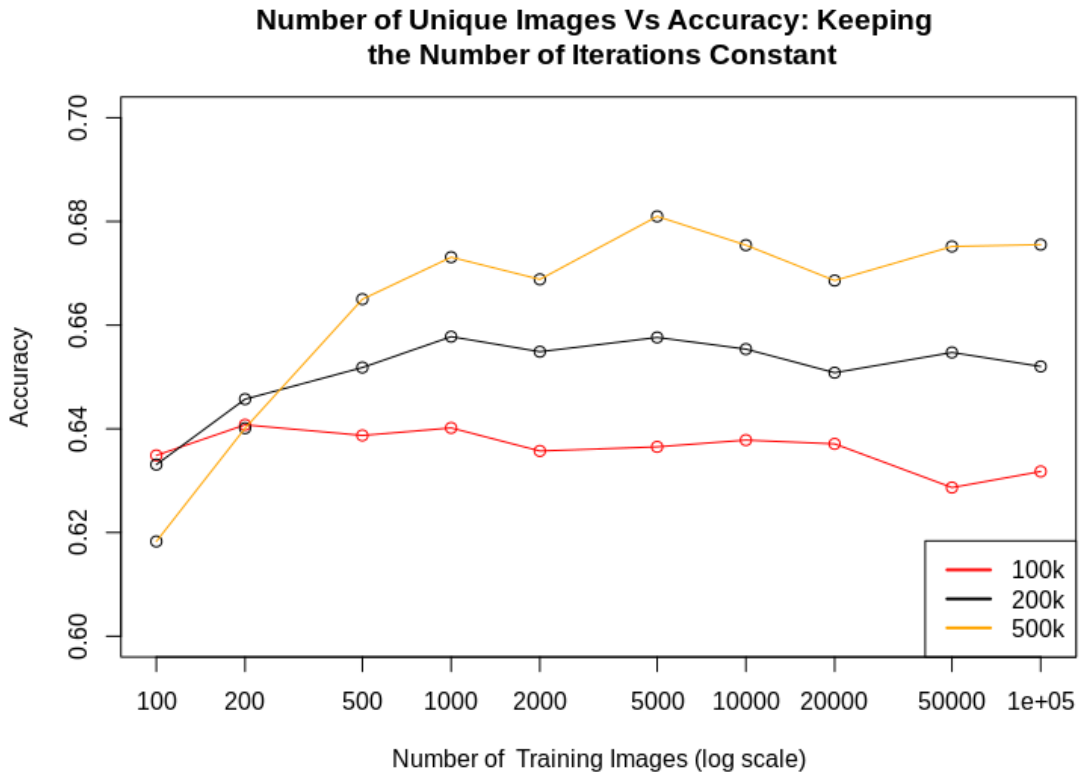

Figure 7.3: Impact of increasing the number of unique samples in the unlabelled dataset while keeping the number of iterations constant at 100k (black), 200k (red), and 500k (yellow).

7.6 Overfitting and Underfitting

Overfitting of a model to a set of data is seen as a major problem within deep learning. Recent advances in GPU technology in addition to Application-specific integrated circuits for AI (such as Google's TPU [147] and NVIDIAs Tensor Core [148]) has led to models that can reach over half a trillion parameters [149], thus exacerbating the issue. Despite this, section 7.5 shows that model underfitting can remain an issue and suggests that models should be trained for longer. When training models for a period of time, there are three possible outcomes: 1) the model has underfit, performance could be gained by increasing the number of epochs; 2) the model has overfit, performance could be gained through decreasing the number of epochs; and 3) the model has plateaued, an insufficiently powerful model has been used and therefore, performance could be gained through increasing the size of the model. In this experimental section, the work of 7.4 (Left) is extended to examine how the network's performance changes as the number of iterations is increased beyond the level seen in that work.

7.6.1 Experimental Design

A number of models are trained on unlabelled datasets consisting of {100, 1k, 10k} unique images. The networks are trained using various numbers of iterations *in* {1k, 2k, 5k, 10k, 20k, 50k, 100k, 200k, 500k, 1m, 2m, 5m, 10m }. The same implementation details as the previous experiments are followed, merely changing training length. Definitions of training protocol and datasets can be found in 7.3.1.

7.6.2 Results

Figure 7.4 shows that as the amount of unique images in the dataset increases, the number of iterations needed to reach peak fitting increases too. For the ResNet11, the peak performance occurs at {50k, 1M, 2M} iterations for {100, 1k, and 10k} unique images respectively.

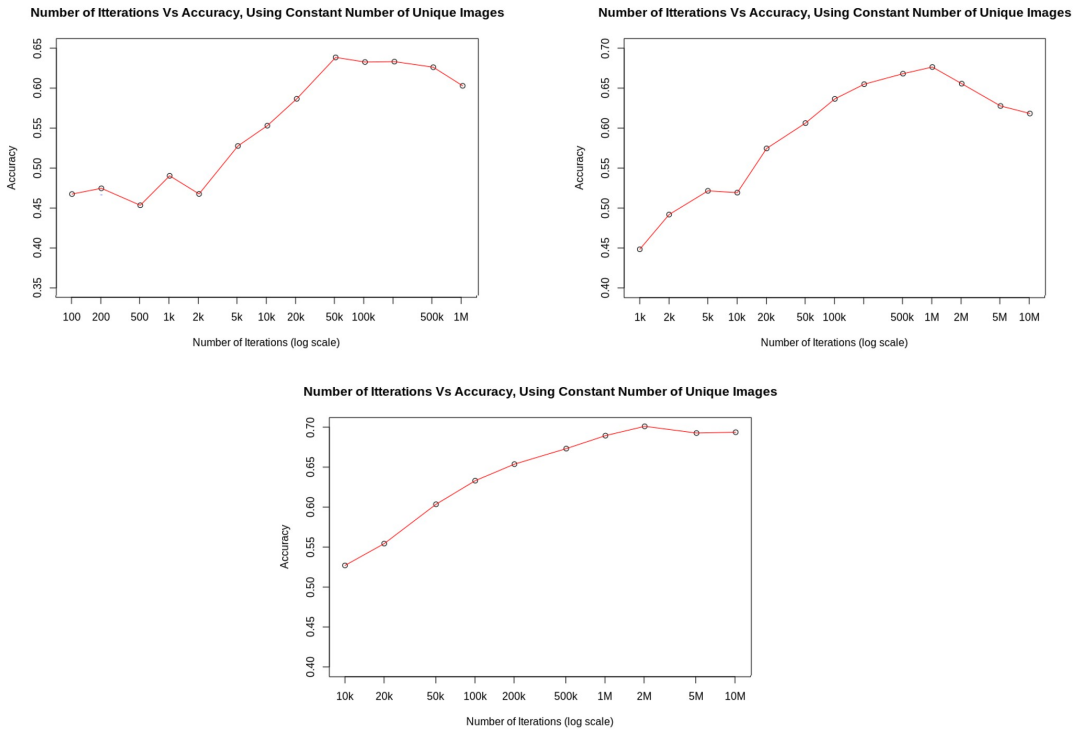

Figure 7.4: Examination of under-fitting vs over-fitting. How performance varies with number of iterations across two model capacities and across three amounts of unique images: **(top-left)** 100 unique images; **(top-right)** 1000 unique images; **(bottom)** 10000 unique images.

7.7 Mitigating the impact of non-optimal fitting

Section 7.6 has shown that choosing an arbitrary number of iterations will most likely lead to a non-optimal result, either due to under-fitting (section 7.5) or over-fitting (section 7.6). In this section, a common method for combating this found in the supervised literature is evaluated: early stopping [37]. The stopping criteria "stop on training convergence"[1] is evaluated in this section. This differs from the early stopping used in chapter 4 as no validation set is used, training is stopped when the training loss stops decreasing. While the usage of early stopping to stop overfitting is common within supervised learning environments, to the best of my knowledge, this is the first time that this has been applied to unsupervised learning. This therefore provides a technically novel solution to the issues raised in the previous sections.

7.7.1 Experimental Design

A set of models are trained using various amounts of unique images, however, rather than using a static number of iterations, an early stopping paradigm is used with a maximum number of iterations set at a computational budget of 10M with a patience of 50 epochs. As with the previous experiments, five sets of models are trained for each unique image amount. This is likely too few at the low end, and too great a number at the high number of unique images, but for comparability, this is kept constant. All models are trained using the protocol as the previous experiments: a batch size of 128 and a learning rate of 0.001 for both unsupervised and supervised training. Unlike the previous experiments, training length is controlled by early stopping during the unsupervised training. Supervised training is for 100 epochs.

7.7.2 Results

Figure 7.5 shows the performance of a network trained on varyingly sized datasets, using the early stopping paradigm proposed. This performance is compared to the estimated peak performance using the results in section 7.6. Using the early stopping paradigm closely matches the estimated peak performance, albeit, slightly below.

[1] The training is stopped when the training loss does not decrease for a set patience.

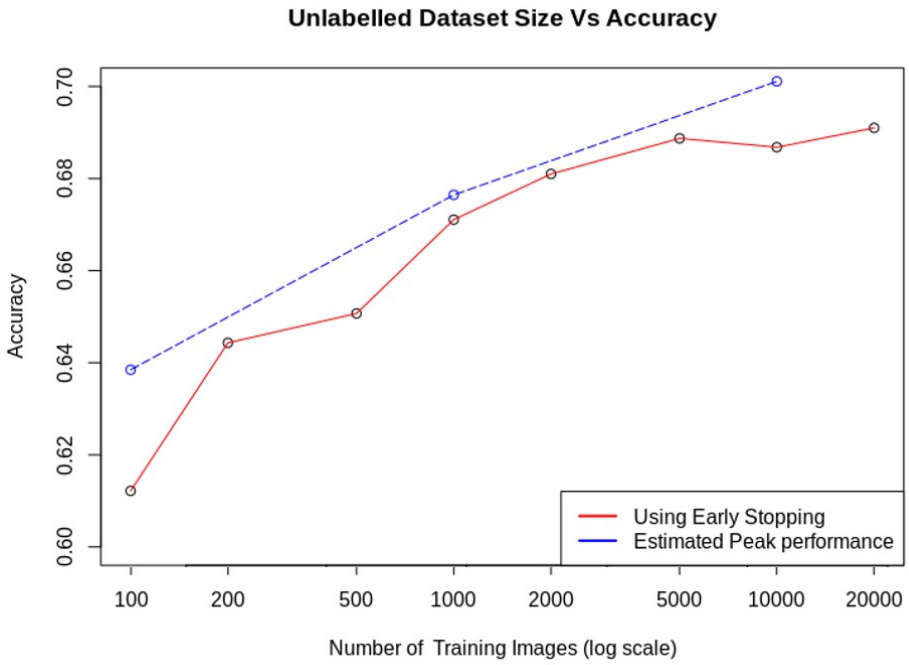

Figure 7.5: Linear classification performance using encoders trained using early stopping on various amounts of unlabelled data. **Red:** uses training convergence as the stopping parameter; **(Blue:)** the estimated peak performance with optimal stopping.

At the larger data amounts, the performance starts to diverge, this is due to the model running out of computational budget. Also, at the lower end, the model underfit, but for a different reason than at the high end: here, the constant number of patience equates to a low number of iterations compared to the higher amounts, thus performance could have been gained by training for longer. Due to the low number of training examples, the training results are more noisy than the larger sized sets. The model is able to achieve this approximate peak performance for the following reason: underfitting is prevented by training until the model would likely overfit, but this is prevented from happening by early stopping.

7.8 Does increasing the semantic variation within the general imaging dataset produce more useful features for a downstream task?

Increasing the size of an unlabelled dataset used for semi-supervised learning improves accuracy [150] (when also increasing the number of iterations, see section 7.5). When dataset size is increased, this increases both number of instances, but also the variability of the dataset. [146] suggests that performance degrades when the distributions differ between the labelled and unlabelled dataset. The hypothesis of this section is that increasing the variability of the dataset will increase the chance that a feature that is useful for the secondary task will be learned. To the best of my knowledge there has been no work examining this. When new unlabelled datasets are created, this work will inform how much emphasis (if any at all) should be placed on variability, over purely size.

7.8.1 Experimental Design

This section evaluates the effect that dataset variation of an unlabelled dataset used with an unsupervised model has on the classification performance of a downstream task. Broadly, this section will take the form of: training multiple encoders (ResNet-11s) at varying amounts of 'variability' of the dataset, ensuring that dataset size is kept constant. The performance of these networks will then be evaluated on a downstream task. Statistical analysis will evaluate whether there is a correlation between the 'variability' and downstream performance. This experiment is repeated with 1k and 10k images used for training.

Proxy of dataset variability: Calculating the variability of a dataset is complex. For this section, a proxy of dataset variability will be used, based upon the semantic information in a dataset. The ImageNet dataset [23] is a large, natural image dataset consisting of over 14 million images [151], however, most commonly the term is used to refer to a subset of this dataset consisting of approximately 1 million images across 1000 non overlapping classes, for the ILSVRC 2012 com-

petition. This section proposes using an increasing number of classes as a proxy of increasing the variability of the dataset. Specifically, unsupervised models are trained on subsets of the ImageNet dataset, X, which contain C number of classes where, $C \in \{20, 30, 40, 50, 60, ..., 880, 890\}$ and $size(X) \approx 10000$. The classes to form the subsets are chosen randomly. This assumption is corroborated with [152].

Training the SimCLR network: The SimCLR networks are trained for 100 epochs on each of the proxy datasets outlined. The SimCLR network learns from approx[2] 10k 96x96 images. A learning rate of 0.001, a temperature parameter of 0.1 and a batch size of 128 is used. The network is implemented in Tensorflow [144] and trained on a single GPU. This experiment is then repeated using 1k images.

Downstream task evaluation: The STL-10 [145] dataset is used as the imaging evaluation dataset. A linear layer with no fine-tuning will be used when evaluating the performance on a supervised dataset, to ensure that only the embeddings learned from the SimCLR method are evaluated. The linear layer is then trained for 100 epochs.

Statistical Tests

H_0: There is no correlation between the variables.

H_1: There is a correlation between the variables.

A statistical test is needed to evaluate whether there is a correlation between the amount of variation in a dataset and the classification performance of a SimCLR network trained on that dataset. There are a number of statistical tests that could be used to evaluate this:

- **Pearson correlation coefficient:** A common method for measuring the correlation of two variables, however, it can only be used for linear relationships. As there is no expected relationship between the two variables, this method would not be appropriate.

[2]The number of images used for the unlabelled dataset is equal to $round(\frac{10000}{NumberOfClasses}) * NumberOfClasses$ then oversampled to ensure that 10000 images are used.

- **Spearman's rank correlation coefficient:** This method is able to be used for non linear relationships due to calculating the rank correlation.

- **Distance correlation:** The distance correlation can be used as a metric for calculating the probability of dependence in a similar way to the bootstrapping method outlined in chapter two. In this, the distance metric is calculated on the data, the data is then shuffled and the value recalculated a number of times. The first value is then compared to this distribution to find whether the level of significance has been reached. This metric does work with non-linear dependencies, but is a relatively uncommon approach.

For the statistical test in this section, Spearman's rank has been chosen due to its common usage and ability to be used on non-linear relationships.

7.8.2 Results

In this experiment, 88 SimCLR models were trained on datasets with varying number of classes included in them as a proxy for dataset variability; a linear layer was then trained on the learned embeddings. In figure 7.6 the accuracy is plotted vs the number of classes included in the dataset. There is no clear trend between the two variables: A Spearman's rank test is performed as set out in the statistical test section, with a calculated p-value of 0.3964, leading to acceptance of the null hypothesis that there is no correlation between the number of semantic classes used in the unlabelled dataset and the downstream classification performance. The repeated result using 1k images shows the same lack of correlation.

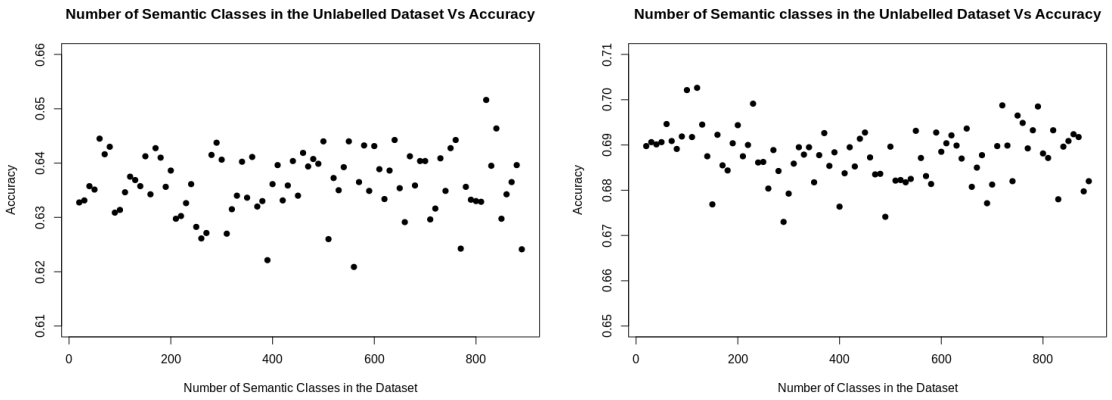

Figure 7.6: A plot of the performance of a SimCLR network trained on different amounts of 'variation' in the unlabelled dataset. The number of semantic classes is varied between 20 and 890 classes, in increments of 10 classes. **(Left)** Shows the variation in performance when using unlabelled 1k training images; **(Right)** shows the variation when using 10k images.

7.9 Does Increasing the Overlap of Semantic Classes Between Labelled and Unlabelled Datasets Increase Performance?

Based on the results of section 7.8, it could be hypothesised that the semantic content does not have an impact on the quality of the representations learned. This hypothesis would contradict previous argument that the distribution of the unlabelled and labelled images should be similar: under the assumption that the same semantic class will have a more similar distribution than a dissimilar semantic class. This experiment compares the performance of networks trained on varying levels of semantic overlap between the unlabelled and labelled dataset.

7.9.1 Experimental Design

Using the same training protocol as the previous experiments, 100 SimCLR encoders are trained using unlabelled datasets consisting of varying amounts of semantic overlap between the unlabelled dataset and labelled dataset. For each of the 10 possible amounts of overlap, 10 repeats are conducted. The encoders are then evaluated by

optimising a linear layer on the output of the encoder, reporting the linear separability of the test dataset after 100 epochs. The same training protocol as 7.8 is used.

Dataset Generation: 10 sets of datasets are generated with varying amount of semantic overlap between the labelled and unlabelled datasets [153]. The amount of overlap is varied between 10% (i.e the unlabelled dataset contains only one semantic class) and 100% (i.e the unlabelled dataset consists of all of the semantic classes). The labelled dataset used is the STL-10 dataset which consists of images of aeroplanes, birds, cars, cats, deer, dogs, horses, monkeys, ships and trucks. To generate these unlabelled datasets, the ImageNet dataset is used, combining multiple ImageNet classes to form the same semantic classes as would be found in the STL-10 dataset. Images are randomly selected from the total population of possible images to generate an unlabelled training set of 1000 images.

7.9.2 Results

Figure 7.7 shows the effect of increasing the amount of semantic overlap between the unlabelled and labelled dataset on the downstream classification performance. There is no trend in the data: performance is static across the amount of overlap. This result is surprising, and disagrees with some conjectures made in the literature, however, this result agrees with the work of [97] that argues that self supervised models optimise for more generic features than class specific features.

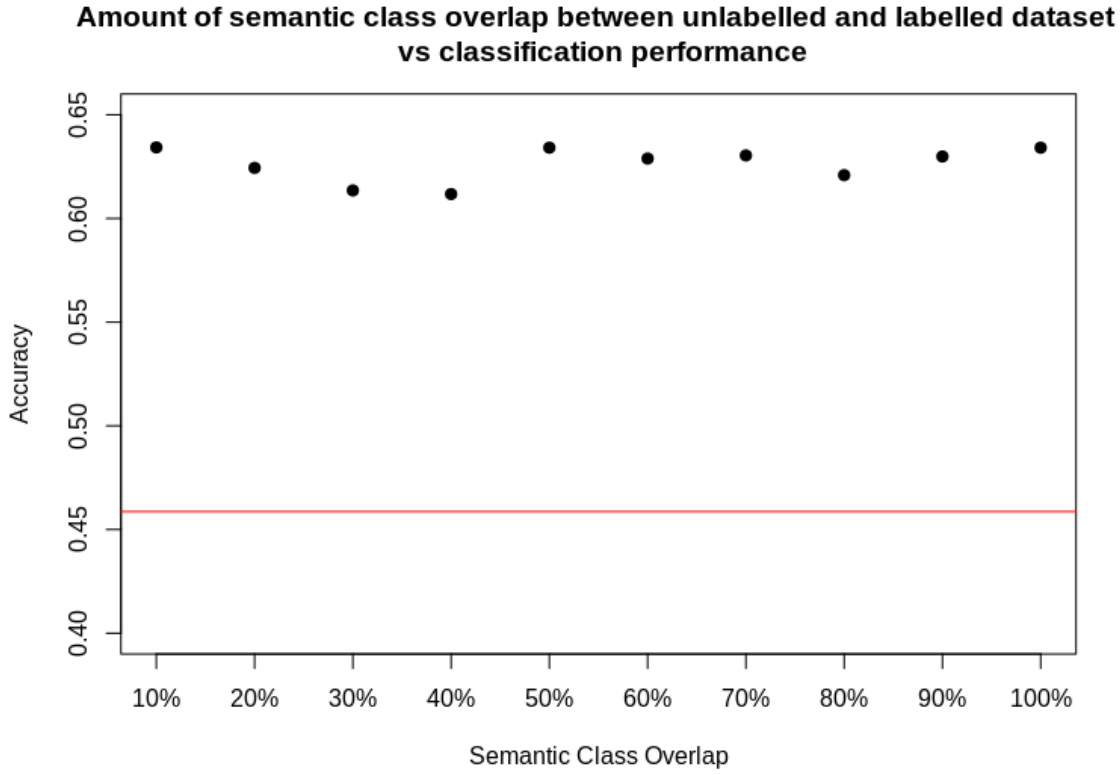

Figure 7.7: Impact of increasing the amount of semantic overlap between the unlabelled dataset and labelled datasets. Varied linearly between 10% and 100%. Red line indicates an untrained encoder.

7.10 Does SimCLR Learn General or Data Specific Features?

In this section, an evaluation is conducted to examine whether the features learned by the SimCLR mechanism are specific to the unlabelled dataset trained on, or they are general image features that have utility for any task. This work is especially useful for areas in which is it often very hard to collect domain specific datasets. If it were to be shown that SimCLR learns general features, very large models could be trained on extremely large general imaging models and these features could then be used for arbitrary tasks, reducing the cost burden of applying deep learning even further. Based on the results of section 7.9, it is expected for there to be no difference between the two sets of networks. This experiment extends 7.9 from synthetic tests to real data tests.

7.10.1 Experimental Design

This section examines whether general or specific features are learned by the SimCLR method. Based on the assumption described below, an experiment is designed to test this.

Assumption: This section works under the assumption that if a network learns dataset specific features, then the network pre-trained on the same dataset as used for supervised training will perform better than the network pre-trained on an arbitrary dataset. This assumption underpins this section.

Datasets: In this experiment there are two datasets: the test dataset consisting of a medical imaging task, and an arbitrary dataset consisting of images that do not overlap with the test dataset. For both sets of data, the images are centre cropped and re-sampled to be 96x96 pixels. For the test dataset, a subset of the HAM10000 [12] is used, utilising 3000 images split into 3 classes: Benign lesions of the keratosis (a non cancerous lesion); Melanoma; and Melanocytic nevi (pigmented moles [117]). To train the encoder networks, the full HAM10000 dataset is used

which consists of ≈ 10k images of various skin lesions. For the arbitrary dataset, 10 sets of 10k image randomly sampled subsets of the full ImageNet are used. ImageNet was chosen due to its large size, ubiquitous usage, and lack of distribution overlap between itself and the test dataset. This experiment is then repeated using 1k sized subsets of the HAM10000 and ImageNet datasets.

Network Training: Firstly, a SimCLR encoder (ResNet-11) is trained on one of the unlabelled datasets following the implementation details set out in 7.8. The encoder is frozen, and add a linear layer is added as before, optimising the weights to linearly separate the three classes from the test dataset.

Statistical Analysis: For each dataset, 10 SimCLR encoders are trained, and evaluated on the dataset described above. A t-test is then applied to test for significance. As two repeats are conducted, the Bonferroni corrected p-value is 0.025.

7.10.2 Results

When tested on the 10k full dataset size, the SimCLR networks trained on the dermatology datasets achieved a mean classification performance of 89.2% compared with a mean classification performance of 87.5% for the networks trained on ImageNet (p=0.04184, non-significant). When trained on the 1k sized subset of the full dataset, the network trained on the dermatology images achieved a mean classification performance of 78.6% compared with 84.5% for the networks trained on ImageNet (p<1e-4). In neither case did the networks trained on the dermatology dataset achieved statistically significantly greater performance than the network trained on a general imaging task. This result holds great importance for the utility of SimCLR and is discussed further in the discussion.

7.11 Does CPC Learn General or Data Specific Features?

Semi-supervised models hope to decrease the cost of applying deep learning by using cheaper, unlabelled data to learn features that can be used to increase performance of some related task. While cheaper than labelled data, unlabelled data still has some cost of acquisition, this is in addition to the training cost of the model itself. If it can be shown that the features learned by semi-supervised methods are generic features and transfer over to multiple tasks, a large general model could be trained and reused for multiple tasks. This would help the uptake of these methods by reducing a barrier to entry.

Experimental Design: This section uses the same model training protocol as chapter 4. Encoders are trained on the four unlabelled datasets: {Colonoscopy, Chest x-ray, OCT, dermatology}. Then the following experimental design is followed:

- Select one of the labelled datasets, eg colonoscopy.
- Using the encoder that was trained on the unlabelled datasets that matches the labelled dataset: train a set of ResNets with that encoder, employing the same methodology as chapter 4. This set of results will be will be termed the "matching results for dataset x".
- Using one of the encoders trained on unlabelled datasets that do not match the labelled dataset. Train a set of ResNets using the same methodology as above. This result is be termed "non matching results 1 for dataset x". Repeat for "non matching results 2 for dataset x" and "non matching results 3 for dataset x"
- Compare the results for the matching and non-matching datasets.
- Repeat for the other three datasets.

Datasets: In addition to the datasets used in chapter 4, this experiment also uses the ChestX-ray8 dataset [154]. 11,400 x-rays are randomly sampled from three classes of images consisting of: Atelectasis (collapsed lung); Infiltration (an abnormal substance that accumulates gradually within cells or body tissues [155]). Plus a set of images with no pathology detected. The full ChestX-ray8 dataset is used for CPC encoder training.

Network Training: This section uses the same CPC training protocol as chapter 4. The CPC encoder (ResNet-50) is trained for 60k iterations, with a batch size of 16, and optimised using the ADAM optimiser [104] and a learning rate of 2e-4. The encoder training is always 60k iterations: in all datasets except the dermatology dataset, this corresponds to a single pass through the data. As the dermatology dataset is only 10k images, this corresponds to 6 epochs of the dataset. As with the work in chapter 4, the classifier (ResNet-11) are trained using the ADAM optimiser [104] with a learning rate of 5e-4, and early stopping [37] with a patience of 50, up to a maximum of 1000 epochs.

Results: The set of four images in Figure 7.8 contains a graphical representation of 1280 models trained on CPC embeddings. The orange lines represent the the networks trained on embeddings learned on the data these models are trained on (i.e the encoder would be trained on CXR images then a linear layer would be trained to predict CXR pathologies). The networks trained on the embeddings of the specific dataset appears to offer no benefit over using a network trained on a completely different dataset. This concurs with the work of [97] that the features learned by semi-supervised learning are more general than the class specific feaures that can be found in supervised learning.

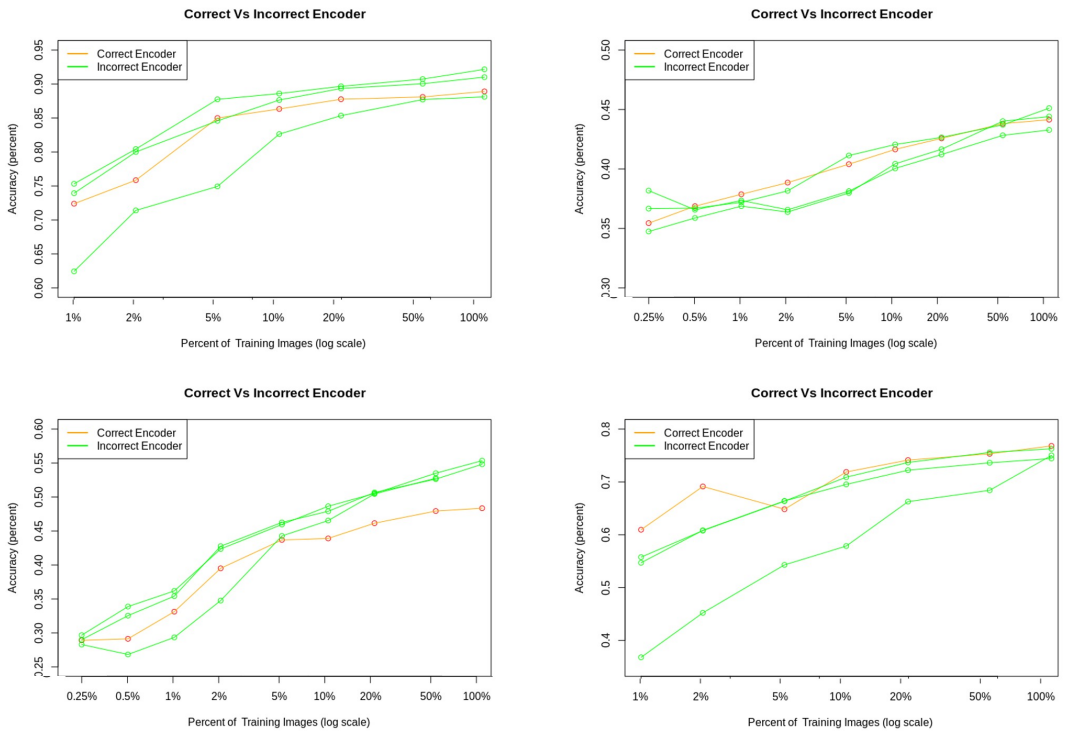

Figure 7.8: Here, exploration of whether the CPC encoder learns dataset specific features or general image features. The green lines show performance when the encoder is trained on a different dataset to the labelled dataset, and the orange line is when it is the same. If CPC learned specific features the expectation would be to see the orange line on top for all datasets. The graphs show colon, CXR, OCT and dermatology left-to-right, top-to-bottom respectively.

7.12 Discussion

This chapter is composed of two macro sections: firstly, sections 7.3 - 7.7 examines the impact of unlabelled dataset size on downstream classification performance; secondly sections 7.8 - 7.11 examine how the content of the dataset (specifically the semantic variability and distribution overlap) affects downstream performance. Overall, the following recommendations are made that contribute to the literature:

- With a fixed computational budget, increasing the number of unique image instances will not increase performance, as long as the dataset is sufficiently large to overcome the initial increase in performance.

- Both under-fitting and over-fitting are problems in contrastive learning, which need to be accounted for in unsupervised training. The technically novel suggestion of early stopping of the unsupervised section is shown to be a mitigation for this. The early stopping does not stop underfitting, it provides a stop point so that an implementer can train for longer without overfitting.

- There is no correlation between the semantic variability of the unlabelled dataset and the downstream linear classification performance of the embeddings. It is therefore unnecessary to spend effort to optimise for this.

- There is no trend between the amount of semantic overlap between the labelled and unlabelled datasets. In addition, no statistically significant correlation between any of the metrics for measuring distribution overlap between the unlabelled and labelled datasets and the downstream classification accuracy was found. Therefore, increasing the overlap between the unlabelled and labelled classes is not necessary for semi-supervised learning.

- SimCLR trained on a dataset taken from the same distribution as the labelled dataset did not produce better results than a dataset trained on ImageNet, leading me to conclude that the features learned by SimCLR are general and are not dataset specific.

Dataset Size: The first macro section of this work examines how dataset size impacts the downstream classification performance of semi-supervised networks. The work complements and extends existing research, however, disagrees with the conclusions made with the prior work. Figure 7.9 shows results given in [4] which show the performance of a network trained using a static number of iterations but varying amounts of data. The authors claim the plateau is evidence that after a certain amount of data, no performance gains are seen. This work extends this, presenting evidence that this plateau is due to underfitting at the larger data ranges, rather than that no performance is gained from larger datasets.

The finding that for a given computational budget, performance is not gained from larger datasets is surprising. I posit that this will have real world impact where

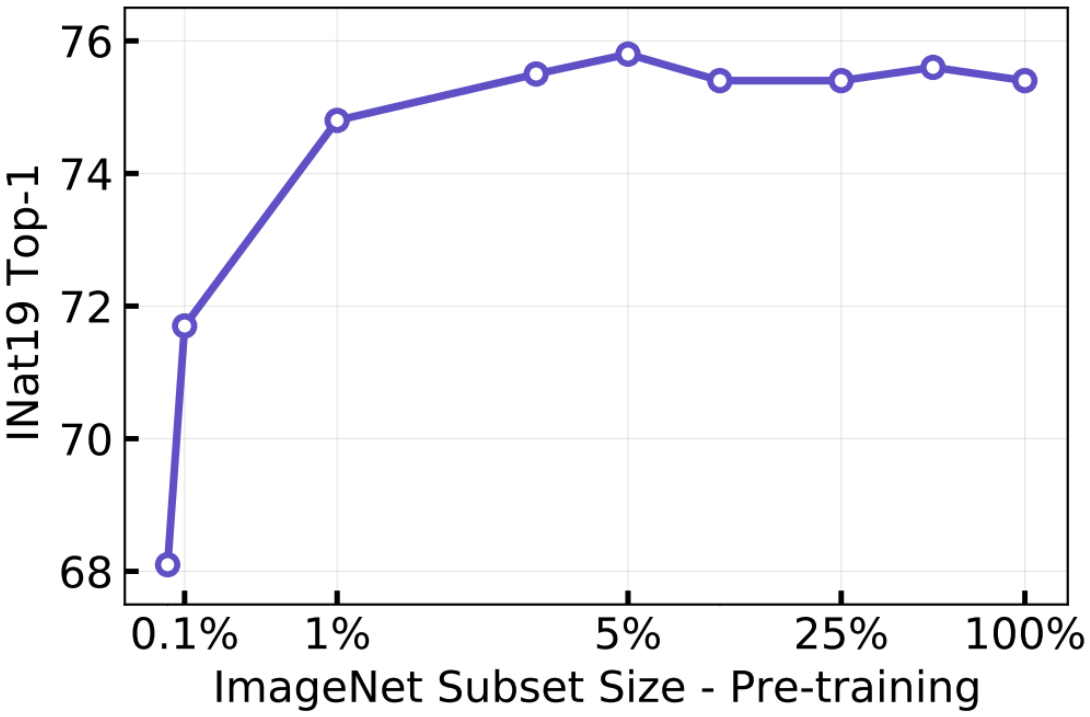

Figure 7.9: ImageNet top-1 performance when training with differently sized unlabelled datasets, taken from [4]

collection of additional data is not cost free. In a setting in which one is collecting their own data, rather than relying on open datasets (ImageNet cannot be used for commercial purposes: the ImageNet project does not own any of the copyright for the images that are within the dataset, therefore, the images cannot be used outside of Fair Use), collecting data has some cost to it. This cost is not uniform, for example the cost to scrape unlabelled images from the web is relatively cheap which leads to very large datasets being able to be collected. On the other hand, collecting images such as medical data can be far more expensive due to ethical and legal restrictions. If the implementer has a fixed computational budget and the cost of generating more data is costly, it may make no difference to use additional unique images. While beyond the scope of this work, it would be interesting to examine at what point this plateau happens. However, it is possible that such an examination is intractable due to the large number of factors that can affect such an outcome, eg: model capacity, augmentation strategy, number of unique images, image size, etc.

The work in section 7.6 shows that as the number of unique images increases, the number of iterations needed to learn from it also increases. Therefore, if an implementer is attempting to increase performance by increasing the size of the unlabelled dataset, it is imperative that they have sufficient computational budget to be able to not underfit the dataset.

Dataset Composition: The second macro section of this chapter examines how the composition of the unlabelled dataset affects downstream performance of a network. This is studied through three dimensions: 1) semantic variation; 2) distribution overlap between the unlabelled dataset and labelled dataset; 3) whether the features learned or general or domain specific.

The hypothesis for the semantic variation experiment (section 7.8) is that as the number of semantic classes used in the unlabelled dataset increases, the chance that a feature useful for the downstream task is learned increases, thus leading to higher downstream performance. This was found not to be the case, and there was no significant correlation between the number of semantic classes used and the downstream classification performance. This is contrary to the work of [152], which showed that, under a supervised transfer learning paradigm, increasing the number of semantic classes in the pretext task increases performance in the downstream task.

Another surprising result is the work in section 7.9. No statistically significant correlation was found between the distribution overlap between the unlabelled dataset and the labelled dataset, and the downstream performance. Based on this result, the performance of a SimCLR trained on a dataset taken from the same distribution as the labelled training set was compared to a network trained on ImageNet and found that there was no increase in performance from the networks trained with distribution overlap. As argued in section 7.9.2, this agrees with the work of [97] that says that self supervised methods learn more general features than supervised learning tasks, and therefore do not need as much overlap as supervised tasks. Additionally, the work of section 7.11 concurs with this work, in that experiment, the importance of distribution overlap between the unlabelled and labelled datasets on

downstream performance was evaluated. As with the work of this chapter, it found no link between the overlap and absolute classification performance. This adds to the argument that distribution overlap between the two datasets is far less important than would be for supervised learning. Additionally, this result is extremely important for the utility of SimCLR. By showing that no performance gain is achieved from retraining the SimCLR encoder for each specific task: encoders are able to be reused between tasks. This saves cost on two fronts: the cost of acquiring the labelled dataset, and the cost of training the encoder.

Limitations: Consistent with the work throughout this thesis, this chapter has made tradeoffs due to the computational cost of training contrastive models. This chapter has made the same tradeoffs as chapter 6 with regards to the the smaller image size and batch size than [1]. These choices are discussed at length in the limitations section of chapter 8.

7.13 Conclusion

This chapter has conducted a number of experiments examining how the size and content of the unlabelled dataset used for unsupervised pre-training impacts the SimCLR methodology. This chapter finds that increasing the size of an unlabelled dataset does not inherently increase performance. Additionally, underfitting remains a problem, and this chapter argues that improvements on the state of the art could be achieved through longer training. This chapter also provides a technically novel solution for how an implementor should conduct unsupervised training. The second macro section evaluates how the content of the images affect the downstream classification performance: no correlation between semantic variation and performance was seen. The degree of semantic overlap between the labelled and unlabelled dataset was not found to affect downstream performance. Evidence is also presented to argue that the distribution overlap between the unlabelled and labelled datasets is less important than under a supervised paradigm.

Link to the aims: This thesis aims to investigate semi-supervised learning and to understand how best to apply this set of methods to obtain the highest performance. Chapter 4 investigated how the size of the labelled dataset impacts the downstream classification performance, however, this only investigates a tiny proportion of the total amount of data used: the unlabelled dataset makes up a much larger proportion of the total training data. Chapter 7 investigates how design choices about the unlabelled dataset affect the downstream performance of a SimCLR model. This chapter presents a novel result in which increasing the quantity of unlabelled images used by a semi-supervised model does not result in greater downstream performance, unless that increase in data is also given an increase in compute power (in the form of additional iterations). This important result improves the current literature, which did not provide enough compute capacity for the large datasets these papers used. This chapter also presents work that shows that the contents of the images used is less important than the quantity of them. This provides meaningful guidance to practitioners when designing their own datasets for use with a semi-supervised model.

Chapter 8

Discussion

The acquisition of labelled data is extremely challenging in some fields, such as medical imaging, with even the most simple of labelling tasks costing vast sums of money to create the ImageNet sized datasets that are needed to apply some of the state of the art approaches that are found in the literature. This high cost of acquisition means that some of the state of the art methods that can be found in the literature are difficult to apply well to these fields. In **chapter 1**, I presented the problem case and discussed how it is often easier to acquire unlabelled data than it is to acquire labelled data. This cost differential can be many orders of magnitudes. This causes some unlabelled datasets to be hundreds of millions of images in size, whereas labelled datasets are often in the thousands range. This cost differential can be felt more acutely in fields such as medical imaging, where the people tasked with labelling the data are highly skilled, and therefore costly, physicians.

Contrastive learning (introduced in **chapter 2**) has been found as one example of a method that could possibly leverage the power of cheap, unlabelled datasets to increase the performance of a network on a downstream task. **Chapter 3** introduced Contrastive Predictive Coding (CPC) as a possible such method. **Chapter 4** showed that CPC could possibly increase the performance of a ResNet in an extremely data limited scenario across two different datasets. However, it also highlighted that this training protocol is not a panacea: the training protocol used does not work equally well across datasets and sometimes cannot beat training directly from the pixels.

From these promising results, **chapter 5** introduced SimCLR. This method has been extensively studied in the literature, therefore, no evaluation of the performance of the method has been conducted as, unlike for Contrastive Predictive Coding, no value would have been gained from this study. Instead, gaps in the literature were identified, focusing on design choices of the training protocol. These were evaluated to examine how to achieve the greatest performance: 1) **chapter 6** investigated how the augmentation protocol affects the downstream classification performance 2) **chapter 7** investigated how the unlabelled dataset affects downstream classification performance.

Chapter 6 found that hyperparameter tuning is necessary to achieve the greatest performance across datasets, and that the conclusions made regarding the performance on ImageNet may not transfer over to alternative datasets. It also validated claims that SimCLR representations are invariant to augmentation, however, finding that augmentation during supervised training is still necessary to achieve the best performance. **Chapter 7** then concluded that SimCLR implementations found in the literature use a training period that is not long enough, this leads to the conclusion that these networks have underfit on the unlabelled dataset. In addition to the experiments on the length of training, **chapter 7** also examined how the content of the unlabelled dataset affects performance. It concluded that the distribution overlap between the unlabelled and labelled datasets are less important than in supervised approaches.

Finally, **chapter 8** looks at how this work fits within the scientific literature and gives suggestions on possible further directions of research.

8.1 Limitations

This thesis presents a number of results based on experimentation into whether contrastive learning can be used for improving the performance of machine learning models when given large amounts of unlabelled data but small amounts of labelled

data. Despite the importance of the results presented throughout this thesis, a number of limitations of the work exist and are summarised here for the reader.

8.1.1 Limitations of Compute

This work has been limited by the computational cost of training contrastive models. Semi supervised methods use extremely large amounts of compute to train these models, for example [3] used 512 GPUs concurrently. This is far greater than the computational capacity available to a PhD student: For reference, I had a computational budget of approximately £1000 for 3 years of the PhD (this is approximately 13 hours of use of a GPU server with just 8 top of the line graphics cards [156]). Due to this limitation on computational capacity, a number of trade offs were made to reduce the computational demand of running the experiments in this thesis.

Chapters 6 and 7 used a ResNet-11 as their backbone model: This model is much smaller than the ResNet-50 used in the SimCLR paper [1], which has approximately an order of magnitude more parameters than the ResNet-11. Reducing the number of parameters in the model reduced the absolute power of the network, [2] found that larger networks performed better than smaller networks with their experimental set up. Despite this limitation, reducing the number of parameters in each model, it allows for a larger number of experiments to be run, increasing the quality of the work in this thesis. Firstly, this reduction in training time allowed for repeat training of models to be conducted, increasing the reproducibility of this work. Prior work [2] [1] report the results from a single training run allowing for possible erroneous or statistically unlikely results to be taken as final results. By conducting repeat runs, with different random initialisations and different random subsets of the dataset, the chance that the results reported in this thesis are a statistical anomaly are greatly reduced. In addition to the increase in repeats in training of these results, the reduction in number of parameters meant that the model could be trained for longer. An example of this limit can be seen in section 7.6 where a subset of unlabelled dataset sizes were used for calculating the peak training amount (100, 1000 and 10000 images). If more computational capacity was available, an

estimated peak value could have been calculated for each unlabelled dataset size. Despite this, I believe that the work presented here is a fair compromise given these limitations. For example, while a subset of unlabelled datasets was studied in 7.6, I believe that it is unlikely that increasing the number of subsets would have changed the conclusion that I came to and therefore is not a large limitation. Similarly, while purely conjecture, I believe that while the absolute levels of performance may differ between network sizes, the trends found in this work would still be relevant for larger network sizes.

In addition to the limits on the amount of compute I had available, there was a limit on the amount of GPU memory available. Larger input images, larger model sizes, and larger batch sizes all contribute to higher memory usage. ResNet-11s were chosen due to the computational cost of increasing model size, therefore a trade-off is left between image size and batch size. As the size of the images increased, fewer images could be used within a single batch during training due to an increase in GPU memory usage. This meant that both for the work on Contrastive Predictive Coding and SimCLR, the experiments in this thesis used smaller batch sizes than presented in the original papers. [1] shows that increasing the batch size of a contrastive learning task leads to greater performance (increasing the batch size increases the number of negative examples, which makes the pretext task harder. By making the task harder, the network has to learn better features). Based on this result, it is likely that higher absolute performance could have been obtained by increasing the batch sizes for both network types. In addition to the batch size, I also reduced the size of the images (in chapters 6 and 7) from a typical 256x256 to 96x96. 96x96 was chosen as this was the size of the images in the STL dataset chosen for the supervised dataset of some of the experiments. Chapter 4 uses an image size of 256x256, a similar image size to the one used in the Contrastive Predictive Coding paper, so this limitation does not apply here. I believe that I used a fair compromise between image size and batch size. Additionally, I believe that while the absolute performance numbers would change with a larger batch size and image size, it is likely that the trends given in this thesis would hold. This is, however, purely speculation as I could not complete the necessary experimentation to confirm

this.

Finally, in addition to the size of the models being a limitation, this thesis also only examined a ResNet backbone in both the work on Contrastive Predictive Coding and SimCLR. A ResNet backbone was used in this work to follow the same model design as the SimCLR and CPC papers, however, there is no intrinsic reason for why they must be used. Both CPC and SimCLR can both use any model backbone that can project an image (or patches) down to a latent embedding space. More recent work has used transformer models as their backbone. While it is worth mentioning in the limitations that this was not studied, I believe that the results in this PhD would be relevant for any neural backbone.

While I highlight these as limitations of this thesis and areas for possible further study, I have no reason to believe that the results presented in this work would not be relevant to an implementor working with more standard sized images. All these changes were made to increase the value of the work in spite of the resource constrained environment that I operated, for example to allow repeat measurements to be conducted. While I do not believe that these results would change for larger batch sizes, images, and networks, this is just speculation. Further work is needed to confirm this speculation.

8.1.2 Use of Non-Medical Datasets

While chapters 6 and 7 do include some work conduced on medical datasets (sections 6.3, 6.4 and 7.10), the majority of this work is conducted on general imaging datasets (STL-10 and ImageNet subsets). These general imaging datasets were chosen to increase the appeal of this work to readers outside of the medical imaging domain, however, this comes at the expense of a loss of applicability to the medical imaging community. Without replication of the results in this thesis to medical imaging datasets, I cannot say with certainty whether these results will translate. This was a conscious choice, as I felt that, while my PhD was on medical imaging, these results would be more applicable to people working on general imaging

datasets.

This choice helped facilitate the work of experiment 7.8. Experiment 7.8 investigated whether increasing the semantic variability of the unlabelled dataset would increase the downstream classification performance of a network. The hypothesis of this experiment was that the larger the semantic variability, the increase the chance that the network would learn a feature that is relevant to the downstream classification task. To test this hypothesis, a dataset was needed that had a large number of classes, each of which could vary quite dramatically. For this reason, subsets of the ImageNet dataset were chosen. Firstly, the ImageNet dataset has a large number of distinct classes (1000 classes), this number of classes in a single dataset is uncommon in medical imaging dataset, especially with a large number of images per class (1400 images per class on average, however, this does vary). Secondly, the variability of the images found in the ImageNet dataset is larger than would be found in many medical imaging datasets. Consider the semantic difference (and under the hypothesis, the features that this would create) between an image of a dog and an image of a car, compared with the image difference between 2 x-rays, one with TB and one with cancer. Finally, ImageNet is a much more common dataset (the de facto standard for computer vision), and given this was a novel hypothesis, I believe that this increased the usefulness of the experiment.

While one could be critical of this decision within a PhD in medical imaging, I believe that this was the correct choice and increased the value of the work presented here. Despite this, I cannot conclusively say that the results presented in this thesis will transfer over to a medical imaging task.

8.1.3 Unanswered Question

There exists an open question within the machine learning community as to whether we should focus on models that can be applied to any scenario or whether application specific models are needed to achieve the highest level of performance.

On the one hand, there is the advent of foundational models (introduced in chapter 2). The high-level idea behind these models is that very large (multibillion parameter) models are trained on extremely large, internet scale, datasets to produce a model that can give good levels of performance across any task. On the other hand, these models are still not perfect, performance can be surpassed on some tasks by fine-tuning these models on more specific datasets and/or more specific tasks. Across the commercial sector, it is common to finetune foundational models such as llama-3 7B. This is done for two main reasons: 1) in a number of cases, finetuned smaller models can outperform untuned larger models; 2) even in cases where these models cannot outperform their larger counterparts, the increase in performance gained by finetuning them reduces the gap enough that the reduction in cost is worth this slight reduction in performance.

This thesis has not settled the debate on whether task specific, fine-tuned models should be produced, or work should be conducted to produce large, generalist models. However, a number of relevant questions were addressed in this work:

- The work in section 7.10 and section 7.11 attempted to answer whether training contrastive methods on datasets that closely align with the downstream tasks produce better results than training on another arbitrary dataset. The literature in this area has not found a definitive answer which is why this was studied. The results from 7.10 and 7.11 say that the performance on models trained on a dataset that aligns with the downstream task do not perform significantly better than models trained on datasets that do not align with the downstream task. This work aligns with the theory behind foundational models. Despite this, this work will obviously not answer this question fully, merely add to the body of evidence. Further work needs to be conducted on more datasets and across a larger variety of semi-supervised methods. It would be interesting to repeat the studies conducted in this thesis on some of the auxiliary task methods presented in chapter 2.

- The work in section 6.4 and 6.5 highlight that hyperparameter tuning to a specific dataset can increase the performance of SimCLR, at least in a resource

constrained environment. This could be argued that this presents the opposite conclusion from the work above: that dataset or task specific networks perform best. As above, this only provides some evidence and further work would be needed on a larger variety of datasets and tasks.

This work currently presents a mixed message: the results from chapter 6 argue that there are improvements to be gained by conducting hyperparameter tuning the augmentations used in simCLR training to each specific task or dataset. This can be argued to be the complete opposite argument given in chapter 7: that an implementor should be spending resources to create a single, task agnostic, generalist model using as much data and training resources as possible. Further work will be required to untangle these contradictory results. Section 8.2 provides a discussion of this mixed message and possible experiments that could be conducted to disentangle this message.

8.2 Disentangling the Mixed Message

In the limitations section "Unanswered Questions", I highlight that there is currently a mixed message between advising to train task specific networks (i.e optimising the augmentations for a specific dataset), and training generalist networks (training models for longer on a large, general dataset). There is currently an interdependence between the results of the two sections that can provide possibly contradictory results. In this section, I highlight these results, provide possible interpretation, but most importantly, give a set of possible future experiments that could be conducted to help untangle the contradiction.

Chapter 6 provides evidence to suggest that the optimal set of augmentations will change depending on the dataset, and that the chosen augmentations can substantially change the performance (an over 20 percentage point difference in performance between the highest performing and lowest performing augmentation on the OCT dataset). The experiments in chapter 7 do not follow this recommendation and instead use the set of augmentations used in [1]. One can say that this is a flaw

in the experimental set up, and possibly led to sub-optimal results. However, this choice more positively means that the results found in this section are more comparable to the literature. Unfortunately, this choice has meant that, without further experimentation that has not been conducted in this thesis, there are contradictory results that are unable to be untangled.

Similarly, the work in the first half of chapter 7 has given evidence to suggest that a large portion of the literature has underfit on the datasets used during their unsupervised training. This has likely led to sub-optimal results. This result has an impact on the interpretation of the results found in chapter 6. When training SimCLR models for investigation of the impact of augmentation, chapter 6 followed the training regime of [1] in which the encoder model was trained for 100 epochs on the unsupervised task. This again has both the pros and cons of the previous paragraph: it provides a more comparable result to the literature, however, could have resulted in sub-optimal results.

8.2.1 Possible Further Experiments

An initial set of experiments that should be conducted would be repetition of the experiments using the larger batch size, model size and network size that were used in the original SimCLR paper. This set of experiments were not possible due to the resource constraints of the PhD discussed in the limitations section, however, an implementor with a larger resource budget could train these models. This would help exclude the possibility that the contradictory results seen in chapters 6 and 7 are due to training models in resource constrained environments. Throughout this discussion I have conjectured that these deviations in model sizes from [1] would not have had an impact on the trends seen in this thesis, merely changing the absolute performance numbers. I also conjecture that this will also be the case for this investigation. If, as I believe, training in a resource constrained environment did not explain these contradictory results, further experimentation would need to be conducted to untangle this.

A possible experiment that could be conducted to help give insight on the contradictory result would be an investigation into how the deviation of augmentation strategies changes as you increase the training amount. In this experiment, an implementor would, for different training amounts (i.e the number of training batches) train a selection of models on different augmentation strategies (i.e different selections of augmentations used during unsupervised pre-training). After training these models, the standard deviation of model performance between augmentation strategies could be calculated. The implementor could then see whether the standard deviation of performances across different augmentation strategies narrows as you increase the size of the datasets (and compute budget). If there was a narrowing in standard deviation, then this would invalidate the recommendation that hyperparameter optimisation should be conducted for finding the optimal augmentation strategy, and any future implementor should just focus on increased training time on a single augmentation strategy. Alternatively, if there was no narrowing in standard deviation, this would add to the argument presented in this work that augmentation should be optimised.

One underlying cause that could cause the large variance in performance between the different augmentation types could be that an (or multiple) augmentation fundamentally performs badly when used with SimCLR. To investigate this, one would need to replicate the experiments from section 6.4 (investigation of composition of augmentations) across a large number of datasets across many domains and compare whether one (or multiple) augmentation type(s) perform substantially worse than the others. This investigation would be a large investigation in itself. However, this could explain why there was a variance in performance of different augmentations.

Finally, it is important to note that the results of this thesis may not actually be contradictory. It is possible that both sets of results hold: 1) given a static set of augmentations, the way to get the best results is to train for longer. 2) given a set number of iterations that you can train a model for, hyperparameter tuning of the model should be conducted. Unfortunately, the body of this thesis does not contain enough experimentation to conclusively say one way or another. It is hoped that

conducting these experiments will help future investigators clarify the contradictory results found in this thesis.

8.3 Decreasing the Cost of Deep Learning

I started this thesis by asking the question of *why*: "why do we wish to use AI anyway?". A number of reasons were presented, including scarcity of healthcare workers and reducing the number of medical mistakes. Fundamentally though, these reasons come down to cost: why is there scarcity of healthcare workers? Because it costs a lot of money to train and employ doctors and other healthcare workers. Why are medical mistakes made? Because healthcare professionals are overworked to save money and it is uneconomical to have all the work checked by multiple people. AI is seen as a nostrum for this problem: an AI has a static cost, it cost roughly the same amount of money to create an AI to see one patient, or 1 million patients. AI can work in unison with both other AI, and human doctors, reducing the chance of a single person being able to make a catastrophic mistake. This does not mean that mistakes will not happen, AI is not infallible, but the chance of a mistake happening through negligence could be greatly reduced.

While possibly morally uncomfortable, we as a society have allocated specific value to human life. Healthcare economists assign a value to possible treatments on offer within a healthcare system to work out if they 'save enough life' to offset the financial cost to these treatments. In the UK, a common metric used for this is pounds per quality-adjusted life year [157] (£/QALY), that is, the cost of every additional year in perfect health that an intervention is likely to achieve. In the UK, this is typically less than £30,000/QALY [158] for a treatment to be considered economical, and therefore likely to be approved, it should be below this level. It is therefore necessary that any AI approach would need to be as cost effective as any other treatment.

As a whole, this thesis investigates reducing the cost of using state of the art deep learning methods to produce good results and thereby allowing us to use the resources we do have most efficiently. While it could be thought that these cost

limitations only apply to the smallest entities in the deep learning space, such as early stage start-ups and university researchers; even the largest companies in the deep learning field are resource constrained. For example, GPT-4 [159] - the state of the art natural language processing model - reportedly cost over $100 million to train [160], and therefore training could only happen once. This is due to the extraordinary cost of training these large, foundational models. This thesis has found the same to be the case in the imaging field, with **chapter 7** hypothesising that the work of Chen for SimCLR has actually underfit on their dataset.

The cost of creating an AI comes from many aspects, not just the cost to train these models: in addition this work has examined the impact of the size of datasets, both labelled and unlabelled. **Chapter 4** presented work examining CPC as a method to improve performance when one has very limited labelled data. This chapter could be considered to be an investigation into the case where the cost of creating labelled data is *vastly* more than the cost of unlabelled data.

Chapter 7 examined whether it is necessary to spend extra money to acquire additional unlabelled data: while usually cheaper than labelled data, it still has some cost. This cost will be application dependant, for example, web scraping images will be far cheaper than commissioning someone to take images for you. The chapter presents evidence that, unless you have the computational budget to fully train the network, more data is probably not necessary. In addition, **chapter 7** analysed the semantic content of the images themselves, finding that images from an unrelated source perform no worse than images from a related source. This can be considered an investigation into a situation where the cost of acquiring data from an unrelated dataset is far cheaper than acquiring data from a related source, thus showing an additional way in which the cost can be reduced. **Chapter 7** showed that there was no correlation in performance on the downstream task and the level of 'overlap' between the labelled and unlabelled dataset based on any of my metrics. In a real application, this would further reduce the cost burden of applying these deep learning approaches: because the features that are learned are non-specific, it allows us to train on a dataset that is the cheapest, rather than having to train on a

potentially costly unlabelled dataset. In addition, this problem could be treated as a foundational model, such as the large language models that have found success in the natural language space [159], [75], [78], in which a very large model is trained (usually in a semi-supervised way) to produce a generic model. This generic model can be used for its zero shot performance directly, or cheaply fine-tuned on a downstream task.

Finally, **Chapter 6** can be thought of as taking an alternative assumption about the cost of various aspects of the training process. It finds that performance can be further improved from the suggestions of Chen et al [1] by performing hyperparameter tuning of both the type and magnitude of the augmentations used during training. Under this assumption, it imagines how an implementer would want to increase performance when the cost of training a network is negligible compared to the cost of any data. Even in scenarios where the cost of training is not negligible, this advantage of hyperparameter tuning will likely remain.

Consistently throughout findings of this thesis, the most cost effective solution will depend on the exact cost differential between all components of the training protocol.

8.4 Future Work

Rather than highlighting what should be studied in the future, I start this section with what *shouldn't* be studied. **Chapter 4** outlined how, given extremely limited labelled data and large amounts of unlabelled data, learning from Contrastive Predictive Coding embeddings could improve the the performance of a ResNet compared with training directly on the pixels. However, it was found that this did not work on all datasets, even with the same experimental setup. In addition, the level of data that the network was found to be successful on is unrealistically small. At the low end of the data scale, the labelled datasets could consist of just six images per class. Outside of some infrequent situations, such as for rare disease detection, it is unlikely that this would be very useful. This conjecture is reinforced by the lack of literature that has come in the years since I started this project: initially,

I believed that this was a failure of the community to investigate this properly. I now, however, believe that the lack of literature is more to do with the positive publication bias that exists today [161].

Contrary to the lack of work found replicating CPC, SimCLR has been extensively studied. While not always replicating the high levels of performance found in [1], it often is found as a powerful baseline result (**chapter 5**) that matches the performance of other related methods. Based upon the results in **chapter 6** and **chapter 7**, I do not believe that we have reached the limits of what is possible to be accomplished with the SimCLR method. Some of the improvements that can be made to the contrastive methods are not necessarily improvements to the methods themselves, they are improvements to engineering practices to allow us to train these larger and larger networks. For example, the work conducted in **chapter 6** concludes that improvement and further utility of SimCLR can be obtained through hyperparameter tuning. This would be extremely cost intensive, and unlikely to be conducted until optimisations in the training time are created. However, if it were possible to run this search cheaply, then this could increase the number of modalities where these methods could be used. The same is also true for the work in **chapter 7**, which concludes that even with the size of dataset that researchers have at the moment, it is unlikely that they have trained the network for long enough to achieve peak performance. This again is more of a cost and engineering challenge than a scientific challenge [1]. I believe that if further resources are made available to train these models to a higher level, with longer training times, using bigger models and bigger batch sizes, trained using the extremely large datasets highlighted in **chapter 7**, then gains can be made with almost no change to the current method. While this is not a long term solution, and in the future more efficient models will come along,

[1]Since the body of this thesis was conducted, foundational models (section 2.7) have become prevalent. These models follow this principal, merely scaling the size of the models and sizes of the datasets they are trained. The main driving force behind this is the innovation in engineering practice that allows for training of these larger models. Examples of this could include increases in the size of GPU memory or software innovations that allow for hardware failure without training run failure [162].

we are at a stage at the moment where the ability of a network to learn features is not decided by the sophistication of the method, but by the economic ability of the designer to bear the cost of training these models. This approach has worked in the past: for example, Convolutional Neural Networks have been around since the 80s [163], however, it was not until it became economically realistic to train these models that deep learning began to take off. One of the main contributions of the Alexnet paper [41] was its use of GPU powered neural networks, taking advantage of the relative cost advantage of training on general purpose GPUs. Recent developments will also possibly lead to increased cost efficiency for network training, for example TPUs [147] and other ASICs [148]. This suggestion has not been explored in this thesis due to the large training cost of these models, this is discussed further in the limitations section.

In addition to these major areas for study, I highlight the following areas for further investigation. (1) Increasing the number of augmentations. **Chapter 6** examined the combination of two augmentations composed together, and Chen et al [1] explored using three augmentations combined together, with one being kept static as random crop. It is certainly possible that increasing the number of randomly applied augmentations above this level could improve performance, such as in the case of [143]. This, however, would require a large amount of compute power to explore and as highlighted before, is unlikely to be economical to study at this time. (2) In addition to extra augmentation, extra network backbones could be studied. For all methods under study in this thesis, a ResNet backbone has been used across all chapters. There is no intrinsic reason for this, any other neural backbone (for example vision transformers [164]) could have been chosen instead. However, as with the augmentation search, this grid-search would be computationally intensive to conduct.

Commentary on Foundation Models: Since the body of this work has been completed, there has been a surge in interest in so called foundational models (chapter 2). These models have followed principles set out in chapter 7, namely that previous models did not train for long enough and that generalist models can perform well across a number of tasks. Foundational models use orders of magnitude more compute than previous models. I highlight in chapter 2 that llama-3.1 used over 16000 top of the line GPUs concurrently, significantly more than was used for either CPC or SimCLR. One of the most influential papers within the foundational model domain is the "chinchilla scaling law" paper [165]. This paper proposed the same conclusions as I did in chapter 7, that state of the art model's performance could be increased scaling the amount of data. I believe this trend will continue for as long as more data is able to be collected and for as long as capital budgets allow for training these models.

8.5 Opinion on the Future

The appeal of semi-supervised learning as a method to improve deep learning's success remains. While semi-supervised learning has not, to date, surpassed the ImageNet accuracy of supervised learning, the fundamental reasons for interest have not changed since its inception: the cost of unlabelled data is substantially cheaper than labelled data. Just as a human learns from limited examples and leverages what it has seen previously despite no labels, I believe that the future of deep learning research will be semi-supervised. Processing power increases exponentially. Neural network model size is increasing rapidly. The amount of unlabelled, unstructured data is rapidly increasing as more people become connected to networks. However, labelled data will always remain expensive since it cannot be created without human input. I, therefore, believe that given further research into semi-supervised approaches, we are likely to see methods appear that outperform supervised performance across a more wide range of tasks.

Reinforcement learning is another area in which researchers are not limited by the labelled data. Under such a scenario, an agent (an AI) interacts with a world to optimise some kind of 'reward signal', thus, there is no constraint on the amount of data that the network can use. Larger constraints are the amount of processing power an implementer has available, and the efficiency of the network. This has been shown to produce superhuman results: [166], [167], [168]. I believe that semi-supervised learning is much the same. While there is technically a constraint on the amount of data that you have available. I do not believe that we have hit that level yet.

Another area for exploration is contrastive methods between multi-modal data. The current state of the art method for ImageNet performance is based on contrasting between images and their textual descriptions [169]. This method is similar to how humans handle multi-modal data, with the brain attempting to link multi-sensory inputs to the same underlying representation [170]. There is possibility that this will increase the sources of data that can be used for learning. While not contrastive, recent advances in Large Language Models such as Google Deepmind's Gemini [171] are inherently multi-modal. This allows them to take advantage of much larger sources of information than could be possible with just images alone.

One area where I think that the contrastive work could be problematic is the 'all vs one' paradigm taken in the contrastive loss. Under this paradigm, each image is contrasted with respect to all others, thus, in the case of a task such as supervised classification, the images of, for example, dogs, the embeddings of all dogs are pushed apart as much as the images of, for example, cats. Admittedly, this could be advantageous: such as in the case that we do not know what the downstream task is, or, when the dataset that we are conducting the downstream task on differs significantly from the unlabelled dataset. As discussed in chapter 7, this could be used to the advantage of the system designer: it becomes less necessary to collect an unlabelled dataset that matches the labelled dataset. By extension, this means that it becomes less necessary to retrain the encoder for every task, once again further reducing the cost of applying deep learning methods. However, my personal opinion

is that this will lead to inefficient training in the case where we do know what the downstream task is. I believe that it will be possible to in some way combine the upstream and downstream learning objectives in such a way to more efficiently learn a useful embedding. Ultimately, the measure of usefulness is defined by how well it works on whatever downstream task we decide. A universal encoder (that is one that works well on any task) may be useful, or it may not, it will entirely depend on what use case the network has.

I believe think that one further area for exploration relates to the efficiency of semi-supervised networks to learn from these large datasets. While earlier in the chapter (section 8.2) I was advocating for using additional resources to learn from these large datasets, a far better approach would rely on more efficient learning. A large number of the current state of the art models rely on huge numbers of GPUs and training time for learning the embeddings from, with [3] using 512 GPUs for training. This is an unreasonable cost for most practitioners in the field. For these networks to have more acceptance within the community, a research focus must be placed on the computational complexity of these high performing models.

8.6 Recommendations for the Future

This thesis has investigated how semi-supervised learning, specifically Contrastive Predictive Coding (CPC) and SimCLR, is able to be utilised for use with medical imaging. In this section, I present a set of recommendations for how best to apply semi-supervised learning. It is hoped that these recommendations, summarising the learnings of this thesis, can help an implementer push the boundaries of the state of the art.

8.6.1 Recommendation 1: Labelled data size

Semi-supervised learning achieves the greatest level of improvement over supervised learning when the number of images per class is low. Section 4.3.1 as well as section

4.4.3 have shown that there is a significant increase in performance when using just 1% of the dataset (20-30 images per class), however, when the number of images per class increases to 100% of the dataset (2-3k images per class) there is no increase in performance seen. This trend concurs with the work of Henaff [2] who showed that when using 1% (roughly 10 images per class), achieved a large increase in performance, but when using the full dataset, achieved a more modest increase. This result is despite the ImageNet dataset having much more images total, with 1% of the dataset corresponding to $\approx$ 14k total images. Implementors should be cognisant of this result: If images per class grows too large, it is unlikely that the benefit of semi-supervised learning will be felt.

8.6.2 Recommendation 2: Transfer Learning

Across this thesis, semi-supervised learning has shown to be extremely suitable for transfer learning. Initially, section 4.3.3 showed that the features learned by transfered well over to new domain. Additionally, the work of 7.10 and section 7.11 show that the features learned by both SimCLR and Contrastive Predictive Coding are general features, and it therefore unnecessary to produce task specific unlabelled datasets, vastly reducing the cost to apply SSL to new tasks. Based upon these results, I recommend that an implementer produces one, larger, more-powerful model, that could be reused across a number of domains. This is especially important due to the very large cost of training these models to saturation.

8.6.3 Recommendation 3: Computational Budget

Multiple experiments in this thesis have shown that to achieve the greatest level of performance with semi-supervised learning, an implementor needs to have sufficient levels of computational budget available. Most notably, chapter 7 argues that for SimCLR to sufficiently saturate a networks capability, a large computational budget is needed, much larger than is typically provided in the literature [3] [4]. This large computational budget, along with techniques for preventing overfitting (section 7.7) was able to fully saturate the network to approximately peak performance. Additionally, the work of chapter 6 hints at improved performance through the use of

additional computational power: by utilising this capability to perform an extensive hyperparameter sweep of possible sets of augmentations a significant increase in performance can be gained. Section 6.4 found that there was an increase in performance of 10.21% between the lowest and the highest performing augmentation set for dermatology and 22.3% for OCT. This increase in performance is well worth the increase in computational cost. Given these results, an implementer must ensure that the computational capacity that they have access to is large.

While this work includes other recommendations for how to increase performance for utilising semi-supervised learning for medical imaging, by following these three recommendations, I believe that the largest gain in performance can be achieved.

8.7 Concluding Remarks

This thesis explored contrastive learning as a method for semi-supervised learning across a number of novel dimensions. It explored the impact of both the unlabelled and labelled datasets on downstream performance, and the impact of augmentation, across two different popular, contrastive methods. This thesis asserts that increases in a model's performance can be achieved under certain conditions using contrastive learning. However, whether this is useful for an implementer is contingent: What is the cost to acquire labelled data? What is the cost of labelled data in comparison to unlabelled data? What is the cost to train a network to convergence? All of these questions have been shown to impact whether contrastive learning is a good choice for certain applications under the constrained environment of this thesis. This work concludes by giving specific, actionable advice on how an implementer can apply these semi-supervised methods to achieve the greatest level of performance.

Bibliography

[1] 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.

[2] Henaff O. Data-efficient image recognition with contrastive predictive coding. In: International Conference on Machine Learning. PMLR; 2020. p. 4182-92.

[3] Goyal P, Caron M, Lefaudeux B, Xu M, Wang P, Pai V, et al. Self-supervised pretraining of visual features in the wild. arXiv preprint arXiv:210301988. 2021.

[4] El-Nouby A, Izacard G, Touvron H, Laptev I, Jegou H, Grave E. Are large-scale datasets necessary for self-supervised pre-training? arXiv preprint arXiv:211210740. 2021.

[5] Caron M, Bojanowski P, Mairal J, Joulin A. Unsupervised pre-training of image features on non-curated data. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2019. p. 2959-68.

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

[7] 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.

[8] Goodfellow I, Bengio Y, Courville A. Deep learning. MIT press; 2016.

[9] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research. 2014;15(1):1929-58.

[10] Borgli H, Thambawita V, Smedsrud PH, Hicks S, Jha D, Eskeland SL, et al. HyperKvasir, a comprehensive multi-class image and video dataset for gastrointestinal endoscopy. Scientific data. 2020;7(1):1-14.

[11] Kermany DS, Goldbaum M, Cai W, Valentim CC, Liang H, Baxter SL, et al. Identifying medical diagnoses and treatable diseases by image-based deep learning. Cell. 2018;172(5):1122-31.

[12] Tschandl P, Rosendahl C, Kittler H. The HAM10000 dataset, a large collection of multi-source dermatoscopic images of common pigmented skin lesions. Scientific data. 2018;5(1):1-9.

[13] Potrimba P. What is an autoencoder?. Roboflow; 2022. Available from: https://blog.roboflow.com/what-is-an-autoencoder-computer-vision/.

[14] Noroozi M, Favaro P. Unsupervised learning of visual representations by solving jigsaw puzzles. In: European conference on computer vision. Springer; 2016. p. 69-84.

[15] Doersch C, Gupta A, Efros AA. Unsupervised visual representation learning by context prediction. In: Proceedings of the IEEE international conference on computer vision; 2015. p. 1422-30.

[16] Misra I, Maaten Lvd. Self-supervised learning of pretext-invariant representations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2020. p. 6707-17.

[17] Anonymous. Report: AI Investments See Largest Year-over-Year Growth in 20 Years; 2021. https://venturebeat.com/ai/report-ai-investments-see-largest-year-over-year-growth-in-20-years/. Available from: https://venturebeat.com/ai/report-ai-investments-see-largest-year-over-year-growth-in-20-years/.

[18] Tesla. Tesla Vehicle Safety Report; 2022. Available from: https://www.tesla.com/VehicleSafetyReport.

[19] Amazon. What Is Alexa?. Amazon;. Available from: https://developer.amazon.com/en-US/alexa.html.

[20] Apple. Apple;. Available from: https://www.apple.com/uk/siri/.

[21] Krizhevsky A, Hinton G, et al. Learning multiple layers of features from tiny images. University of Toronto Technical Report. 2009.

[22] MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges;. Available from: http://yann.lecun.com/exdb/mnist/.

[23] Deng J, Dong W, Socher R, Li LJ, Li K, Fei-Fei L. Imagenet: A large-scale hierarchical image database. In: 2009 IEEE conference on computer vision and pattern recognition. Ieee; 2009. p. 248-55.

[24] NHS medical staffing data analysis;. Available from: https://www.bma.org.uk/advice-and-support/nhs-delivery-and-workforce/workforce/nhs-medical-staffing-data-analysis.

[25] Carter A. Germany's looming GP shortage threatens regional coverage;. Available from: https://www.iamexpat.de/expat-info/german-expat-news/germanys-looming-gp-shortage-threatens-regional-coverage.

[26] Boyle P. U.S. physician shortage growing. AAMC;. Available from: https://www.aamc.org/news-insights/us-physician-shortage-growing.

[27] Haakenstad A, Irvine CMS, Knight M, Bintz C, Aravkin AY, Zheng P, et al. Measuring the availability of human resources for health and its relationship to universal health coverage for 204 countries and territories from 1990 to 2019: a systematic analysis for the Global Burden of Disease Study 2019. The Lancet. 2022.

[28] Parkinsons uk; 2020. Available from: https://www.parkinsons.org.uk/news/poll-finds-quarter-people-parkinsons-are-wrongly-diagnosed.

[29] Pearson C, Fraser J, Peake M, Valori R, Poirier V, Coupland VH, et al. Establishing population-based surveillance of diagnostic timeliness using linked cancer registry and administrative data for patients with colorectal and lung cancer. Cancer epidemiology. 2019;61:111-8.

[30] Lee JY, Jeong J, Song EM, Ha C, Lee HJ, Koo JE, et al. Real-time detection of colon polyps during colonoscopy using deep learning: systematic validation with four independent datasets. Scientific reports. 2020;10(1):1-9.

[31] Yamada M, Saito Y, Imaoka H, Saiko M, Yamada S, Kondo H, et al. Development of a real-time endoscopic image diagnosis support system using deep learning technology in colonoscopy. Scientific reports. 2019;9(1):1-9.

[32] Miller DD, Brown EW. Artificial intelligence in medical practice: the question to the answer? The American journal of medicine. 2018;131(2):129-33.

[33] Chen B, Solebo AL, Taylor P. Automated Image Quality Assessment for Anterior Segment Optical Coherence Tomograph. In: 2023 IEEE 20th International Symposium on Biomedical Imaging (ISBI). IEEE; 2023. p. 1-4.

[34] Papert S. The summer Vision Project. MIT;. Available from: `https://dspace.mit.edu/bitstream/handle/1721.1/6125/AIM-100.pdf?sequence=2&isAllowed=y`.

[35] 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.

[36] Bay H, Ess A, Tuytelaars T, Van Gool L. Speeded-up robust features (SURF). Computer vision and image understanding. 2008;110(3):346-59.

[37] Goodfellow IJ, Bengio Y, Courville A. Deep Learning. Cambridge, MA, USA: MIT Press; 2016. Http://www.deeplearningbook.org.

[38] edge-case. Cambridge University Press;. Available from: `https://dictionary.cambridge.org/dictionary/english/edge-case`.

[39] Pan SJ, Yang Q. A survey on transfer learning. IEEE Transactions on knowledge and data engineering. 2009;22(10):1345-59.

[40] Amazon Mechanical Turk;. Available from: `https://www.mturk.com/`.

[41] Krizhevsky A, Sutskever I, Hinton GE. Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems; 2012. p. 1097-105.

[42] Bowel Polyps. NHS;. Available from: `https://www.nhs.uk/conditions/bowel-polyps/`.

[43] Felicity. Explaining OCT scans with their mechanism and benefits; 2022. Available from: `https://www.rsipvision.com/explaining-oct-scans/`.

[44] What is Choroidal Neovascularization?. BrightFocus Foundation; 2021. Available from: `https://www.brightfocus.org/macular/article/what-choroidal-neovascularization`.

[45] Diabetic retinopathy. NHS;. Available from: `https://www.nhs.uk/conditions/diabetic-retinopathy/`.

[46] Pietrangelo A. Diabetic retinopathy vs. Diabetic Macular edema: Your faqs. Healthline Media; 2021. Available from: `https://www.healthline.com/health/diabetes/diabetic-retinopathy-vs-diabetic-macular-edema`.

[47] Silvestri G, Williams M, McAuley C, Oakes K, Sillery E, Henderson D, et al. Drusen prevalence and pigmentary changes in Caucasians aged 18–54 years. Eye. 2012;26(10):1357-62.

[48] Cafasso J. Drusen in eyes. Healthline Media; 2023. Available from: `https://www.healthline.com/health/drusen`.

[49] Risk calculator for the general public. Alfred Health;. Available from: `https://www.alfredhealth.org.au/melanoma-risk-calculator/public`.

[50] Benign mole (melanocytic naevi). MySkinDoctor; 2022. Available from: `https://www.myskindoctor.co.uk/benign-mole-melanocytic-naevi`.

[51] Ben-Yair S. Updating Google Photos' storage policy to build for the future. Google; 2020. Available from: `https://blog.google/products/photos/storage-changes/`.

[52] Shahinfar S, Meek P, Falzon G. "How many images do I need?" Understanding how sample size per class affects deep learning model performance metrics for balanced designs in autonomous wildlife monitoring. Ecological Informatics. 2020;57:101085.

[53] Lee DH, et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In: Workshop on challenges in representation learning, ICML. vol. 3; 2013. p. 896.

[54] Ren P, Xiao Y, Chang X, Huang PY, Li Z, Gupta BB, et al. A survey of deep active learning. ACM computing surveys (CSUR). 2021;54(9):1-40.

[55] Budd S, Robinson EC, Kainz B. A survey on active learning and human-in-the-loop deep learning for medical image analysis. Medical Image Analysis. 2021;71:102062.

[56] Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, et al. Generative adversarial networks. arXiv preprint arXiv:14062661. 2014.

[57] Schlegl T, Seeböck P, Waldstein SM, Schmidt-Erfurth U, Langs G. Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In: International Conference on Information Processing in Medical Imaging. Springer; 2017. p. 146-57.

[58] Google Inc;. Available from: `https://www.tensorflow.org/tutorials/generative/autoencoder`.

[59] Ahmed F, Courville A. Detecting semantic anomalies. arXiv preprint arXiv:190804388. 2019.

[60] Larsson G, Maire M, Shakhnarovich G. Colorization as a proxy task for visual understanding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition; 2017. p. 6874-83.

[61] Pathak D, Krahenbuhl P, Donahue J, Darrell T, Efros AA. Context encoders: Feature learning by inpainting. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 2536-44.

[62] Golhar M, Bobrow TL, Khoshknab MP, Jit S, Ngamruengphong S, Durr NJ. Improving colonoscopy lesion classification using semi-supervised deep learning. IEEE Access. 2020;9:631-40.

[63] Wang J, Song Y, Leung T, Rosenberg C, Wang J, Philbin J, et al. Learning Fine-grained Image Similarity with Deep Ranking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2014. .

[64] Hoffer E, Ailon N. Deep metric learning using triplet network. In: Similarity-Based Pattern Recognition: Third International Workshop, SIMBAD 2015, Copenhagen, Denmark, October 12-14, 2015. Proceedings 3. Springer; 2015. p. 84-92.

[65] Sohn K. Improved deep metric learning with multi-class n-pair loss objective. Advances in neural information processing systems. 2016;29.

[66] Schroff F, Kalenichenko D, Philbin J. Facenet: A unified embedding for face recognition and clustering. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2015. p. 815-23.

[67] Khosla P, Teterwak P, Wang C, Sarna A, Tian Y, Isola P, et al. Supervised contrastive learning. Advances in Neural Information Processing Systems. 2020;33:18661-73.

[68] 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.

[69] Bachman P, Hjelm RD, Buchwalter W. Learning representations by maximizing mutual information across views. Advances in neural information processing systems. 2019;32.

[70] Hjelm RD, Fedorov A, Lavoie-Marchildon S, Grewal K, Bachman P, Trischler A, et al. Learning deep representations by mutual information estimation and maximization. arXiv preprint arXiv:180806670. 2018.

[71] Tian Y, Krishnan D, Isola P. Contrastive multiview coding. In: European conference on computer vision. Springer; 2020. p. 776-94.

[72] Grill JB, Strub F, Altché F, Tallec C, Richemond PH, Buchatskaya E, et al. Bootstrap your own latent: A new approach to self-supervised learning. arXiv preprint arXiv:200607733. 2020.

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

[74] Bommasani R, Hudson DA, Adeli E, Altman R, Arora S, von Arx S, et al. On the opportunities and risks of foundation models. arXiv preprint arXiv:210807258. 2021.

[75] 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.

[76] Dubey A, Jauhri A, Pandey A, Kadian A, Al-Dahle A, Letman A, et al. The Llama 3 Herd of Models. arXiv preprint arXiv:240721783. 2024.

[77] Schuhmann C, Beaumont R, Vencu R, Gordon C, Wightman R, Cherti M, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. Advances in Neural Information Processing Systems. 2022;35:25278-94.

[78] Touvron H, Martin L, Stone K, Albert P, Almahairi A, Babaei Y, et al.. Llama 2: Open Foundation and Fine-Tuned Chat Models; 2023.

[79] Jiang AQ, Sablayrolles A, Mensch A, Bamford C, Chaplot DS, Casas Ddl, et al. Mistral 7B. arXiv preprint arXiv:231006825. 2023.

[80] Devlin J. Bert: Pre-training of deep bidirectional transformers for language understanding. arXiv preprint arXiv:181004805. 2018.

[81] Roth HR, Lu L, Farag A, Shin HC, Liu J, Turkbey EB, et al. Deeporgan: Multi-level deep convolutional networks for automated pancreas segmentation. In: Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part I 18. Springer; 2015. p. 556-64.

[82] Rahman T, Khandakar A, Kadir MA, Islam KR, Islam KF, Mazhar R, et al. Reliable tuberculosis detection using chest X-ray with deep learning, segmentation and visualization. Ieee Access. 2020;8:191586-601.

[83] Kirillov A, Mintun E, Ravi N, Mao H, Rolland C, Gustafson L, et al. Segment anything. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2023. p. 4015-26.

[84] 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.

[85] Machine learning in the AWS cloud: Add intellegence to applications with Amazon Sagemaker and Amazon Rekognition. Amazon;. Available from: `https://aws.amazon.com/sagemaker/data-labeling/`.

[86] V7 Labs. V7;. Available from: `https://www.v7labs.com/`.

[87] Zhai X, Oliver A, Kolesnikov A, Beyer L. S4l: Self-supervised semi-supervised learning. In: Proceedings of the IEEE international conference on computer vision; 2019. p. 1476-85.

[88] Donahue J, Simonyan K. Large scale adversarial representation learning. In: Advances in Neural Information Processing Systems; 2019. p. 10541-51.

[89] 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.

[90] Gutmann M, Hyvärinen A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In: Proceedings of the thir-

teenth international conference on artificial intelligence and statistics. JMLR Workshop and Conference Proceedings; 2010. p. 297-304.

[91] Pearson K. LIII. On lines and planes of closest fit to systems of points in space. The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science. 1901;2(11):559-72.

[92] Lu MY, Chen RJ, Wang J, Dillon D, Mahmood F. Semi-supervised histology classification using deep multiple instance learning and contrastive predictive coding. arXiv preprint arXiv:191010825. 2019.

[93] Taleb A, Loetzsch W, Danz N, Severin J, Gaertner T, Bergner B, et al. 3d self-supervised methods for medical imaging. arXiv preprint arXiv:200603829. 2020.

[94] Zhu J, Li Y, Hu Y, Zhou SK. Embedding Task Knowledge into 3D Neural Networks via Self-supervised Learning. arXiv preprint arXiv:200605798. 2020.

[95] Zhou HY, Lu C, Yang S, Han X, Yu Y. Preservational Learning Improves Self-supervised Medical Image Models by Reconstructing Diverse Contexts. In: Proceedings of the IEEE/CVF International Conference on Computer Vision; 2021. p. 3499-509.

[96] Mehari T, Strodthoff N. Self-supervised representation learning from 12-lead ECG data. arXiv preprint arXiv:210312676. 2021.

[97] Liu H, Zhao Z, She Q. Self-supervised ECG pre-training. Biomedical Signal Processing and Control. 2021;70:103010.

[98] Kallidromitis K, Gudovskiy D, Kazuki K, Iku O, Rigazio L. Contrastive Neural Processes for Self-Supervised Learning. arXiv preprint arXiv:211013623. 2021.

[99] Mohsenvand MN, Izadi MR, Maes P. Contrastive representation learning for electroencephalogram classification. In: Machine Learning for Health. PMLR; 2020. p. 238-53.

[100] Haggar FA, Boushey RP. Colorectal cancer epidemiology: incidence, mortality, survival, and risk factors. Clinics in colon and rectal surgery. 2009;22(04):191-7.

[101] Aslanian HR, Shieh FK, Chan FW, Ciarleglio MM, Deng Y, Rogart JN, et al. Nurse observation during colonoscopy increases polyp detection: a randomized prospective study. Official journal of the American College of Gastroenterology— ACG. 2013;108(2):166-72.

[102] Lee CK, Park DI, Lee SH, Hwangbo Y, Eun CS, Han DS, et al. Participation by experienced endoscopy nurses increases the detection rate of colon polyps during a screening colonoscopy: a multicenter, prospective, randomized study. Gastrointestinal endoscopy. 2011;74(5):1094-102.

[103] Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation. arXiv preprint arXiv:14061078. 2014.

[104] Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:14126980. 2014.

[105] Chollet F, et al.. Keras; 2015. https://keras.io.

[106] Representation Learning with Contrastive Predictive Coding;. Available from: `https://github.com/davidtellez/contrastive-predictive-coding`.

[107] Buslaev A, Parinov A, Khvedchenya E, Iglovikov VI, Kalinin AA. Albumentations: fast and flexible image augmentations. ArXiv e-prints. 2018.

[108] Akhtar N, Mian A. Threat of adversarial attacks on deep learning in computer vision: A survey. Ieee Access. 2018;6:14410-30.

[109] Sánchez-Peralta LF, Picón A, Sánchez-Margallo FM, Pagador JB. Unravelling the effect of data augmentation transformations in polyp segmentation. International journal of computer assisted radiology and surgery. 2020;15(12):1975-88.

[110] Perone CS, Ballester P, Barros RC, Cohen-Adad J. Unsupervised domain adaptation for medical imaging segmentation with self-ensembling. NeuroImage. 2019;194:1-11.

[111] Wang W, Tian J, Zhang C, Luo Y, Wang X, Li J. An improved deep learning approach and its applications on colonic polyp images detection. BMC Medical Imaging. 2020;20(1):1-14.

[112] What is An OCT scan?: Optical Coherence Tomography. Specsavers UK;. Available from: https://www.specsavers.co.uk/eye-health/oct-scan.

[113] Dunaief J, Institute SE. What is Choroidal Neovascularization?. BrightFocus Foundation; 2020. Available from: https://www.brightfocus.org/macular/article/what-choroidal-neovascularization.

[114] Diabetic Macular Edema (DME). Prevent Blindness; 2020. Available from: https://preventblindness.org/diabetic-macular-edema-dme/.

[115] By the way, doctor: What are drusen, and why do I have them?; 2008. Available from: https://www.health.harvard.edu/newsletter_article/By_the_way_doctor_What_are_drusen_and_why_do_I_have_them.

[116] Radiation: Ultraviolet (UV) radiation and skin cancer. World Health Organization; 2017. Available from: https://www.who.int/news-room/q-a-detail/radiation-ultraviolet-(uv)-radiation-and-skin-cancer.

[117] Melanocytic Naevi (Pigmented Moles). British Association of Dermatologists; 2005.

[118] Mooney CZ. Bootstrap statistical inference: Examples and evaluations for political science. American Journal of Political Science. 1996:570-602.

[119] Dehaene O, Camara A, Moindrot O, de Lavergne A, Courtiol P. Self-supervision closes the gap between weak and strong supervision in histology. arXiv preprint arXiv:201203583. 2020.

[120] Stacke K, Lundström C, Unger J, Eilertsen G. Evaluation of Contrastive Predictive Coding for Histopathology Applications. In: Machine Learning for Health. PMLR; 2020. p. 328-40.

[121] Urban G, Tripathi P, Alkayali T, Mittal M, Jalali F, Karnes W, et al. Deep learning localizes and identifies polyps in real time with 96% accuracy in screening colonoscopy. Gastroenterology. 2018;155(4):1069-78.

[122] Wang P, Xiao X, Glissen Brown JR, Berzin TM, Tu M, Xiong F, et al. Development and validation of a deep-learning algorithm for the detection of polyps during colonoscopy. Nature biomedical engineering. 2018;2(10):741-8.

[123] Byrne MF, Chapados N, Soudan F, Oertel C, Pérez ML, Kelly R, et al. Real-time differentiation of adenomatous and hyperplastic diminutive colorectal polyps during analysis of unaltered videos of standard colonoscopy using a deep learning model. Gut. 2019;68(1):94-100.

[124] Su JR, Li Z, Shao XJ, Ji CR, Ji R, Zhou RC, et al. Impact of a real-time automatic quality control system on colorectal polyp and adenoma detection: a prospective randomized controlled study (with videos). Gastrointestinal endoscopy. 2020;91(2):415-24.

[125] Ye M, Zhang X, Yuen PC, Chang SF. Unsupervised embedding learning via invariant and spreading instance feature. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition; 2019. p. 6210-9.

[126] He K, Fan H, Wu Y, Xie S, Girshick R. Momentum contrast for unsupervised visual representation learning. arXiv preprint arXiv:191105722. 2019.

[127] Azizi S, Mustafa B, Ryan F, Beaver Z, Freyberg J, Deaton J, et al. Big self-supervised models advance medical image classification. arXiv preprint arXiv:210105224. 2021.

[128] Sowrirajan H, Yang J, Ng AY, Rajpurkar P. Moco pretraining improves representation and transferability of chest x-ray models. In: Medical Imaging with Deep Learning. PMLR; 2021. p. 728-44.

[129] Sun J, Wei D, Ma K, Wang L, Zheng Y. Unsupervised Representation Learning Meets Pseudo-Label Supervised Self-Distillation: A New Approach to Rare Disease Classification. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer; 2021. p. 519-29.

[130] Dufumier B, Gori P, Victor J, Grigis A, Duchesnay E. Conditional Alignment and Uniformity for Contrastive Learning with Continuous Proxy Labels. arXiv preprint arXiv:211105643. 2021.

[131] Manna S, Bhattacharya S, Pal U. SSLM: Self-Supervised Learning for Medical Diagnosis from MR Video. arXiv preprint arXiv:210410481. 2021.

[132] Zeng D, Wu Y, Hu X, Xu X, Yuan H, Huang M, et al. Positional Contrastive Learning for VolumetricMedical Image Segmentation. arXiv preprint arXiv:210609157. 2021.

[133] Dufumier B, Gori P, Victor J, Grigis A, Wessa M, Brambilla P, et al. Contrastive Learning with Continuous Proxy Meta-Data for 3D MRI Classification. arXiv preprint arXiv:210608808. 2021.

[134] Zhao Q, Liu Z, Adeli E, Pohl KM. Longitudinal self-supervised learning. Medical Image Analysis. 2021;71:102051.

[135] Abbet C, Studer L, Fischer A, Dawson H, Zlobec I, Bozorgtabar B, et al. Self-Rule to Adapt: Learning Generalized Features from Sparsely-Labeled Data Using Unsupervised Domain Adaptation for Colorectal Cancer Tissue Phenotyping. In: Medical Imaging with Deep Learning; 2021. .

[136] Li Z, Cui Z, Wang S, Qi Y, Ouyang X, Chen Q, et al. Domain Generalization for Mammography Detection via Multi-style and Multi-view Contrastive Learning. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer; 2021. p. 98-108.

[137] Hosseinzadeh Taher MR, Haghighi F, Feng R, Gotway MB, Liang J. A Systematic Benchmarking Analysis of Transfer Learning for Medical Image Analysis. In: Domain Adaptation and Representation Transfer, and Affordable Healthcare and AI for Resource Diverse Global Health. Springer; 2021. p. 3-13.

[138] Chen YP, Lo YH, Lai F, Huang CH. Disease Concept-Embedding Based on the Self-Supervised Method for Medical Information Extraction from Electronic Health Records and Disease Retrieval: Algorithm Development and Validation Study. Journal of Medical Internet Research. 2021;23(1):e25113.

[139] Torop M, Ghimire S, Liu W, Brooks DH, Camps O, Rajadhyaksha M, et al. Unsupervised Approaches for Out-Of-Distribution Dermoscopic Lesion Detection. arXiv preprint arXiv:211104807. 2021.

[140] 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.

[141] 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.

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

[143] Huang Y, Lin L, Cheng P, Lyu J, Tang X. Lesion-based contrastive learning for diabetic retinopathy grading from fundus images. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. Springer; 2021. p. 113-23.

[144] Abadi M, Agarwal A, Barham P, Brevdo E, Chen Z, Citro C, et al.. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems; 2015. Software available from tensorflow.org. Available from: `https://www.tensorflow.org/`.

[145] 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.

[146] Oliver A, Odena A, Raffel C, Cubuk ED, Goodfellow IJ. Realistic evaluation of deep semi-supervised learning algorithms. arXiv preprint arXiv:180409170. 2018.

[147] Jouppi NP, Young C, Patil N, Patterson D, Agrawal G, Bajwa R, et al. In-datacenter performance analysis of a tensor processing unit. In: Proceedings of the 44th annual international symposium on computer architecture; 2017. p. 1-12.

[148] Markidis S, Der Chien SW, Laure E, Peng IB, Vetter JS. Nvidia tensor core programmability, performance & precision. In: 2018 IEEE international parallel and distributed processing symposium workshops (IPDPSW). IEEE; 2018. p. 522-31.

[149] Narang S, Chowdhery A. Pathways Language Model (PaLM): Scaling to 540 Billion Parameters for Breakthrough Performance. Google;. Available from: `https://ai.googleblog.com/2022/04/pathways-language-model-palm-scaling-to.html`.

[150] Yalniz DMIZ, Jégou H. Billion-scale semi-supervised learning for state-of-the-art image and video classification; 2019.

[151] About ImageNet;. Available from: `https://image-net.org/about.php`.

[152] Mahajan D, Girshick R, Ramanathan V, He K, Paluri M, Li Y, et al. Exploring the limits of weakly supervised pretraining. In: Proceedings of the European conference on computer vision (ECCV); 2018. p. 181-96.

[153] Darlow LN, Crowley EJ, Antoniou A, Storkey AJ. Cinic-10 is not imagenet or cifar-10. arXiv preprint arXiv:181003505. 2018.

[154] Wang X, Peng Y, Lu L, Lu Z, Bagheri M, Summers RM. Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In: Proceedings of the IEEE conference on computer vision and pattern recognition; 2017. p. 2097-106.

[155] Weerakkody Y. Pulmonary infiltrates: Radiology Reference Article. Radiopaedia;. Available from: https://radiopaedia.org/articles/pulmonary-infiltrates-1?lang=gb.

[156] Amazon Web Services. Amazon EC2 On-Demand Pricing; 2024. Available from: https://aws.amazon.com/ec2/pricing/on-demand/.

[157] Weinstein MC, Torrance G, McGuire A. QALYs: the basics. Value in health. 2009.

[158] Timmins N. Ministers, not NHS England, should decide on the affordability of cost-effective new treatments. The King's Fund; 2017. Available from: https://www.kingsfund.org.uk/publications/articles/ministers-not-nhs-england-should-decide-affordability-of-treatments.

[159] OpenAI. GPT-4 Technical Report. arXiv preprint arXiv: 230308774. 2023.

[160] Knight W. OpenAI's CEO Says the Age of Giant AI Models Is Already Over. Wired. 2023 Apr.

[161] Mlinarić A, Horvat M, Šupak Smolčić V. Dealing with the positive publication bias: Why you should really publish your negative results. Biochemia medica. 2017;27(3):447-52.

[162] AI M. Facebook AI, AWS Partner to Release New PyTorch Libraries; 2020. https://ai.meta.com/blog/facebook-ai-aws-partner-to-release-new-pytorch-libraries-/.

[163] Fukushima K, Miyake S. Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition. In: Competition and cooperation in neural nets. Springer; 1982. p. 267-85.

[164] Dosovitskiy A. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:201011929. 2020.

[165] Hoffmann J, Borgeaud S, Mensch A, Buchatskaya E, Cai T, Rutherford E, et al. Training compute-optimal large language models. arXiv preprint arXiv:220315556. 2022.

[166] Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, et al. Mastering the game of Go with deep neural networks and tree search. nature. 2016;529(7587):484-9.

[167] Silver D, Hubert T, Schrittwieser J, Antonoglou I, Lai M, Guez A, et al. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. arXiv preprint arXiv:171201815. 2017.

[168] Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, et al. Human-level control through deep reinforcement learning. nature. 2015;518(7540):529-33.

[169] Yu J, Wang Z, Vasudevan V, Yeung L, Seyedhosseini M, Wu Y. Coca: Contrastive captioners are image-text foundation models. arXiv preprint arXiv:220501917. 2022.

[170] Lewkowicz DJ, Ghazanfar AA. The emergence of multisensory systems through perceptual narrowing. Trends in cognitive sciences. 2009;13(11):470-8.

[171] Gemini Team. Gemini: A Family of Highly Capable Multimodal Models. Google; 2023.

[172] Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: from error visibility to structural similarity. IEEE transactions on image processing. 2004;13(4):600-12.

[173] Mnih A, Kavukcuoglu K. Learning word embeddings efficiently with noise-contrastive estimation. Advances in neural information processing systems. 2013;26.

[174] Hinton G, Vinyals O, Dean J, et al. Distilling the knowledge in a neural network. arXiv preprint arXiv:150302531. 2015;2(7).

Appendix A

Additional Experimental Results

This appendix includes findings from my PhD research that, while not directly contributing to the main narrative, were still conducted and are included for completeness. These results are referred to throughout the main body of the work.

Experiments A.1 and A.2 investigate changes to the CPC training protocol and experiment A.3 investigates whether CPC learns features specific to a dataset or more general features.

A.1 Change in CPC Protocol

[2] suggests a number of changes to the CPC training protocol of CPCv1 intended to increase the performance of the protocol (see section 3.3). This experiment examines whether changing the learning mechanism from a linear layer to a ResNet improves performance on a set of medical imaging datasets.

Experimental Design: In this experiment, the performance of the CPC protocol is compared when using either a linear layer or ResNet as the downstream learning mechanism. For dataset subsets in {1%, 2%, 5%, 10%, 20%, 50%, 100%} of the full dataset, a ResNet and Linear layer is trained and evaluated on its classification performance for the three datasets under study. Twenty repeats are conducted for each of the data amounts, model types and datasets. This experimental design broadly

follows the design of section 4.4.

Datasets: The same medical imaging datasets used in chapter 4 have been used in this experiment.

Training Details: Following the experimental design of chapter 4: the networks are trained for a maximum of 1000 epochs, using early stopping with a patience of 50 and The ADAM optimiser with a learning rate of 5e-4. Twenty repeats are conducted, reporting the mean value with 95% confidence intervals.

Results: Figure A.1 shows the performances of both the ResNets and linear layers trained on the same CPC embeddings. The ResNet learning mechanism has a consistently higher mean performance, albeit not always a statistically significant difference. Interestingly, the linear layer trained on OCT data peaks at below 50% accuracy, whereas the ResNet does not peak. This result concurs with the work of Henaff et al [2]: the change from a linear layer either increases performance or does not harm performance. Based on the results of this section, it is recommended that, to achieve the best results from Contrastive Predictive Coding, ResNets should be used as the learning mechanism in the downstream task.

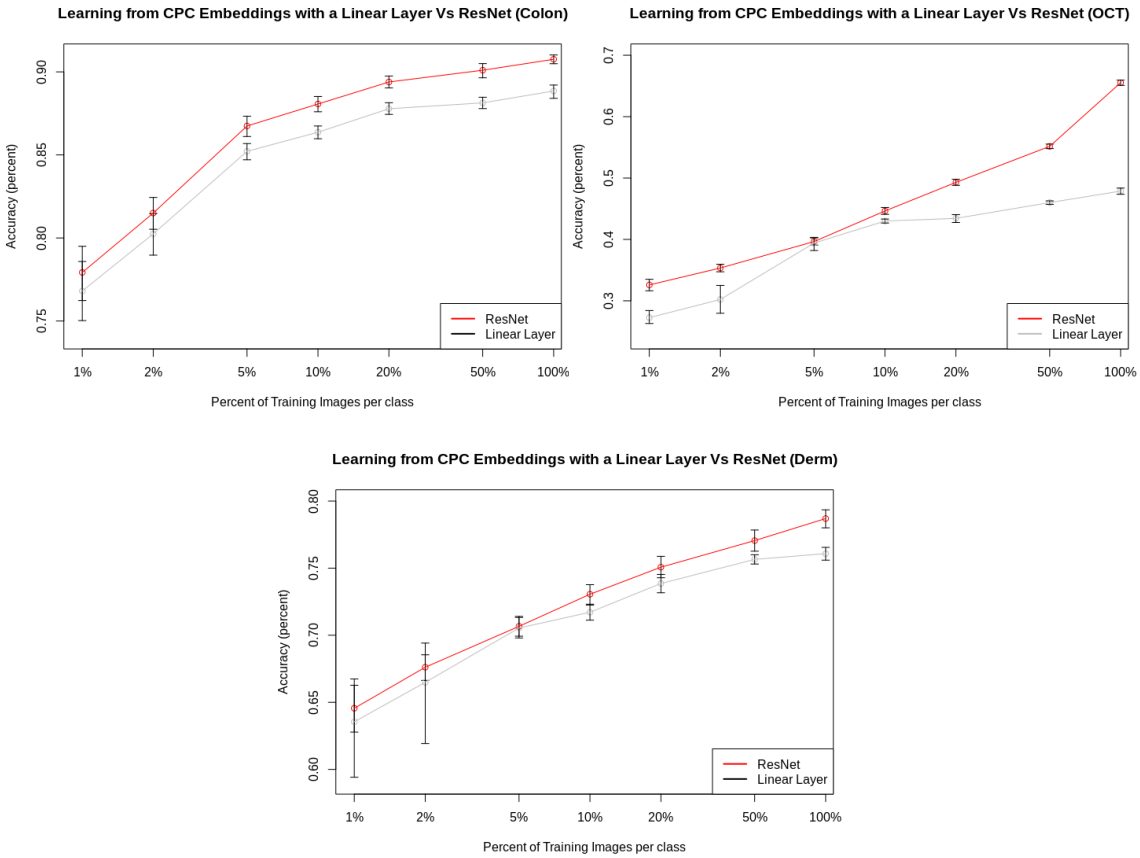

Figure A.1: Impact of changing secondary learning mechanism, comparing: a linear layer and a ResNet across the three datasets (colonoscopy, OCT, Dermatology), using variously sized subsets of the full dataset. None overlapping bars show significance.

A.2 Magnitude of Augmentation: Additional Result

Additional result from section 6.5, the full experimental details can be found there. Section 6.5 investigates whether the level of augmentation affects downstream performance for classification. This additional result contains a far greater amount of noise than in section 6.5 (10x), however, the same conclusion is reached: The amount of additive noise does not affect performance of the network.

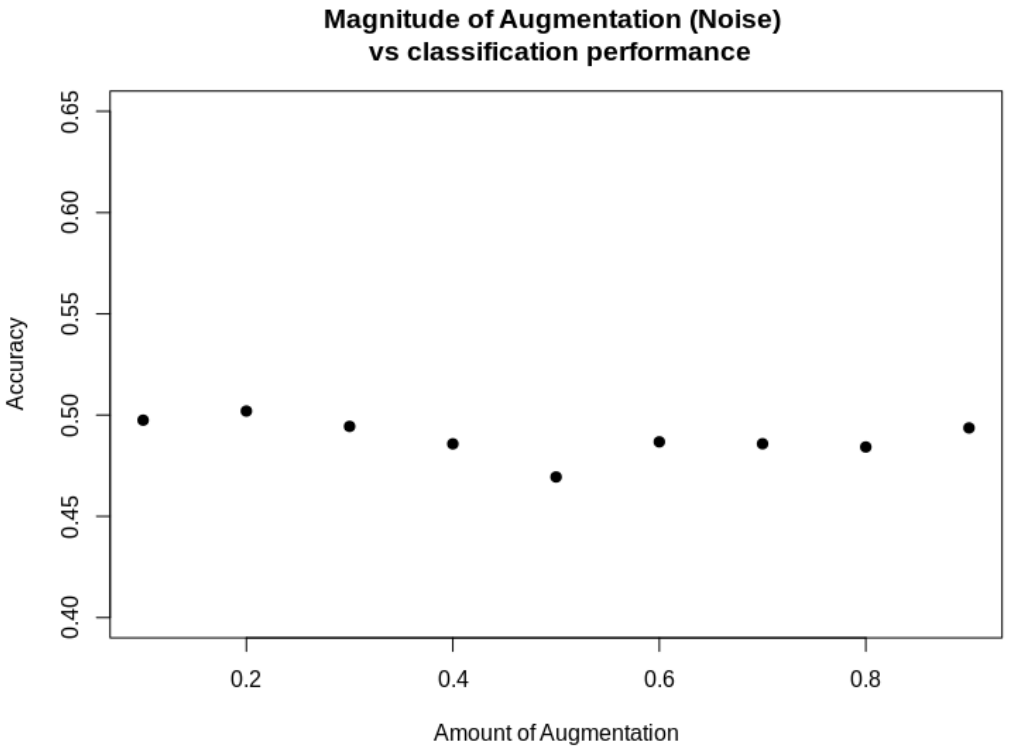

Figure A.2: Network performance against augmentation amount for additive noise.

A.3 Investigating metrics for distribution overlap

Prior work has suggested that as the distribution of the unlabelled dataset deviates from the training dataset, the performance gained by the semi-supervised approach diminishes [146]. In this section, metrics are examined which could be used to test how well the distributions overlap. If a metric could be found that could partially predict how well SimCLR could learn the data, more informative unlabelled datasets could be produced.

A.3.1 Metrics

In the section 7.8, a proxy measure of 'variability' was used: the number of semantic classes in the unsupervised dataset. This proxy would obviously not work in a real setting, as if one had access to the labels, one could just train a supervised model and get the high levels of performance that supervised networks enjoy over even state-of-the-art semi-supervised models. In this section, a number of candidate metrics are examined, ones that do not rely on semantic label:

Structural Similarity

Structural Similarity (SSIM) [172] is a metric for testing the similarity between two images. In this section, a distribution is generated by repeatedly taking two random images: one from the unsupervised dataset and one from the labelled dataset and calculating the SSIM between the two images. The standard deviation is then calculated on this distribution. This works as a proxy to the 'variability' of the dataset, a proxy that does not require labels unlike the proxy given in section 7.8. It is, therefore, able to be used on an unlabelled dataset.

Mean Squared Error

The Mean Squared Error (MSE) metric is conducted in a similar way to the SSIM metric. In this metric, two random images are selected: one image from the training dataset; and one from the proposed unsupervised dataset. The mean squared difference of the two images is taken and stored, this is repeated 10000 times. The sum of these metrics over the 10000 repeats is calculated and reported as the single dataset metric.

Kolmogorov-Smirnov Test

If two images are similar, they should have similar greyscale distributions. For this test, a similar methodology as the previous two metrics is followed, in that two images are randomly selected: one from the training dataset, and one from the unlabelled dataset. Each of the images are converted to greyscale and their distributions compared using the Kolmogorov-Smirnov (KS) Test. This is then repeated 10000 times, and the average value for this test is plotted across all datasets used in the experiment from section 7.8.

A.3.2 Experimental Design

This section is attempting to find a metric that can be used to examine a unlabelled dataset to predict whether it will perform well at the downstream prediction task. Three metrics are proposed as possible metrics to predict this performance: SSIM; Mean Squared Error; and the Kolmogorov-Smirnov Test. The 88 networks and

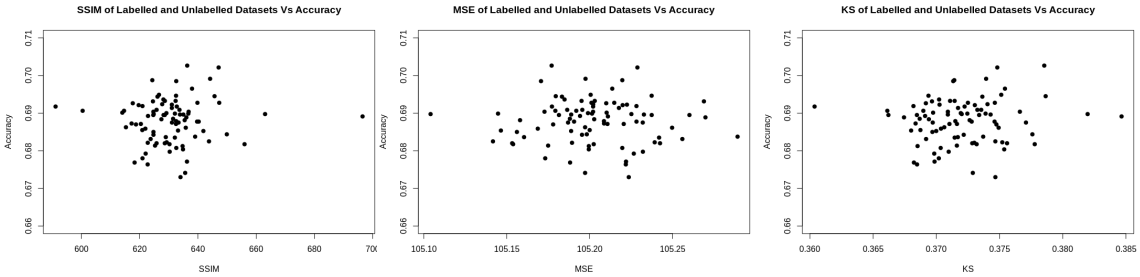

Figure A.3: Proposed metric vs downstream accuracy across all 88 unsupervised datasets. From left-to-right there is: Structural Similarity; Mean Squared Error; and KS.

datasets used for section 7.8 are reused for this section. The metrics are calculated for each of the datasets used, and plotted vs the accuracy of the downstream supervised task. Statistical analysis is then conducted to find the most suitable metric for dataset description analysis.

Statistical Tests: As with the test in the previous section, this section explores whether there is a correlation between the proposed metrics (listed above) and the performance of a network found in the previous section. For this reason, Spearman's rank has also been chosen for this section. Bonferroni correction is used due to having multiple metrics being tested. Following the equation set out in chapter 2, the Bonferroni corrected significance value will be $\frac{0.05}{3} = 0.0167$.

A.3.3 Results

Figure 7.8 shows three plots of the three metrics given, plotted against their downstream classification performance. There was no statistically significant correlation found between the SSIM overlap metric and the performance of the network for downstream classification performance (P = 0.6324). There was also no statistically significant correlation found between the Mean Squared Error metric and the performance of the network for downstream classification performance (P = 0.9481). There was no statistically significant correlation between the K-S metric and the downstream classification performance (P = 0.4162).

Appendix B

Understanding Noise Contrastive Estimator

B.1 Introduction

This appendix summarises and explains the evolution of the loss function widely used in contrastive learning. It covers how Noise Contrastive Estimators evolves through a number of changes to become NT-Xent. Broadly, all the methods covered in this section have the the same goal: to train an encoder such that similar information is together in the latent space, and dissimilar information is far apart. This appendix does not introduce any new information, merely provides a convenient summary for the reader. This appendix start by introducing the Noise Contrastive Estimator, a method for learning the underlying distribution of a dataset. This model is then computationally simplified using the Simplified Scalable log-bilinear models set out in B.3. The method is then changed to take the form of a multi-class problem, while at the same time giving a possible solution for how the noise should be generated. Finally, this chapter explores NT-Xent, and look at the performance increasing features introduced.

B.2 Noise Contrastive Estimation

Noise Contrastive Estimation (NCE) [90] is a estimation principle to model the underlying distribution of a set of data. This modelling is accomplished through distinguishing between the data within the set and from some kind of artificial noise. This loss function is conceptualised as a supervised learning problem: training a network to complete a binary classification problem, classifying real data from noise. Given a set of real data: $\{X_1, X_2, X_3, X_4, ...X_N\}$ and a set of noise $\{Y_1, Y_2, Y_3, Y_4, ...Y_N\}$, the following objective function is optimised:

$$J_T(\theta) = \frac{1}{2T} \sum_t ln[h(\mathbf{x}_t; \theta)] - ln[1 - h(\mathbf{y}_t; \theta)] \tag{B.1}$$

Where:

$$h(\mathbf{u}; \theta) = \frac{1}{1 + exp(-G(u; \theta))} \tag{B.2}$$

$$G(\mathbf{u}; \theta) = lnp_m(\mathbf{u}; \theta) - lnp_n(\mathbf{u}) \tag{B.3}$$

That is, the objective is to maximise the difference between a positive item and noise, and minimise the distance between noise and noise. $h(\mathbf{u}; \theta)$ is the sigmoid function (eq B.2) of the difference between the probability that the datum comes from and the noise distribution (eq B.3).

The authors note that the closer the noise distribution is to the data distribution, the better the model will be. It is important to point out that the representations produced by the NCE are not normalised: the model must learn to produce normalised vectors by itself. Optimisation of the objective will result in a statistical model of the data which can then be used in a further task.

In contrast to a lot of previous unsupervised methods such as PCA [91], the Noise Contrastive Estimator method conceptualises the unsupervised task as a supervised learning problem.

B.3 Simplified Scalable log-bilinear models

To use the previous method, the probability $p_m(\mathbf{u};\theta)$ and $p_n(\mathbf{u})$ must first be calculated. An approach taken is Scalable log-bilinear models which are derived from Neural Probabilistic Language Models (NPLM). These Scalable log-bilinear models are trained to give a probability for a given word in a given context. Due to the very large vocabularies found in natural language processing tasks, computational efficiency is paramount. [173] proposed the following scoring function as a simplification of their main proposed method:

$$S(w_i, w) = r_w^\top q_{wi} + b_{wi} \tag{B.4}$$

That is, that the similarity of two words (specified as vectors) can be found by taking the dot product with some context vector, and accounting for the context independent frequency with a bias term (b_{wi}).

B.4 InstDisc

InstDisc [140] present a novel framework for semi-supervised learning in which they consider each instance in a unlabelled dataset to be a class by themselves. This work was conducted in parallel to the work below.

B.5 InfoNCE

The InfoNCE loss function is a loss based on the work outlined in the previous three sections (B.2-B.4), introduced in [6]. The InfoNCE objective function combines the NCE principal of learning to distinguish between noise and real data, with the computational efficiency of the similarity metric found in the previous section. In addition to this, they add two changes to the methodologies:

Change 1: Rather than being analogous to a binary classification problem, Oord et al define the problem as a multi-class classification problem. In this change, the network is 'shown' a large batch of latent representations and it must 'decide' which

of the representations are from the data distribution, and which are from the noise distribution, given a context representation.

Change 2: As noted in the Noise Contrastive Estimator section: a noise model is needed that is close to the true data distribution. To create a a noise model that is similar, but distinct, from the non-noise, samples are taken from a large set of data. In the InfoNCE task, the large set of data consists of data taken from the dataset that is not currently being used in the batch.

This leads to the following loss function:

$$L_{CPC} = -\sum_{i,j,k} log \frac{exp(\hat{z}^T_{i+k,j} z_{i+k,j})}{exp(\hat{z}^T_{i+k,j} z_{i+k,j}) + \sum_l exp(\hat{z}^T_{i+k,j} z_l)} \tag{B.5}$$

This loss function calculates the probability of a specific latent embedding being next in the sequence using the categorical cross-entropy of the softmax across the dot products of the latent representations of the noise and the context. The term within the exp of the numerator refers to the dot product (used as a similarity score, B.3) Using the softmax ensures that the probabilities sum to 1 across the batch.

B.6 NT-Xent

The NT-Xent loss function is a further evolution of the contrastive loss function found in many contrastive methods papers: CMC; MoCo; PIRL; and SimCLR. This loss function adds a temperature scaling parameter and, in contrast to the NCE and InfoNCE loss functions, normalises the vector. The equation can be found below:

$$l_{i,j} = -log \frac{exp(sim(z_i, z_j)/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k \neq i]} exp(sim(z_i, z_k)/\tau)} \tag{B.6}$$

This loss function introduces two changes from the InfoNCE loss function:

Change 1: Normalisation of the vectors while measuring similarity. It has been found that normalising these vectors leads to higher performance, this is despite the

NCE loss function explicitly not requiring normalisation.

Change 2: Temperature scaling of the feature spaces has been included that were first proposed in [174]. This has been found to increase performance.