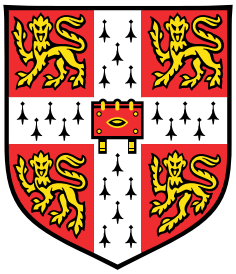
Minimal Labels, Maximum Gain. Image Classification with Graph-Based Semi-Supervised Learning



Philip Sellars

Department of Applied Maths and Theoretical Physics
University of Cambridge

This dissertation is submitted for the degree of
Doctor of Philosophy

Selwyn College October 2021

Declaration

This thesis is the result of my own work and includes nothing which is the outcome of work done in collaboration except as declared in the Preface and specified in the text. I further state that no substantial part of my thesis has already been submitted, or, is being concurrently submitted for any such degree, diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. It does not exceed the prescribed word limit for the relevant Degree Committee

Philip Sellars
October 2021

Acknowledgements

All good things must come to an end. I would like to thank all my colleagues in the Cambridge Image Analysis group and all my research collaborators for their help, support, ideas and guidance. In particular I offer my deepest thanks to my supervisors Dr Angelica Aviles-Rivero and Professor Carola-Bibiane Schönlieb for their constant support and help, and without whom this PhD would not have been realised. The unwavering love of my parents: Paul and Diane and my partner Maria kept me strong throughout and I owe them more than an acknowledgement could ever say. I also want to give thanks to all my friends from Girton and Selwyn college: Amy, Ingrid and others, for a lifetime of memories.

Abstract

In the last decade, the use and deployment of machine learning systems for computer vision has risen dramatically. To train a machine learning model it is often assumed that the practitioner has access to a large and representative labelled dataset from which they can optimise their model in a supervised manner. However, in many domains, there is a large cost to obtaining labelled data. In technical fields we need manual annotations from domain experts and for deep learning models we need large datasets to reduce over-fitting.

Acting as a potential solution, the paradigm of semi-supervised learning extracts information from both labelled and unlabelled data and reduces the number of labels needed for training. This thesis deals with the development of novel classical and deep machine learning approaches for semi-supervised image classification. Our approaches are centred around graph-based learning, and we apply them to a range of real-world problems including hyperspectral, natural and medical imaging.

Firstly, we propose and design a superpixel contracted semi-supervised learning framework to classify hyperspectral images. This approach is built around the $p = 2$ graph Laplacian and uses over-segmentation to greatly reduce the size of the graph as well as providing a regularizing prior. Secondly, we combine graph based semi-supervised learning with deep neural networks and re-examine modern data ablation to create a state-of-the-art framework for natural image classification. Finally, we combine graph-based approaches, optimising the more demanding $p = 1$ graph Laplacian, with deep neural networks architectures and apply it to the field of medical imaging. We design a general framework for diagnosis and apply it to chest X-rays, including the diagnosis of COVID-19. For all the approaches in the paper, we show, through rigorous experimental and detailed ablation studies, that our models produce state-of-the-art results and are competitive with fully supervised models whilst only using a fraction of the available labels.

Overall, the contributions of this thesis are focused on the design and implementation of new graph-based semi-supervised frameworks for image classification, which include geometrical and data constraints along with deep neural-networks. Highlighting the power of semi-supervised learning to overcome the need for costly labelled datasets.

Table of contents

List of figures

List of tables

Glossary

AA	Average accuracy
AUC	Area under curve
AVIRIS	Airborne visible/infrared imaging spectrometer
CNN	Convolutional neural network
CT	Computerised tomography
CXR	Chest X-ray
DAG	Density aware graphs
DL	Machine learning
DSSL	Deep semi-supervised learning
EMA	Exponential moving average
EPF	Edge preserving filter
ERS	Entropy rate superpixels
ERW	Extended random walker
FE	Feature extraction
GAN	Generative adversarial networks
GCN	Graphical convultional networks
GPU	Graphics processing unit
GT	Ground truth
HMS	Hyper manifold simple linear iterative clustering
HSI	Hyperspectral image
ICT	Interpolation consistency training
IFRF	Image fusion and recursive filtering
LBP	Local binary patterns
LCMR	Local co-variance matrix representation
LED	Log-Euclidian distance
LGC	Local and global consistency
LPDSSL	Label propogation for deep semi-supervised learning
MC	Monte carlo
MKL	Multiple kernel learning

ML	Machine learning
MSLIC	Manifold simple linear iterative clustering
MT	Mean teacher
OA	Overall accuracy
PCA	Principal component analysis
RBF	Radial basis function
ROSIS	Reflective optics system imaging spectrometer
SCMK	Superpixel-based classification via multiple kernels
SGL	Superpixel graph learning
SL	Supervised learning
SLIC	Simple linear iterative clustering
SNTG	Smooth neighbors on teacher graphs
SOTA	State of the art
SSL	Semi supervised learning
SVM	Support vector machine
SWA	Stochastic weight averaging
TSSDL	Transductive semi-supervised deep learning
TSVM	Transductive support vector machine
UDA	Unsupervised data augmentation
UL	Unsupervised learning
VAT	Virtual adversarial training
WRN	Wide residual network

Chapter 1

Introduction

Coinciding with advances in modern technology, we as a society produce and consume increasing amounts of data. In particular, the amount of imaging data we interact with has skyrocketed. This does not only include handheld cameras but other complex domains such as hyperspectral imaging which consists of a range of wavelengths from 400 - 1100 nm, medical imaging and a variety of sensor types. Due to the volume of data produced, the bottleneck in extracting the contained information is the time is takes us as humans to analyse it. Automating the analysis of this imaging data would have huge beneficial impacts for our quality of life.

In this thesis, we investigate the automation of a fundamental image analysis task: *image classification*, where we seek to assign a set of labels to an image that classifies its content. Given its fundamental nature, image classification is widespread in computer vision and success in this area of research would be a breakthrough to the large-scale analysis of imaging data. However, due to the complexity of image classification, typical algorithmic approaches which have worked so well in computer science [56] completely fail in producing tractable solutions. Instead, the community have attempted to imitate the learning process of humans motivated by the ease at which we handle imaging data.

The framework through which the majority of this research has been undertaken is that of *classical machine learning*. Machine learning approaches seek to tackle the problem of image classification by learning the algorithm from data provided to them [153], much like we do growing up. A commonality of these classical machine learning methods is that the representation learning of the data is kept separate from the classification task. Given the scale of the research produced in tackling image classification, there have been a multitude of proposed classical methods and there exist long and detailed surveys which give a general overview of this diversity [143, 158]. There are traditionally two

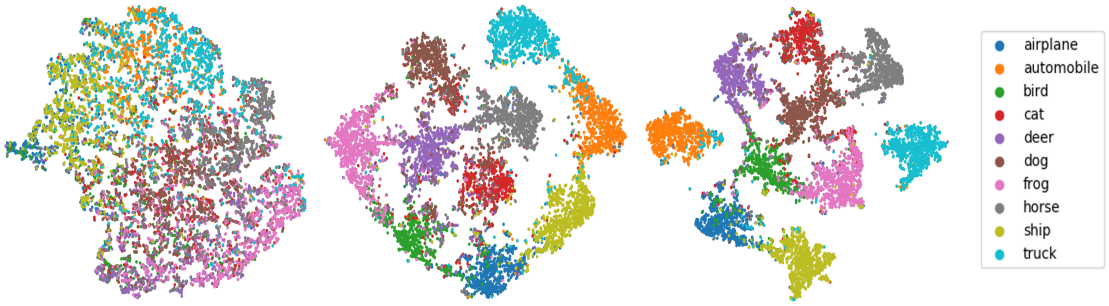


Fig. 1.1 In this figure we show how the 2D feature representation, produced via t-SNE, of the CIFAR-10 dataset [188] changes as a deep learning model is trained for image classification. We visualise the extracted representation at the first, middle and last epoch of training from left to right respectively. We colour each image with its class label and we see that during training classes are separated from their initial muddled mixing.

fundamentally different learning paradigms which we use to approach the problem of image classification in machine learning: *supervised* and *unsupervised* learning.

In *supervised* learning there exists a joint distribution $\mathscr{X} \times \mathscr{Y}$ from which we have n data samples, typically assumed to be sampled i.i.d, given by $\{x_i, y_i\}_{i=1}^n$ with y_i being referred to as labels or targets. The task is to learn a mapping from $f : \mathscr{X} \to \mathscr{Y}$ so that we can predict the target y for unseen data inputs. In the case that y takes values in a finite set the task is referred to as classification. Examples of commonly used techniques to solve the supervised learning problem include parametric methods such as support vector machines [206] or non-parametric approaches such as k-nearest neighbours [167, 121] or random forests [163]. For *unsupervised* learning, we are given n samples $\{x_i\}_{i=1}^n$ from some distribution $\mathscr{X}$, which we again assume to be sampled i.i.d, and our task is to find f using $\{x_i\}_{i=1}^n$ alone. Examples of common unsupervised algorithms are clustering methods [89, 135], dimensionality reduction [212, 233] and outlier detection [37].

Despite the success of classical machine learning methods in image classification, and other computer vision tasks such as image denoising and image segmentation, many classical machine learning approaches have been left behind since the rise of *deep learning*. For both supervised and unsupervised machine learning there is a question on how the images are represented in feature space. This general problem is referred to as representation learning [21]. For classical methods the feature representations are often handcrafted by domain experts or obtained using unsupervised learning algorithms. However, these representations generalised poorly to unseen data and were a major bottleneck in performance. Deep learning methods sought to solve the task of representation learning and

classical machine learning in an end-to-end fashion, learning high-level feature representations during training and eliminating the need for hand-crafted features. We present an example of this process in Fig 1.1.

Deep learning methods typically utilise a neural network architecture [20] composed of a stack of feature-extraction layers with free parameters, such as the popular convolutional neural network architecture [5]. The network parameters are trained using first order methods, such as stochastic gradient descent [26], so that the generated features are useful for the task and domain at hand. With the development of novel optimisation techniques and advances in hardware, we are now able to effectively train deep neural networks that contain hundreds of millions of parameters [204] thereby producing complex feature representations which have revolutionised performance in tasks such as image classification [197, 97]. A key example being the famous work of Krizhevsky [127] which used a large-scale deep learning model to significantly improve the state-of-the-art on the ImageNet classification dataset.

1.1 Semi-Supervised Learning

However, despite the outstanding performance of deep supervised techniques, these methods often rely upon access to a large and extensive labelled dataset. In many domains the collection of labelled data is time consuming, expensive and may require expert knowledge, such as in the case of medical imaging. This issue of data collection has been magnified with the development of deep learning as models are often data hungry and susceptible to simply memorising the data rather than learning [244]. Increasing the requirement of labelled data represents a significant barrier to the deployment of deep learning methods in label starved domains. Whilst some techniques have deployed fully unsupervised approaches for image classification to remove the need for labels altogether, the lack of correspondence between the produced clusters and the original classes often makes the problem ill-posed and as yet the results are significantly short of that of supervised learning [252, 160].

In this thesis we focus on the development and implementation of semi-supervised (SSL) approaches for image classification. Semi-supervised learning can be seen as a half way point between supervised and unsupervised learning [44] . In semi-supervised learning a portion of the data is labelled $\{x_i, y_i\}_{i=1}^{n_l}$, where y denotes label. The vast majority of the data is unlabelled $\{x_i\}_{i=n_l+1}^{n_l+n_u}$, with $n_u \gg n_l$. In the case where the classes are known beforehand, semi-supervised learning can be viewed as supervised learning with extra information known on the distribution of $\mathscr{X}$. A visualisation of this principle is

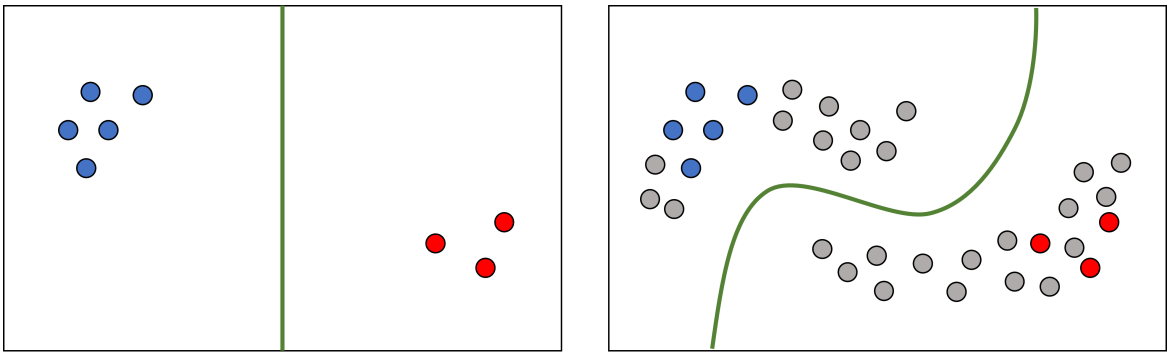


Fig. 1.2 Demonstration of supervised learning (left) against semi-supervised learning (right). Given a binary classification task and some initial labelled data (red and blue points), shown on the left hand side, a reasonable looking decision boundary (green) would be a simple line splitting the two classes. However, if we were given access to a large number of unlabelled samples (grey dots), shown on the right side, we would want our decision boundary to lie in the low-density regions and subsequently improve our generalisation.

shown in Figure 1.2. Many classical approaches have been proposed for semi-supervised learning for both the transductive [253, 249, 221, 115, 98] and inductive [255, 62, 19] settings and we refer readers to [44, 162, 213] for an in-depth analysis and taxonomy. The most common classic semi-supervised learning methods can be split into *low density separation approaches* , *generative approaches* and *graph-based methods*.

Generative models Given some free parameters θ, generative models seek to learn the class conditional density $P(\mathbf{x}|y,\theta)$ and use Bayes rule to compute the decision function $P(y|\mathbf{x},\theta) \propto P(\mathbf{x}|y,\theta)P(y|\theta)$. Incorporating unlabelled data into this framework can be simply done by taking a *missing value approach* and a commonly used algorithm is expectation maximization [154]. In this algorithm we compute $P(y|x_i,\theta)$ for all the unlabelled data examples before maximising over the log-likelihood L of the labelled and unlabelled data.

$$L = \sum_{i=1}^{n_l} \log(P(x_i|y_i,\theta)P(y_i|\theta)) + \sum_{i=n_l+1}^{n_l+n_u} \sum_{y} P(y|x_i,\theta)\log(P(x_i|y,\theta)P(y|\theta)) \tag{1.1}$$

Unfortunately, due to the complexity of modelling the class conditional density and the requirement of strict modelling assumptions, this family of approaches is relatively uncommon.

Low density separation approaches The goal of low-density separation approaches is to find decision boundaries which lie in low density regions with respect to both the labelled and unlabelled data samples, giving the alternate name of margin-max methods.

The most popular low-density method is the transductive support vector machine [49] which maximises the margin on both labelled and unlabelled data with the main drawback being the non-convex optimisation of the problem.

Graph-based methods Graph-based methods construct a graphical representation of the data and represent the labelling of the points as a function defined on the set of nodes. The information from the labelled points is then propagated across the graph using a defined energy functional. Examples include using the graphical Laplacian [248] or more general energies related to Poisson's equation [36]. The major obstacle to implementation is constructing a graph which effectively captures the geometry present in the data.

Despite the success of these classical methods in improving performance with limited labels, semi-supervised methods for image classification greatly benefit from being provided more complex feature representations. Therefore, there has been a large body of research developed in combining semi-supervised learning with neural network architectures [162]. There are a wide variety of approaches, including works which focus on entropy minimisation [24], generative networks [123] and holistic approaches [197] which seek to combine several different principles into a unified deep learning framework. There are two dominant families of approaches in deep semi-supervised learning which shape this thesis: *pseudo-labelling* and *consistency regularisation*.

1. **Pseudo-labelling**: Pseudo-label methods seek to estimate the labels for unlabelled data points. The generated pseudo-labels are then used in combination with the originally labelled points in either a composite batch or composite loss function. In doing so, they increase the amount of information the model has access to. Furthermore, they decrease the prediction entropy on unlabelled data which moves the decision boundaries to low-density regions, one of the fundamental assumptions required in semi-supervised learning [44]. For image classification tasks the labels are typically estimated by the output of the neural network [8, 131, 22], though some graphical techniques have been used [106, 132] as well.

2. **Consistency Regularisation**: Consistency regularisation methods, also named perturbation methods, purely utilise the unlabelled data to move the decision boundaries to lie in low-density regions. They do so by demanding that the classification output should be unchanged to small data perturbations to the data input and the model is optimised to minimise this change. Example perturbations include data augmentation techniques [236], adversarial training [151], perturbations to the model parameters [205] and data interpolation [205].

1.2 Research Contributions

With the creation of deep learning approaches, there has been a rapid increase in the accuracy of semi-supervised classification methods. Despite this progress there are still many open problems and issues with the current state of semi-supervised learning. To summarise the research contributions of this thesis, we firstly highlight how this research tackles some of the open questions in the field. Subsequently, we describe the overall contributions made in this thesis.

1.2.1 Contributions to Open Problems

We now highlight three open problems in the field of semi-supervised learning for image classification and detail the specific contributions made in this thesis towards tackling them.

***Improving Pseudo-label Performance**. In many pseudo-labelling approaches, the produced pseudo-labels are generated directly by the neural network. As pointed out by [8], network generated pseudo-labels fail to solve toy problems and instead rely upon technical tricks to improve their performance [24]. However, given that the basis for many of these tricks are being questioned [91], other more accurate approaches are needed.*

In this thesis, we demonstrate that combining a neural network with graphically produced pseudo-labels results in a far more accurate model than a purely network-based approach. Furthermore, we theoretically justify and implement a multi-sampling approach to data augmentation and show that by combining this with a graph-based energy model we remove the need for several technical tricks whilst outperforming state-of-the-art pseudo-labelling methods.

***The Domain Dependence of Consistency Regularisation:** Consistency regularisation works well when a suitable perturbation can be defined. Doing so is domain dependent and requires expert knowledge to craft. Therefore, the creation of alternative regularisation techniques which are not sensitive to the applied domain can greatly ease the deployment of semi-supervised methods.*

To tackle this problem, we propose a novel cluster based regularisation to directly implement the *cluster assumption* of SSL. Our cluster regularisation is completely domain independent and simple to implement. We demonstrate that our CycleCluster [188] approach outperforms many consistency regularisation approaches and isn't sensitive to the chosen number of clusters.

Applying Semi-supervised Learning to Complex Domains. *Despite the successful deployment of semi-supervised methods in the natural image domain, fewer semi-supervised methods have been implemented for the medical and hyperspectral domains due to the increased domain complexity. On the other hand, in these domains there are more hurdles in obtaining labelled data and as such semi-supervised methods are even more valuable.*

In this thesis we propose novel semi-supervised approaches for transductive learning for hyperspectral image classification [190]. We highlight that by clever adaption to the hyperspectral domain by using a novel hyperspectral over-segmentation algorithm, we can accurately classify entire hyperspectral images by using only a handful of labelled data points. To the best of our knowledge, we also produce one of the first deep semi-supervised frameworks for medical image classification which is built around the graph $1-$Laplacian. We propose both a transductive and inductive framework. We apply our framework to diagnosing chest X-rays, including COVID-19. We demonstrate that we can outperform fully supervised approaches whilst using a fraction of the available labels.

1.2.2 Chapter Contributions

As well as tackling the previously outlined problems, in this thesis we develop and implement new graph-based semi-supervised learning approaches for image classification in a range of imaging domains: hyperspectral, natural and medical. Each of these three domains are discussed in separate chapters. In the following section, we detail the content of each chapter which includes discussing published works and highlighting key contributions.

Chapter 3: Hyperspectral Image Classification

In Chapter 3 we present a research project for hyperspectral image classification that resulted in the publication of [190]. In this project *we proposed a novel approach to the semi-supervised classification of hyperspectral images*. We noticed that typically in graph based methods each pixel is represented as a node which leads to very poor scaling as demonstrated in [38]. To counter this we created a *novel superpixel method for hyperspectral imaging data* from which we can define meaningful local regions. Using these *regions as the nodes of a graph* we were able to vastly decrease the size of the node set and intelligently regularise the final classification map. We then used this graphical representation and the technique of Zhu [248] based around the graphical Laplacian to propagate information from the initially labelled pixels to the entire image. We extensively validated our proposed approach and demonstrate that our *graphical superpixel approach gives*

state of the art results for hyperspectral image classification. The main contributions are as follows.

- A novel content sensitive algorithm for the over-clustering of hyperspectral images. We demonstrate our approach outperforms popular superpixel algorithms currently used in the field.
- We perform label diffusion on a superpixel-contracted graphical representation using the graph $p = 2$ Laplacian which greatly shrinks the node set and allows better scaling to larger images.
- We demonstrate high performance over several hyperspectral benchmarks with only a handful of labels per class. Additionally, we compare and outperform against the current state-of-the-art.

Chapter 4: Natural Image Classification

In Chapter 4 we present the research undertaken in the natural imaging domain. This research culminated in two approaches CycleCluster [188] and LaplaceNet [189] with are under review as of the submission of this thesis. Both approaches share the same base, a neural network iteratively trained with graph-based pseudo-labels but seek to tackle problems in consistency regularisation and pseudo-labelling respectively.

In CycleCluster *we propose a novel approach to semi-supervised regularisation by directly implementing the cluster assumption* rather than relying upon data perturbations which are domain specific and technically complex. Our proposed clustering regularisation is used in sequence with a pseudo-label loss whilst being domain independent and simple to implement. We demonstrate through benchmark datasets that *our approach outperforms several perturbation based approaches* and that our method is not sensitive to the initial choice of the number of clusters.

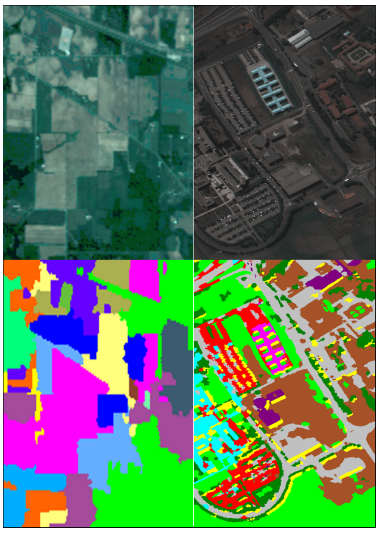
In addition, we aimed to produce state-of-the-art results in semi-supervised natural image classification whilst greatly reducing model complexity and eliminating technical tricks, which culminated in LaplaceNet. We mathematically investigate the use of data augmentation and *theoretically motivate and justify an augmentation averaging approach.* We combine this strategy with graph-based pseudo-labelling and a *simple optimisation scheme* across several benchmark dataset. Through detailed experiments we highlight the advantage of *using graphical pseudo-labels over the standard network based* predictions and experimentally validate the *theoretical predictions of augmentation averaging.* The main contributions are as follows.

- A novel clustering based regularisation for deep neural networks which is both simple to implement and not sensitive to the number of clusters used.
- We mathematically investigate data augmentation and propose a multi-augmentation approach which we experimentally show improves generalisation.
- A state-of-the-art model for natural image classification, LaplaceNet. We demonstrate our graphical pseudo-label approach outperforms the current state-of-the-art.

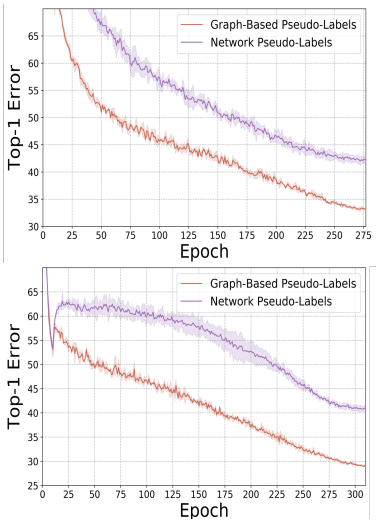
Chapter 5: Medical Image Classification

In Chapter 5 we provide the research we undertook in the field of medical imaging. This resulted in the publications of [14, 15] with collaborators Nicolas Papadakis, Ruoteng Li, Qingnan Fan and Robby T Tan. *Medical imaging uniquely suffers from a lack of labelled data* due to the domain expertise needed to annotate medical images. However, as of the time of submission, there has only been a small amount of research undertaken to investigate the role that semi-supervised learning can play in medical image classification. Therefore, for the first time in the medical domain, *we implement a $p = 1$ graph Laplacian approach for label diffusion.* We do this by *implementing an accelerated primal-dual algorithm originally proposed by Chambolle et al [41].* We demonstrate that this approach is effective for both transductive node classification and for inductive learning by iteratively training a neural network. We apply our transductive model to the classification of chest X-rays and our inductive model to the classification of COVID-19. To the best of our knowledge, *this works represents one of the first deep-semi-supervised methods proposed for medical imaging and the first for COVID-19.* We demonstrate, for both tasks, that *our frameworks outperform leading supervised methods whilst requiring a fraction of the labels.* The main contributions are as follows.

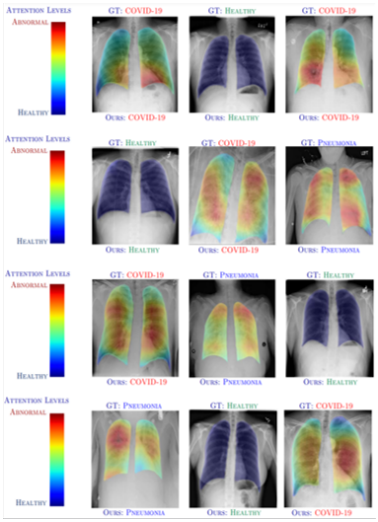
- For the first time in the medical domain, we use the $1-$Laplacian graphical energy and optimise using an accelerated primal-dual algorithm to produce accurate pseudo-labels.
- We propose a novel transductive and inductive framework for medical image classification. To the best of our knowledge, this represents one of the first deep semi-supervised methods proposed for medical image classification.
- Using these frameworks we tackle the task of chest x-ray diagnosis for both general diagnosis and specifically COVID-19. We outperform state-of-the-art supervised methods whilst training on a fraction of the labels.



Chapter 3 *Superpixel Contracted Graph-Based Hyperspectral Image Classification.*



Chapter 4 *Pseudo-Labelling Approaches for Natural Image Classification.*



Chapter 5 *Semi-Supervised Medical Image Classification with $P = 1$ graph Laplacian.*

Fig. 1.3 The research structure of this thesis can be split into works in hyperspectral, natural and medical imaging classification, which are contained in Chapters 3, 4 and 5 respectively. For each domain we design and implement novel algorithms which improve upon current state-of-the-art methods.

1.3 Thesis Overview

The remainder of this thesis is structured in the following way.

In **Chapter 2** we start by introducing the suitable mathematical background to the main topics of this thesis: machine learning (including deep learning), semi-supervised learning and a concise overview of relevant graph theory. We then proceed to the main research topics and present a condensed overview of their structure in Figure 1.3.

In **Chapter 3** we present our novel approach to semi-supervised hyperspectral image classification *Superpixel Graph Learning* (SGL) [190]. After introducing the hyperspectral domain, we give the mathematical preliminaries on the task of image over-segmentation. We then explore the related work surrounding hyperspectral image classification before detailing the methodology of our approach. We then compare our approach against a range of state-of-the-art methods and test the importance of using a domain specific over-segmentation algorithm.

In **Chapter 4** we explore the topic of deep semi-supervised learning for natural image classification and propose two different approaches CycleCluster and LaplaceNet. We firstly introduce natural imaging before providing background information on data

augmentation and a detailed analysis of related work in the field. We describe the methodology of CycleCluster and its novel use of *clustering regularisation* before presenting a detailed experimental comparison to perturbation approaches. We then turn to LaplaceNet and explore the methodology of augmentation averaging and our simple optimisation scheme before experimentally demonstrating state-of-the-art performance. We additionally present detailed ablations on augmentation averaging and graph-based pseudo-labels.

In **Chapter 5** we explore the medical imaging domain. We begin by presenting the mathematical background of the $p = 1$ graph Laplacian before exploring the related work of medical image classification. We then document the methodology of our transductive framework for image classification which we used to produce the GraphX$^{\text{Net}}$ model. We then expand our work to the inductive GraphX$^{\text{COVID}}$ which we used to diagnose COVID-19 via chest X-rays. Through several experiments on the leading benchmark, we demonstrate that both our frameworks outperform the leading supervised algorithms using a fraction of the labels.

Finally, we conclude the thesis in **Chapter 6** by summarising our contributions and focusing on two areas of further work, strong augmentation and graphical representations, which I belief to be fundamental to the development of graph-based semi-supervised learning in the imaging domain.

1.4 Published Work

The work presented in this thesis led to the publication of several research papers which we list here.

1. *Semi-supervised Learning with Graphs: Covariance Based Superpixels For Hyperspectral Image Classification*: Philip Sellars, Angelica Aviles-Rivero, Nicolas Papadakis, David Coomes, Anita Faul and Carola-Bibiane Schönlieb. IEEE International Symposium on Geoscience and Remote Sensing, 2019.

2. *Superpixel Contracted Graph-Based Learning for Hyperspectral Image Classification*: Philip Sellars, Angelica Aviles-Rivero and Carola-Bibiane Schönlieb. IEEE Transactions on Geoscience and Remote Sensing, Volume 58, 2020.

3. *CycleCluster: Modernising Clustering Regularisation for Deep Semi-Supervised Classification* Philip Sellars, Angelica Aviles-Rivero and Carola-Bibiane Schönlieb. Under review. Preprint available at arXiv:2001.05317, 2020.

4. *LaplaceNet: A Hybrid Energy-Neural Model for Deep Semi-Supervised Classification*: Philip Sellars, Angelica, Aviles-Rivero, and Carola-Bibiane Schönlieb. Under review. Preprint available at arXiv:2106.04527, 2021.

5. *GraphX-NET - Chest X-Ray Classification Under Extreme Minimal Supervision*: Angelica Aviles-Rivero, Nicolas Papadakis, Ruoteng Li, Philip Sellars, Qingnan Fan, Robby T. Tan and Carola-Bibiane Schönlieb. International Conference on Medical Image Computing and Computer-Assisted Intervention, 2019.

6. *GraphXCOVID: Explainable Deep Graph Diffusion Pseudo-Labelling for Identifying COVID-19 on Chest X-rays*: Angelica Aviles-Rivero, Philip Sellars, Carola-Bibiane Schönlieb and Nicolas Papadakis. Pattern Recognition, Volume 122, 2022.

Chapter 2

Preliminaries

In this chapter, we introduce the mathematical background to important topics which are necessary for understanding later research chapters. We assume that the reader is familiar with basic concepts of mathematical analysis and linear algebra. We begin by exploring the fundamentals of graph based semi-supervised learning before covering the mathematics of deep learning where we describe the architecture and optimisation of neural network architectures. For a large-scale in-depth review of both semi-supervised and deep learning we refer readers to the works of [44, 87] respectively.

2.1 Semi-Supervised Learning

Semi-supervised learning (SSL) is often viewed as being in-between supervised and unsupervised learning as some but not all points are labelled. Formally, we have a labelled set $Z_l = \{x_i, y_i\}_{i=1}^{n_l}$ of n_l joint data pairs, assumed to be sampled i.i.d, from some joint data distribution $\mathscr{X} \times \mathscr{Y}$. We refer to $X_l = \{x_i\}_{i=1}^{n_l}$ as samples and $Y_l = \{y_i\}_{i=1}^{n_l}$ as labels. Additionally we have a unlabelled set $X_u = \{x_i\}_{i=n_l+1}^{n_u+n_l}$ drawn from the same distribution $\mathscr{X}$. Note that typically $n_u >> n_l$. In the case of classification, we are then tasked with either a *transductive* or *inductive* learning problem. In the transductive case, we seek to learn a mapping $f : \mathscr{X} \to \mathscr{Y}$ which accurately predicts the labels for a finite set of points, such as X_u. On the other hand, in the inductive case we seek to learn a mapping $f : \mathscr{X} \to \mathscr{Y}$ which accurately predicts the label for a potentially infinite set of samples from the same distribution $\mathscr{X}$

For semi-supervised learning to be a useful concept, the information we extract from the unlabelled set X_u must be useful in obtaining the mapping f. In other words, the underlying marginal data distribution $p(x)$ over the data space must be useful in estimating the posterior distribution $p(y|x)$ [44]. If this condition is not met, then it is impossible for

the unlabelled data to help and we would be better of using purely supervised algorithms. For this to happen we must make certain assumptions about the underlying joint distribution $\mathscr{X} \times \mathscr{Y}$. These assumptions are fundamental to semi-supervised learning and every semi-supervised algorithm, including modern deep learning approaches, rely upon at least one of them.

Whilst there are ways to formalise the following assumptions in a probably approximately correct (PAC) framework [44] we find it far more intuitive to consider them in their informal notation. We begin with perhaps the most common assumption of SSL.

Definition 2.1. The Smoothness Assumption If two points x_1, x_2 are in a high density region and are close then their corresponding outputs y_1, y_2 should also be close. Note that high-density and low-density regions refer to the density $P(x)$ [125].

Whilst a simpler version of this assumption is fundamental to supervised learning, in the semi-supervised setting we have the added benefit of incorporating the unlabelled data into this framework. For example, if two points are connected by a path through a high density region then we know that all points on that path, whether they be unlabelled or labelled, should have similar outputs. The reason for specifying *high density regions* is that low-density regions often signify a jump between data structures and as such we would not naturally expect outputs to be similar across structures.

Definition 2.2. The Cluster Assumption If points are in the same cluster, they are likely to be of the same class.

This assumption speaks to some underlying structure of the data space $\mathscr{X}$ with areas of high and low density. However, given that the existence of classes themselves is dependent upon this structure it is reasonable to assume its existence. Note that with this assumption, we are not expecting that each class is represented by a single cluster but that each cluster is unlikely to contain multiple classes. The *cluster assumption* is a specific realisation of the *smoothness assumption*. Commonly in the SSL literature, the cluster assumption is talked about in a equivalent density assumption.

Definition 2.3. The Low Density Separation Assumption The decision boundaries should lie in low-density regions.

Definitions 2.2 and 2.3 can be seen to be equivalent as if the decision boundaries lie in low density regions then points in the same cluster are likely to be of the same class. Thus, high density regions, such as clusters, are assigned the same label. Despite the conceptually equivalence of these assumptions, they lead to vastly different implementations,

especially when dealing with deep semi-supervised learning. We further research these implementation differences in Chapter 4. Finally, we present the manifold assumption, where we define a manifold to be a topological space that is locally Euclidean.

Definition 2.4. The Manifold Assumption The high-dimensional data space $\mathscr{X}$ can be approximated as lying on a low-dimensional manifold $\mathscr{M}$.

The immediate benefit of such an assumption is that we can work in a lower dimensional feature space and avoid problems relating to the curse of dimensionality [212]. Furthermore, our prior assumptions, such as the smoothness and low-density assumption, equally apply to the surface of the manifold as they did to the original data space.

All classical and deep semi-supervised learning algorithms can be traced back to these four fundamental assumptions. However, each of these assumptions can be implemented in a variety of ways. As one example, the low-density separation assumption leads to works based around entropy minimisation [88], pseudo-labelling [131] and consistency regularisation [197]. In this thesis, we exclusively work with graph-based approaches to semi-supervised learning. Thus, we provide a concise background into graphical representations that acts as a basis for discussing our proposed techniques in later chapters.

2.2 Graphical Representations

In this section we first build up a basic introduction to graph theory where we examine necessary relevant concepts. We then examine how graphical methods are used in semi-supervised learning.

2.2.1 Basic Definitions

In graph theory, a *graph* $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ is a structure which captures the geometry of a set of objects. The node set $\mathscr{V} = \{v_1, .., v_n\}$ $|\mathscr{V}| = n$ represents the objects and each element $v \in \mathscr{V}$ is referred to as a *node*. The set $\mathscr{E} = \{e_{ij}\}$ $|\mathscr{E}| = m$ contains the connections between the points and each element $e_{ij} \in \mathscr{E}$ is the *edge* between nodes $v_i, v_j \in \mathscr{V}$. As a first point of call we define common types of graphs. In this thesis we exclusively work with *undirected simple* graphs and define each of these terms in order.

Definition 2.5. Undirected graphs: In undirected graphs the edges have no order such that if $e_{ij} \in \mathscr{E}$ then node i is connected to node j and node j is connected to node i.

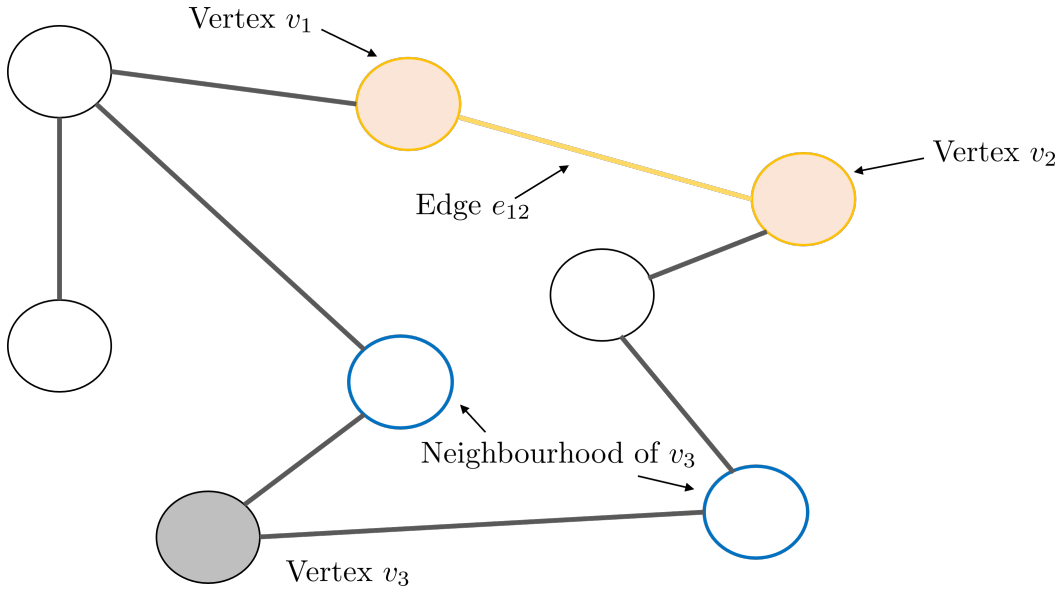


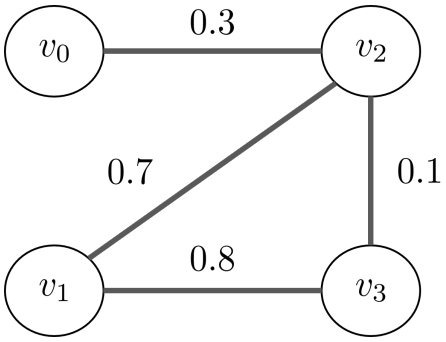
Fig. 2.1 An example of a simple undirected graph $\mathcal{G}$. Each circle represents a node and each line represents an edge connecting two nodes. Some nodes and edges are highlighted to provide examples of these concepts.

Definition 2.6. Simple graphs: Simple graphs are graphs which do not have more than one edge between any two vertices and additionally have no self-loops. These two conditions can be formalised in the following respective ways. For each edge $e_{ij} \in \mathcal{E}$ there is no other edge $e_{i'j'} \in \mathcal{E}$ such that $i' = i, j' = j$. Furthermore, for each edge $e_{ij} \in \mathcal{E}$ $i \neq j$.

We provide an example of a simple undirected graph in Fig 2.1. In the context of semi-supervised learning we often expand our graphs to include edge weightings. Each edge is assigned a weight $w \in \mathbb{R}$ and these weights are stored in the set $\mathcal{W} = \{w_{ij}\}$ where w_{ij} is the edge weight between nodes v_i and v_j. Thus producing weighted undirected simple graphs $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$.

Definition 2.7. Weighted graphs: In a weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ each edge $e_{ij} \in \mathcal{E}$ is assigned a weight $w_{ij} \in \mathcal{W}$. This weight information is stored in the weight set $\mathcal{W}$.

Note that in this work we assume that $w_{ij} \geq 0$. The weights in a weighted graph may reflect some real quantity such as distance or cost but in our case the weight reflects the similarity between different objects with larger weights indicating more similar objects.



$$W = \begin{bmatrix} 0 & 0 & 0.3 & 0 \\ 0 & 0 & 0.7 & 0.8 \\ 0.3 & 0.7 & 0 & 0.1 \\ 0 & 0.8 & 0.1 & 0 \end{bmatrix}$$

Fig. 2.2 An example of a weighted simple undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ and its associated adjacency matrix W. Notice that for simple undirected graphs the adjacency matrix is symmetric.

2.2.2 Graph Operators

Consider the space of real functions defined on a graph's nodes $f : \mathcal{V} \to \mathbb{R}$. Such functions assign a real number to each node of the graph and we write $\vec{f} = (f(v_1), f(v_2), .., f(v_n))$. Many important topics in graph theory involve operators, which we denote by **A**, acting on this space of functions. For these operators **A** the eigenvectors $\mathbf{A}\vec{x} = \lambda\vec{x}$ are also functions on the nodes. The first operator we will consider is the adjacency matrix.

Adjacency Matrices

In graph theory, the *adjacency matrix* W is a square matrix $W \in \mathbb{R}^{n\times n}$ which contains the connectivity information of a graph. For a weighted undirected simple graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ the entries of W are determined by

$$W_{ij} = \begin{cases} w_{ij} & \text{if } e_{ij}, \in \mathcal{E} \text{ or } e_{ji} \in \mathcal{E} \\ 0, & \text{otherwise.} \end{cases} \tag{2.1}$$

Note that for undirected graphs the matrix W is always symmetric and for simple graphs the diagonal is populated with zeros. An example of a weighted simple undirected graph and its corresponding adjacency matrix is shown in Fig 2.2. For the adjacency matrix acting on a function $\vec{g} = W\vec{f}$ we have that $g(v_i) = \sum_j w_{ij} f(v_j)$ and a corresponding quadratic form of $\vec{f}^T W \vec{f} = \sum_{i,j} f(v_i) w_{ij} f(v_j)$. Using the adjacency matrix we are able to succinctly define the *degree* of each node.

Definition 2.8. Degree: The degree d of node v_i is given by $d_i = \sum_j w_{ij}$ and represents the overall connectedness of a node within the graph.

The node degrees can be used to create the degree matrix $D \in \mathbb{R}^{n\times n}$ whose entries are determined by

$$D_{ij} = \begin{cases} d_i & \text{if } i = j, \\ 0, & \text{otherwise.} \end{cases} \tag{2.2}$$

From the weight matrix W and degree matrix D we are able to construct arguably the most important operator for graphical structures, the *graphical Laplacian*.

The Graph Laplacian

The graph Laplacian, alternatively named the discrete Laplacian, is used in many graph-related applications and can be used to calculate the number of spanning trees, approximate sparsest cuts and is extensively used in spectral graph theory [218]. The unnormalised graph Laplacian L is defined as

$$L = D - W, \tag{2.3}$$

where the elements of L are given by

$$L_{ij} = \begin{cases} d_i & \text{if } i = j, \\ -w_{ij} & \text{if } e_{ij} \text{ or } e_{ji} \in \mathscr{E} \\ 0, & \text{otherwise.} \end{cases} \tag{2.4}$$

There are several remarkable properties of the unnormalised graph Laplacian and we remark upon a few relevant ones and refer readers to the work of [218] for detailed proofs.

Proposition 2.9. ***Properties of the unnormalised graph Laplacian*** *L The matrix L of the graph $\mathscr{G} = (\mathscr{V}, \mathscr{E}, \mathscr{W})$ satisfies the following properties.*

1. *For every function f defined on the nodes of the graph $\mathscr{G}$ we have that*

$$(Lf)(v_i) = \sum_j w_{ij}(f(v_i) - f(v_j)). \tag{2.5}$$

$$\vec{f}^T L\vec{f} = \frac{1}{2}\sum_{i,j=1}^{n} w_{ij}(f(v_i) - f(v_j))^2. \tag{2.6}$$

2. *L is symmetric and positive semi-definite.*

3. *The eigenvalues of L are non-negative and real valued with* $0 = \lambda_1 \leq \lambda_2 ... \leq \lambda_n$. *The eigenvector of* λ_1 *is equal to* $\mathbb{1}_n$.

In the literature, there exist two normalisations of the graph Laplacian which aim to reduce the influence of nodes within the graph that have large degrees. Both of these normalisations are connected but differ in their applications. From the Markov chain perspective, the underlying transition matrix is given by $D^{-1}W$ and this leads to the *random walk* normalised Laplacian $\mathscr{L}_{rw} = I - D^{-1}W$. The other normalisation of the Laplacian matrix is the symmetric graph Laplacian $\mathscr{L}_{sym} = I - D^{-1/2}WD^{-1/2}$. There are several connections between these two different normalisations and we provide a fundamental one.

Proposition 2.10. ***Connection between normalised Laplacians:*** λ *is an eigenvalue of* $\mathscr{L}_{rw}$ *with a corresponding eigenvector* u *if and only if* λ *is an eigenvalue of* $\mathscr{L}_{sym}$ *with eigenvector* $w = D^{1/2}u$.

Proof: Given that

$$(I - D^{-1/2}WD^{-1/2})D^{1/2}u = \lambda D^{1/2}u,$$

we can multiply the lhs by $D^{-1/2}$ to give

$$D^{-1/2}(I - D^{-1/2}WD^{-1/2})D^{1/2}u = \lambda D^{-1/2}D^{1/2}u,$$

$$(I - D^{-1}W)u = \lambda u \qquad \square.$$

In our implementations of graphical learning we exclusively work with the symmetric graph Laplacian and for notational conciseness we write $\mathscr{L}_{sym} = \mathscr{L}$. The normalised Laplacian has its own properties which are immensely valuable for semi-supervised learning and we list these below.

Proposition 2.11. ***Properties of the normalised graph Laplacian*** $\mathscr{L}$ *The matrix* $\mathscr{L}$ *of the graph* $\mathscr{G} = (\mathscr{N}, \mathscr{E}, \mathscr{W})$ *satisfies the following properties.*

1. *For every function* f *defined on the nodes of the graph* $\mathscr{G}$ *we have that*

$$\vec{f}^T \mathscr{L} \vec{f} = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} \left(\frac{f(v_i)}{d_i^{1/2}} - \frac{f(v_j)}{d_j^{1/2}} \right)^2. \tag{2.7}$$

2. $\mathscr{L}$ *is symmetric and positive semi-definite and has non-negative real eigenvalues* $0 = \lambda_1 \leq \lambda_2 ... \leq \lambda_n$.

The Graph P-Laplacian

Given (2.6), we may ask if there is an operator L_p, for $p \geq 1$ whose quadratic form, for a function f, is

$$\vec{f}^T L_p \vec{f} = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} |f(v_i) - f(v_j)|^p. \tag{2.8}$$

From research, such as the work of Amghibech [6], the answer is yes and this operator is known as the graph p-Laplacian. Furthermore we have that for this operator,

$$(L_p f)(v_i) = \sum_j w_{ij} \psi_p(f(v_i) - f(v_j)), \tag{2.9}$$

where $\psi_p : \mathbb{R} \to \mathbb{R}$ and is defined as $\psi_p(x) = |x|^{p-1}\text{sign}(x)$. When we subsitute $p = 2$ we see that $\psi_2 = x$ and we recover the original graph Laplacian, leading to the original graph Laplacian often being referred to as $p = 2$ Laplacian. We can also degree normalise the graph p-Laplacian to produce the normalised graph p-Laplacian $\mathscr{L}_p$ and in this work we choose the symmetric normalisation. The quadratic form of $\mathscr{L}_p$ is given by

$$\vec{f}^T \mathscr{L}_p \vec{f} = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} \left(\frac{f(v_i)}{d_i^{1/p}} - \frac{f(v_j)}{d_j^{1/p}} \right)^p. \tag{2.10}$$

2.2.3 Graph Based Semi-Supervised Learning

For image classification, we are in the situation that whilst we can extract complex feature representations of the points, the data is not intrinsically graphically structured. Therefore, if we want to use graph-based semi-supervised learning we need to construct a graph to capture the geometry of the data. This process of transductive learning on graphs is generally composed of three steps: choosing a function or kernel for *estimating the similarities* (affinities) between nodes, *finding a sparse subgraph* from the fully connected weighted graph and finally choosing an algorithm to *propagate the information* from the labelled nodes to the whole graph.

Similarity Estimation

Given that we are constructing the graph from non-graphical data, the edge weights have no physical meaning but instead we design them to signify the similarity between objects with larger weights demonstrating a stronger similarity. For the purpose of semi-supervised learning we typically restrict ourselves to working with positive weights so

that $w_{ij} \geq 0 \; \forall i, j$. For each node v_i we are able to extract a feature representation from the corresponding image x_i which we denote by $t(x_i)$ for some function $t : \mathscr{X} \to \mathbb{R}^d$. The function t is typically a trained neural network model or a handcrafted feature representation. Typically [106, 190] the edge weights are then calculated by using a kernel function $K : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ [79]. A very standard choice is the Radial Basis Function (RBF) kernel which is given by

$$w_{ij} = \exp\left(\frac{-||t(x_i) - t(x_j)||^2}{2\sigma^2}\right), \tag{2.11}$$

where $||\cdot||^2$ is the squared Euclidean distance and $\sigma \in \mathbb{R}^+$ is a free parameter.

Edge Selection

Due to both memory requirements and performance concerns, such as robustness to outliers, graphical methods for semi-supervised learning do not typically use complete graphs, which have fully non-zero adjacency matrices.

Definition 2.12. Complete graphs: A complete graph is a simple undirected graph with the additional condition that for every pair of vertices $v_i, v_j \in \mathscr{V}$ $e_{ij} \in \mathscr{E}$.

Therefore, graph construction methods seek to extract a subgraph $\mathscr{G}' = (\mathscr{V}, \mathscr{E}', \mathscr{W}')$ from the original complete graph $\mathscr{G} = (\mathscr{V}, \mathscr{E}, \mathscr{W})$ which is much sparser in terms of the size of the edge set $|\mathscr{E}'| << |\mathscr{E}|$. Whilst many different approaches have been used [222, 109, 248], the most common approaches are $k-$nearest neighbours and ϵ-neighbourhood which we will cover now.

Definition 2.13. $k-$**Nearest neighbours**: This algorithm keeps an edge if it connects a node to one of its k most similar neighbours. Therefore, we have that $e_{ij} \in \mathscr{E}'$ if and only if the size of the set $\{m | w_{im} > w_{ij} \;\; \& \;\; m \neq j\}$ is less than k. Note that we are assuming that the weights are distinct in value.

Definition 2.14. ϵ**-neighbourhood**: This algorithm keeps the edge between two nodes v_i, v_j if and only if $w_{ij} \geq \epsilon$.

From these edge selection rules we obtain a reduced edge set $\mathscr{E}'$ which results in a sparse weight matrix W. However, both k-nearest neighbours and ϵ-neighbourhoods produce graphs with a large variability in node degree and do not produce symmetric weight matrices without post-fixing. This downside is outweighed by the fact both are computationally cheap compared to more complex methods such as β-matching [109].

Due to ϵ-neighbourhood's heavy sensitive on the ϵ-parameter, $k-$nearest neighbours is typically much easier to apply and in this thesis we select $k-$nearest neighbours for all our graphical approaches.

Label Diffusion

The final task in transductive graph-based semi-supervised learning is to propagate the label information from the small number of initially labelled points across the entire graph. Firstly, we need to capture the initial label information. Therefore, we define a label matrix $Y \in \mathbb{R}^{n\times C}$, where we have n nodes and C classes, whose elements are given by the following rule.

$$Y_{ij} = \begin{cases} 1 & \text{if } y_i = j, \\ 0, & \text{otherwise.} \end{cases} \tag{2.12}$$

where we indicate by 1 the label of a node. For nodes v_i for which no label was provided the row Y_i remains empty. This label information is diffused on the graph by optimizing a predefined energy function which connects the nodes. Although many different energy functions have been used [36, 25, 109], in this work we focus on the graph p-Laplacian energy due to the large body of research that has shown the graph p-Laplacian is an effective general tool for semi-supervised learning [13, 156, 9, 248]. Whilst we leave the details on optimisation for later research chapters, the general mathematics of p-Laplacian approaches is as follows.

Let $\mathscr{F}$ denote the set of $n \times C$ matrices with non-negative entries. For normalised graph p-Laplacian methods, the final labelling matrix is given by $F^* = \text{argmin}_{F\in\mathscr{F}} F^T \mathscr{L}_p F$ where

$$F^T \mathscr{L}_p F = \frac{1}{2}\sum_{i,j=1}^{n} W_{ij} \left|\left| \frac{F_i}{{D_{ii}}^{1/p}} - \frac{F_j}{{D_{jj}}^{1/p}} \right|\right|^p. \tag{2.13}$$

It is common to add a data fidelity term to this energy function to encourage the final predictions to match the initial label information. As an example, in the work of [248] the authors proposed an energy term of the form

$$Q(F) = \frac{1}{2}\sum_{i,j=1}^{n} W_{ij} \left|\left| \frac{F_i}{{D_{ii}}^{1/p}} - \frac{F_j}{{D_{jj}}^{1/p}} \right|\right|^p + \mu\frac{1}{2}\sum_{i=1}^{n} ||F_i - Y_i||^2, \tag{2.14}$$

where $\mu \in (0,1)$ balances between the quadratic form of the graphical Laplacian and the data fidelity term. In the special case that $p = 2$, the above $Q(F)$ has a closed form solution given by

$$F^* = \beta\left(I - \alpha D^{-\frac{1}{2}} W D^{-\frac{1}{2}}\right) Y, \tag{2.15}$$

where $\beta = \frac{\mu}{1+\mu}$ and $\alpha = 1 - \beta$. Note that the final label prediction for an unlabelled point x_i can either be a distribution $y_i = F_i$ or a single class label $y_i = \text{argmax}_j F^*_{ij}$.

2.3 Deep Learning

In this thesis, we make extensive use of deep learning approaches for machine learning. We train a model to learn an effective feature representation and perform the task of image classification simultaneously. Therefore, in this section we provide a mathematical introduction to the architecture and optimisation of deep learning models.

2.3.1 Neural Networks

For the task of image classification a deep learning model is a *trainable* mapping $f_\theta : \mathscr{X} \rightarrow \mathscr{Y}$ from the image space to the label space and is defined by free parameters $\theta = \{\theta_1, ..., \theta_p\}$. The fundamental building block of these mappings is the perceptron, otherwise known as the neuron [155].

The Perceptron

Inspired by the behaviour of neurons in the brain, the perceptron (artificial neuron), which we visual illustrate in Fig 2.3, has become the building block of deep learning models.

A perceptron takes in an input vector $\vec{x} \in \mathbb{R}^n$ and calculates a weighted linear combination $\sum_{i=1}^n \omega_i x_i$, where the weight vector is denoted by $\vec{w} \in \mathbb{R}^n$. To this weighted combination we add a bias $b \in \mathbb{R}$. Afterwards, this output is generally passed to a *non-linear activation function* $\sigma : \mathbb{R} \rightarrow \mathbb{R}$. Therefore, the output of a single perceptron is given by

$$\sigma\left(\sum_{i=1}^{n} x_i w_i + b\right) \equiv \sigma(\vec{w} \cdot \vec{x} + b). \tag{2.16}$$

However, there are cases where we drop the activation function all together. Whilst many activation functions are used in the field, the two activation functions that we utilise in this thesis are

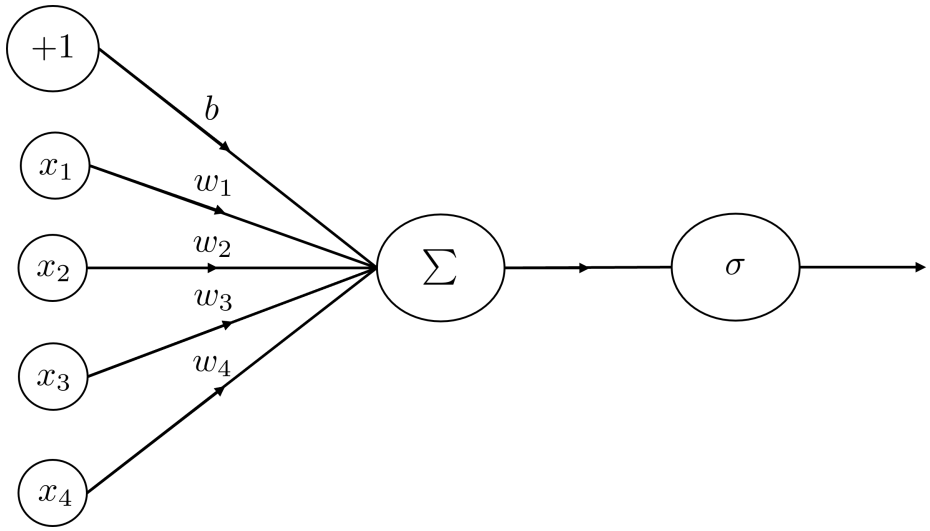


Fig. 2.3 The structure of a perceptron (artificial neuron). A weighted $\vec{w} \in \mathbb{R}^4$ summation of an input vector $\vec{x} \in \mathbb{R}^4$ is combined with the bias b. This result is then passed to an activation function σ.

1. The rectified linear function (ReLU) [86] $\sigma : \mathbb{R} \to \mathbb{R}^+$,

$$\sigma(x) = \max(x, 0). \tag{2.17}$$

2. The sigmoid function $\sigma : \mathbb{R} \to (0, 1)$,

$$\sigma(x) = \frac{1}{1 + \exp(-x)}. \tag{2.18}$$

The trainable mappings f_θ are created by structuring together many groups of these perceptions into a topology or architecture. Therefore, these trainable mappings are referred to as neural networks. Throughout the development of deep learning, many different network architectures, structures and activation functions have been proposed and tested and we refer readers to the work of [138] for a detailed overview of these differing approaches.

In our research, and common for computer vision, we can split our trainable mapping f_θ into the composition of two functions t_θ, g_θ such that $f_\theta(x) = g_\theta(t_\theta(x))$. The function $t_\theta : \mathscr{X} \to \mathbb{R}^d$ is referred to as the feature extraction layers and maps the data to some d-dimensional feature space. The function $g_\theta : \mathbb{R}^d \to \mathbb{R}^C$ is referred to as a classifier and maps from this produced feature space to the label space, where C is the number of classes in the image classification task. Due to their differing tasks, each of these mappings has greatly different network architectures. The classifier g_θ is typically made up of a multi-

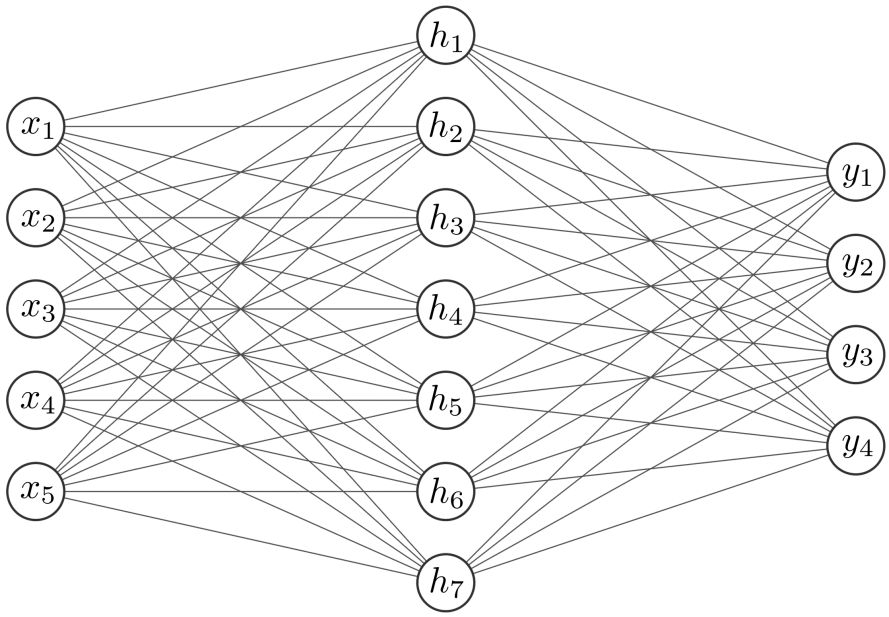


Fig. 2.4 A two-layer perceptron structure. The input layer $\vec{x}$ is outputted to a *hidden* layer with representation $\vec{h}$ before being outputted to the final output layer $\vec{y}$. The output of each layer is fed into the next in a *feed forward* fashion.

layer perceptron, whilst for computer vision the feature extraction layers t_θ are dominated by convolutional operators. Therefore, in the following section we review the mathematics of both multi-layer perceptions and convolutional neural networks.

2.3.2 Multi-layer Perceptrons

A neural network is a feed-forward structure in that the structure is composed of a set of layers and each layer passes its output only to the layer directly after it. In Figure 2.4 we present a simple two layer perceptron as a visualisation and use this structure to introduce the mathematical notation. For the jth neuron in the hidden layer, the output is given by

$$h_j = \sigma_1\left(\sum_{i=1}^{n} x_i w_{ji} + b_j\right) = \sigma\left(\vec{x}\cdot\vec{w}_j + b_j\right), \tag{2.19}$$

where $\vec{x} \in \mathbb{R}^n$ is the input vector, $b_j \in \mathbb{R}$ is the neuron bias, $w_j = \{w_{j1}, .., w_{jn}\} \in \mathbb{R}^n$ are the neuron's weights and σ_1 is the activation function on the first layer. Each neuron in this layer has n weights, equal to the dimension of the input vector. If we define $\vec{h} = \{h_1, .., h_m\}$ to be the vector output of the m neurons in the hidden layer, then we can write

$$\vec{h} = \sigma_1\left(\mathbf{W}_1\vec{x} + \vec{b_1}\right). \tag{2.20}$$

where each row of the matrix $\mathbf{W}_1 \in \mathbb{R}^{n\times m}$ and each element of $\vec{b}_1$ are the weights and bias of each neuron in the first layer. This notation is both elegant and very useful for practically implementing machine learning methods due to its matrix structure. Similarly, for the jth neuron in the second layer, the output is given by

$$y_j = \sigma_2\left(\sum_{i=1}^{m} h_i w_{ji} + b_j\right) = \sigma_2\left(\vec{h}\cdot\vec{w}_j + b_j\right). \tag{2.21}$$

By denoting the output vector $\vec{y} = \{y_1, .., y_k\}$ we have that

$$\vec{y} = \sigma_2\left(\mathbf{W}_2\vec{h} + \vec{b}_2\right), \tag{2.22}$$

where each row of the matrix $\mathbf{W}_2 \in \mathbb{R}^{m\times k}$ and each element of $\vec{b}_2$ are the weights and bias of each neuron in the second layer. Combining equations (2.20) and (2.22) we have that

$$\vec{y} = \sigma_2\left(\mathbf{W}_2\sigma_1\left(\mathbf{W}_1\vec{x} + \vec{b}_1\right) + \vec{b}_2\right). \tag{2.23}$$

It is very easy to generalise this structure to any arbitrary number of hidden layers as each layer i is fully described by a linear activation function σ_i, a weight matrix $\mathbf{W} \in \mathbb{R}^{d_{i-1}\times d_i}$ with d_i being the number of neurons in the ith layer and a bias vector $\vec{b} \in \mathbb{R}^{d_i}$.

2.3.3 Convolutional Neural Networks

Given the large dimensionality of imaging data, using fully connected multi-layer perceptron models for computer vision throws away relevant spatial information and is too computationally expensive. Therefore, another approach was needed. Just as the biological neuron inspired the development of multi-layer perceptrons, the visual cortex inspired a new direction: *convolutional neural networks* [169, 134].

Convolutional neural networks have several advantages over multi-layer perceptrons which make them more suitable for imaging data.

1. Convolutional networks are *equivariant* to image translation [55].

2. The connections between neurons are local in a convolutional network. Each neuron is not connected to each neuron in the previous layer. This greatly reduces the number of parameters and accelerates training.

3. The weight parameters are often shared between connections, again reducing the number of parameters.

4. Through operators which are explained later, a convolutional layer is able to down-sample images which reduces the dimensionality of the data.

Convolutional networks generally consist of two different layers: *convolution layers* and *pooling layers* and we explain both in turn.

Convolution Layers

Convolutional layers are built using the convolutional operator. Informally, a convolution is done by multiplying a local patch of pixels in an image by a tensor kernel which extracts local image information. Formally, given an input tensor $X \in \mathbb{R}^{h \times w \times d}$ of height h, width w and depth d the convolutional operator applies a kernel $K \in \mathbb{R}^{n \times m \times d}$ at all possible places over the input tensor such that we obtain an output tensor $M \in \mathbb{R}^{h \times w}$. Each element M_{ab} of the output tensor is given by

$$M_{ab} = \sigma\left(\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{d} K_{ijk} X_{a+i-1,b+j-1,k} + b\right), \tag{2.24}$$

where σ, b are a non-linear activation function and a kernel-specific bias respectively. Note that the depth of the kernel must match the depth of the input tensor and that the input tensor can be padded on the image boundary to ensure the output and input tensor have the same height and width.

This output tensor represents one extracted visual feature. However, just one feature is insufficient to capture the complex problem at hand. Therefore, we expand our kernel tensor $K \in \mathbb{R}^{n \times m \times d \times c}$ so that we obtain an output tensor $M \in \mathbb{R}^{h \times w \times c}$ which contains c visual features with each one having it's own 3-D kernel. Each pixel is now represented by a vector $M_{ab} \in \mathbb{R}^{c}$ where each element of the vector M_{abe} is given by

$$M_{abe} = \sigma\left(\sum_{i=1}^{n}\sum_{j=1}^{m}\sum_{k=1}^{d} K_{ijke} X_{a+i-1,b+j-1,k} + b_e\right), \tag{2.25}$$

where b_e is the bias belonging to the eth 3-D kernel. This final output tensor is commonly referred to as a *feature map*. This convolution operator leads to the translational equivariance of the convolutional neural network.

Pooling Layers

The feature maps produced by modern convolutional neural networks often contain hundreds of visual features. Unfortunately, [92] this sensitivity makes them prone to overfitting. *Pooling layers* combat sensitivity by *down-sampling* the output tensor and

subsequently make the network more robust to the position of the visual features within the tensor.

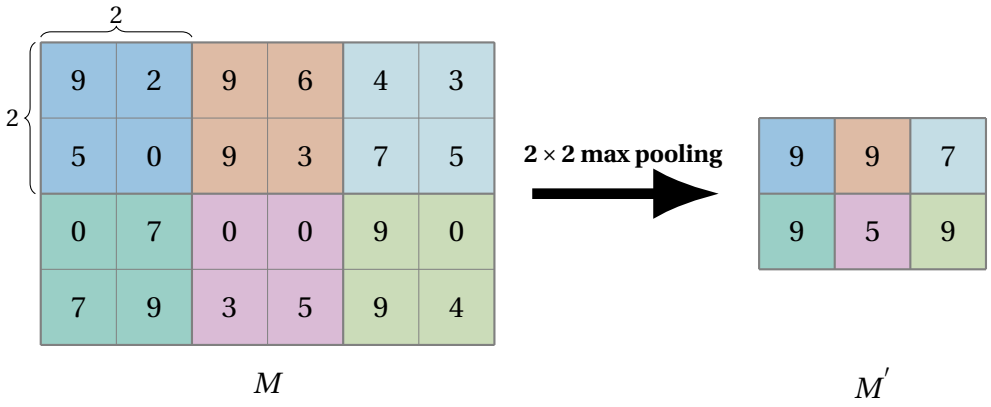


Fig. 2.5 A pooling operator applied to an output tensor. Given an output tensor $M \in \mathbb{R}^{4\times6\times1}$, a *max pooling operator* of size 2×2 splits M into corresponding regions. These regions are mapped to their maximum element giving the reduced tensor $M' \in \mathbb{R}^{2\times3\times1}$.

Formally, given an output tensor $M \in \mathbb{R}^{h\times w\times c}$, we split M into a set of disjoint regions of size $n \times m$, using padding to expand the output tensor if necessary. We then map each of these regions to a single output by using a pre-defined operator such as the mean or maximum. This produces a reduced tensor $M' \in \mathbb{R}^{\lceil \frac{h}{n} \rceil \times \lceil \frac{w}{m} \rceil \times c}$ which is much smaller in size but keeps the local spatial information. We present an example pooling operation in Fig 2.5.

2.3.4 An Example Network

Convolutional neural networks are often built up by an alternating sequence of convolutional and pooling layers before a fully-connected perceptron layer is used at the end to project the output to the space $\mathscr{Y}$. This structure of repeated blocks allows the network to learn low-level features in the early layers and high-level features in the deeper layers. A full example architecture is provided in Fig 2.6.

2.3.5 Optimising Neural Networks

For all deep learning models, the parameters, whether they be perceptron weights, kernels or biases, must be learnt from the available data. This process is the *optimisation* of the model. In this section we will concisely detail the optimisation of deep neural networks

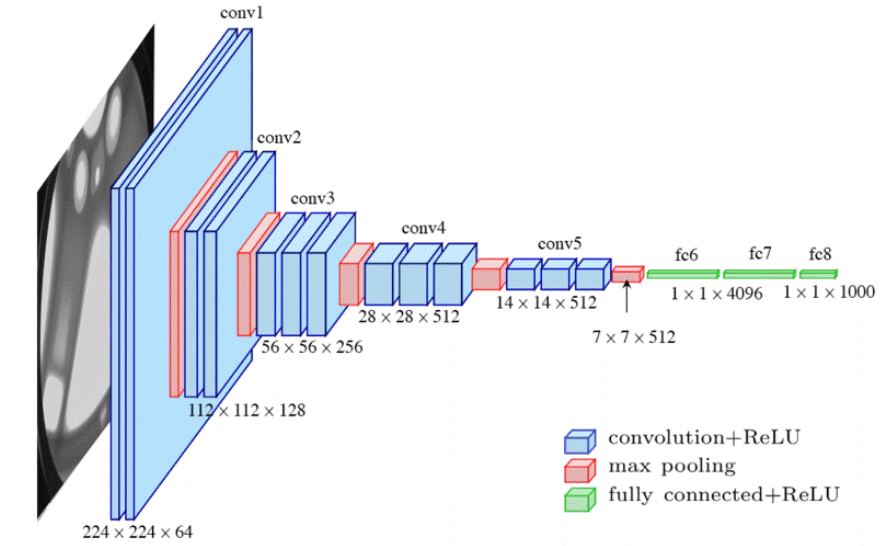


Fig. 2.6 The architecture of the famous VGG network [195] which is composed of a combination of convolutional layers, pooling layers and fully connected multi-layer perceptrons. Visualisation taken from [77].

for the task of image classification and the methods used in our work. We refer readers to [215] for a detailed overview of the topic.

Empirical Risk Minimisation

To first train a network we need to define a differentiable loss function l_s which measures the *error* or *risk* between the output of the network $f_\theta(x_i)$ and the ideal response y_i.

$$l_s(f_\theta(x_i), y_i). \tag{2.26}$$

In an ideal situation we would have access to the full distribution $\mathscr{X} \times \mathscr{Y}$ and we could integrate over the distribution to obtain the full error $R(\theta)$ of the model,

$$R(\theta) = \int l_s(f_\theta(x), y) dP(x, y), \tag{2.27}$$

Our task would then be to select the parameter set θ^* which gives the lowest value of this error. However, we never have access to the full distribution of the data. Therefore, we instead use the *empirical risk* L over the samples we have available to us,

$$L = \frac{1}{n} \sum_{i=1}^{n} l_s(f_\theta(x_i), y_i). \tag{2.28}$$

We utilise gradient descent [184] based algorithms to minimise $L(\theta)$. Due to the feed forward structure of the network, we can use the backpropagation [185] approach to calculate the partial differential of $L(\theta)$ of each parameter in $\theta = \{\theta_1, .., \theta_p\}$. However, due to the computational cost of calculating $L(\theta)$, standard gradient descent approaches are too time consuming. Therefore, throughout this thesis we use *stochastic gradient descent* (SGD) [26] to optimise our deep learning models.

For each iteration t of SGD a mini-batch $\mathscr{B}$ of size b is sampled from the dataset and the batch-loss is $L_{\mathscr{B}}$ is calculated via

$$L_{\mathscr{B}} = \frac{1}{b}\sum_{i=1}^{b} l_s(f_\theta(x_i), y_i). \tag{2.29}$$

The backpropagation algorithm is used to calculate the gradient of $L_{\mathscr{B}}$ w.r.t θ $\nabla_\theta L_{\mathscr{B}}$. Then the parameters θ are updated via

$$\theta^{t+1} = \theta^t - \eta \nabla_{\theta^t} L_{\mathscr{B}}, \tag{2.30}$$

where $\eta \in \mathbb{R}$ is a hyper-parameter termed the *learning rate*. Setting the value of η is very important for both performance and convergence and in our work we use cosine decay such that for a decay endpoint of T iterations, the learning rate at iteration t is given by

$$\eta_t = \eta_0 \cos\left(\frac{t\pi}{2T}\right), \tag{2.31}$$

where η_0 is the initial learning rate. This decays the learning rate to 0 as $t \to T$.

However, due to the over-parametrised nature of deep-learning models they are able overfit to, and in some cases memorise [244], datasets. Therefore, alongside SGD a variety of regularisation techniques are used to discourage overfitting and improve generalisation. In our work we use weight regularisation and batch normalisation [105].

Weight Regularisation

If the weights of a neural network are very large then the output of the network varies greatly with only small changes to the data input. This is a sign that the network has overfit to the training data. The most common solution to this problem is to add an additional loss term which regularises the weights of the network itself. The most popular weight regularisation is l_2 regularisation, where we have a Gaussian prior on the network weights such that they do not differ greatly from zero. From some training loss L, l_2 regularisation is added to give a new loss $\hat{L}$

$$\hat{L} = L + \frac{\lambda}{2}||\theta||^2, \tag{2.32}$$

where λ is a balancing parameter between the two terms.

Batch Normalisation

The backpropagation algorithm for training neural networks assumes that the weights in the layers prior to the current layer are fixed. Therefore, the weights of the current layer are updated based upon a given distribution of the later layers which themselves are changed by the algorithm. As neural networks have many hidden layers, this distribution mis-alignment effect builds up and was coined *internal covariate shift* [105]. The most common approach to reducing this shift is *batch normalisation* [105] and we use batch normalisation throughout this thesis.

Batch normalisation re-normalises the activations of the network at every mini-batch. For a given hidden layer, let $\mathscr{H} = \{\vec{h}_1, .., \vec{h}_m\}$ be the perceptron outputs for a mini-batch $\mathscr{B}$ of size m. The adjusted outputs $\tilde{\mathscr{H}} = \{\vec{\tilde{h}}_1, .., \vec{\tilde{h}}_m\}$ are given by

$$\vec{\mu}_{\mathscr{B}} = \frac{1}{m}\sum_{i=1}^{m}\vec{h}_i \qquad \sigma^2_{\mathscr{B}} = \frac{1}{m}\sum_{i=1}^{m}(\vec{h}_i - \vec{\mu_{\mathscr{B}}})^2 \tag{2.33}$$

$$\vec{\hat{h}}_i = \frac{\vec{h}_i - \vec{\mu}_{\mathscr{B}}}{\sqrt{\sigma^2_{\mathscr{B}} + \epsilon}} \qquad \vec{\tilde{h}}_i = \gamma\vec{\hat{h}}_i + \beta, \tag{2.34}$$

where γ, β are trainable parameters and ϵ is a small positive number to avoid $\vec{\hat{h}}_i \to \infty$. Although the above mathematics suggests a multi-layer perceptron architecture, batch normalisation is also applied to convolutional kernels in convolutional neural networks.

Chapter 3

Superpixel Contracted Graph-Based Hyperspectral Image Classification

In this chapter, we present a superpixel-contracted graph-based approach for the classification of hyperspectral images (HSI). This research was done jointly with Angelica I. Aviles-Rivero, Nicolas Papadakis, David Coomes, Anita Faul and Carola-Bibiane Schönlieb, who all had important advisory roles on the project, and resulted in the publication of [187] and [190].

3.1 Introduction

In modern applications, hyperspectral images capture a detailed light distribution over several hundred spectral bands, compared to the standard three colour bands of RGB images. A visual example of a hyperspectral image is given in Fig 3.1. For a hyperspectral imaging system, the typical wavelength bands used range from 400 to 1100 nm, which includes the visible and infrared spectrum of light. This detailed spectral information increases the discriminative ability of HSI compared to conventional colour images or multi-spectral images. As a result, hyperspectral imaging has been used in a wide range of applications including classification [149, 72, 69], object tracking [227, 210, 211], environmental monitoring [68, 148] and object detection [164, 139, 246].

In recent years, the classification of hyperspectral data has been an active topic of research. Classifying HSI requires assigning a class label, from a set of predefined classes, to each pixel within the image rather than assigning a single or small number of labels to the entire image. There are several large hurdles to overcome when designing a hyperspectral classification technique: the high dimensionality of the spectral information, the large

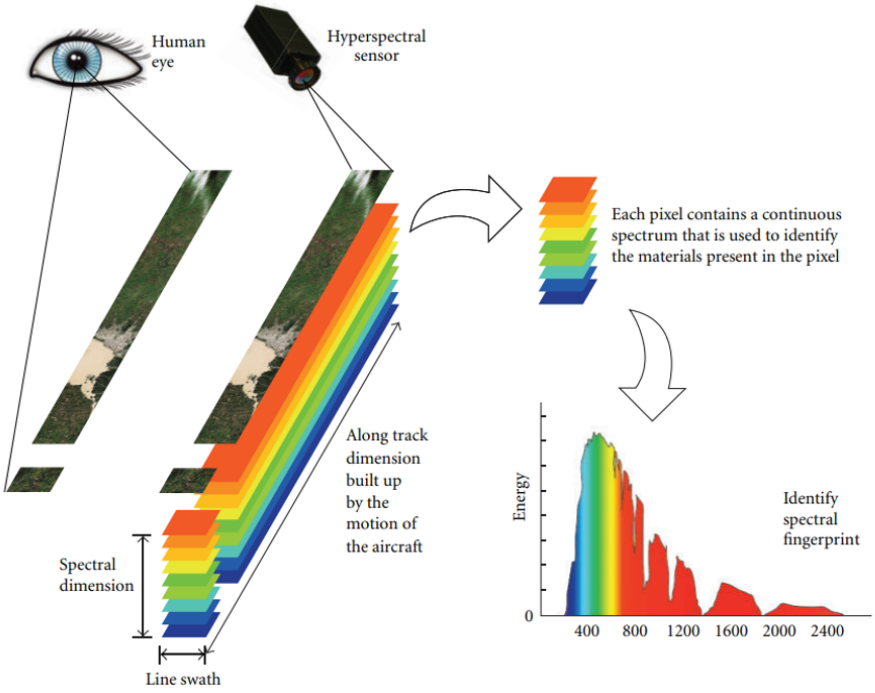


Fig. 3.1 Example of a hyperspectral data product being collected from airborne measurements. The hyperspectral image is made of many spectral bands which form a continuous spectrum for each pixel, and it contains far more information than a typical RGB image. Taken from Aiazzi et al [4].

spatial variability of the data and the limited number of training samples available due to the cost of obtaining labelled data. There have been numerous different attempts to deal with these problems and the majority of solutions rely on supervised learning. Here we briefly review the major supervised approaches.

Kernel based classifiers A commonly used classical approach is kernel based classifiers, such as support vector machines (SVM) [149, 150]. Whilst initial kernel methods only used spectral features, many later kernel methods included spatial features. An example being the multiple kernel learning (MKL) approach of Fang et al [71] which used MKL to combine spatial based feature vectors alongside spectral features. Other examples from this perspective are the works of Sun [201] and Gu [90].

Feature Extraction Approaches Feature extraction (FE) methods aim at finding a low dimensional subspace in which the separability among samples is maximised. Kang et al used image fusion and recursive filtering to extract meaningful features [117], Li at al [133] exploited local binary patterns to extract local features and textural information and Fang et al [70] used local co-variance matrix representation to characterize the correla-

tion between the spectral and spatial information in hyperspectral data. For a detailed discussion on this category of techniques we refer the reader to work of Rasti et al [172].

Deep Learning Motivated by the remarkable success of deep learning in computer vision, different works have used deep learning for hyperspectral image classification. Convolutional neural networks (CNN) are commonly used to extract high level spectral and spatial features [147] [247]. For example, Makantasis et al [147] used a CNN to extract spatial and spectral features and passed these into a multi-layer perceptron. In recent work, generative adversarial networks (GAN), which simultaneously train a generator and discriminator, have also been explored for hyperspectral image classification [251]. For a detailed overview on deep learning approaches for hyperspectral image classifiers we refer the readers to the work by Paoletti et al [165].

Although supervised classifiers have shown good results on hyperspectral data, their performance is heavily reliant on having a large quality training set, that can be obtained either by on-the-spot investigations or visual determination, which is a costly investment. Moreover, hyperspectral labelled dataset often contains inherent human error leading to the propagation of labelling error and uncertainty. As an alternative to supervised learning, a small number of works have attempted to use unsupervised learning, in which the key idea is to attempt to separate the data into distinct classes without prior knowledge on the labels. Although works such as Zhu et al [256] reported promising results on unsupervised hyperspectral image classification, the major problem with unsupervised learning is that the classification task becomes an ill-posed problem that needs specific assumptions to mitigate the lack of correspondence between the produced clusters and the known classes.

The aforementioned constraints associated with supervised and unsupervised learning make semi-supervised learning (SSL) a clear alternative for obtaining an improved classification performance. The advantages of SSL when using HSI data are two-fold: we decrease the need for large amounts of labelled data and we gain further understanding of the relationships present in the data.

3.1.1 Contributions

In this chapter, we introduce a novel superpixel contracted graph-based transductive learning framework for semi-supervised hyperspectral image classification, that we named *Superpixel Graph Learning* (SGL). Our approach combines graphical models, semi-supervised learning and over-segmentation. It produces state-of-the-art results, especially when the amount of labelled data is small.

The framework of SGL is composed of three main parts. Firstly, we modify a common superpixel method *Manifold Simple Linear Iterative Clustering* (MSLIC or Manifold SLIC) [140] to improve its performance on hyperspectral data and allow it to accurately partition our images into adaptive regions termed superpixels. Secondly, we perform feature extraction on each superpixel to extract discriminative features. Finally, we use the superpixels and features to produce a weighted graphical representation of our image which is then classified using a graphical-learning method (LGC [248]). Our main contributions are:

1. **Hyperspectral superpixels** We create and implement a new modified state-of-the-art superpixel method specifically with hyperspectral data in mind, which we call Hyper Manifold SLIC (HMS) . This modification uses a new clustering distance, which combines a Euclidean spectral distance with the Log-Euclidean distance of a covariance matrix representation. This allows us to define meaningful local regions to boost the overall classification performance. We demonstrate that our modified version outperforms other superpixel algorithms commonly used in hyperspectral approaches.

2. **Superpixel graph classification**. We show that a graphical superpixel representation and a purely graphical classifier brings two major advantages. Firstly, it vastly decreases the size of the node set which allows for graph classification in computationally feasible times without the need for matrix approximation methods. Secondly, it allows for the intelligent regularisation of the final classification map by using superpixels as adaptive local regions.

3. **Experimental Results**. We extensively validate our proposed approach by using three benchmarking datasets and six comparison methods and provide a range of experimental results. We demonstrate that our graphical superpixel approach (SGL) gives state of the art results for hyperspectral image classification across all datasets considered.

The remainder of this Chapter is organised as follows. In Section 3.2 we give a detailed overview of related semi-supervised learning methods. In Section 3.3 we discuss the problem of image over-segmentation and provide an overview of the key approaches from which we took inspiration. In Section 3.4 we present SGL and detail the methodology of the key components: a superpixel technique for HSI, feature extraction and graphical construction and label propagation. In Section 3.5 we detail the datasets used, the chosen experimental protocol, parameter selection and experimental results. In Section 3.6 we conclude the chapter and discuss further work and the limitations of the current approach.

3.2 Related Work

The problem of semi-supervised classification of hyperspectral images has been previously investigated by the remote sensing community. In this section, we review the existing techniques in turn. The literature regarding SSL algorithms can be roughly categorised into three different categories: *generative models*, *low-density separation* and *graph-based methods*.

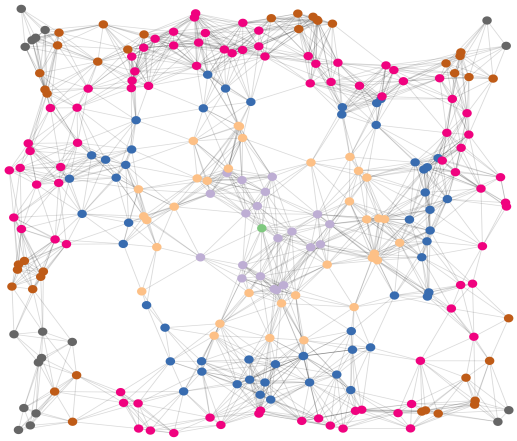


Fig. 3.2 Data visualisation using a weighted undirected graph. The different node colours represent the minimum path length between each node and the central node of the graph, which is coloured in green. Graphs are incredibly useful tools for capturing and visualising the detailed information present in data. Furthermore, graphs are particularly useful for visualising high dimensional data such as that present in hyperspectral images.

Several previous methods have utilised *graph-based learning*, and our approach is closely related to these. Graph-based methods rely upon constructing a graphical representation, where the data points are represented by nodes and the similarity between these data points are shown by edges and weights (see Fig 3.2). The first graph-based learning method for HSI classification was proposed by Camps-Valls in [39]. This paper used different spectral and spatial kernels alongside the Nyström extension as a matrix approximation tool. However, the produced accuracy was poor compared to other methods at the time. Gao et al [84] used a bi-layer graph-based learning algorithm to improve classification performance. The two layers were composed of a pixel-based graph, similar to [39], and a hypergraph built from grouping relations estimated using unsupervised learning. Cui et al [59] used an extended random walker (ERW) on a superpixel-based graph to optimise a classification map produced from an SVM. Showing that the accuracy of the SVM could be greatly improved by using the information present in the graph.

Another group of semi-supervised methods seek to directly implement the *low density separation* assumption [44] by moving the decision boundary away from unlabelled points. The first paper published in this area was by Bruzzone et al [33] which used a novel transductive SVM (TSVM) for HSI classification. A TSVM differs from a typical support vector machine (SVM) as it seeks to maximise the margin on a combination of labelled and unlabelled data. Building upon these ideas came self-learning algorithms such as the work by Dópido et al [67], in which they sought to adapt active learning, in which a user actively selects unlabelled samples, to a self learning framework in which the computer automatically selects the most informative unlabelled samples for classification purposes. Ratle et al [173] took a different path and tackled low density separation using a semi-supervised neural network architecture. An embedding regularizer was added to the loss function to inject the unlabelled information.

The rise of deep learning methods, has led to an increase in popularity of *generative methods* for semi-supervised learning. However, these methods are in still in their infancy. One of the most popular approaches by Zhan et al [243] uses a generative adversarial network (GAN) to simultaneous train a discriminator and generator. However, this paper uses a 1D-GAN and can only exploit spectral features and the produced accuracy suffers as a result. Zhu et al [251] developed a 3D-GAN which used a CNN for the discriminator and generator. This architecture allows the approach to exploit the spectral-spatial information present in the HSI. Therefore, the produced accuracy was much higher than [243].

Although works based on generative models and low-density separation have shown encouraging results, in this work, we concentrate on producing a graph-based method, the motivation for which is three-fold. Firstly, data can be naturally represented on graphs. Secondly, a graph representation is motivated by its mathematical background and properties including spareness. Thirdly, data can be represented in an uniform space even if the data is highly heterogeneous which is particularly relevant for hyperspectral imaging. We seek to produce a graph-based method that is based on a superpixel representation of the image similar to that of [59]. However, unlike [59] we seek to produce a fully graph-based learning method rather than a graph-based optimisation of a non graph-based method.

Fig. 3.3 Sample outputs of over-segmentation produced by the SLIC algorithm for superpixels of approximately 64,256 and 1024 pixels respectively. Taken from the work of Achanta et al [1].

3.3 Preliminaries

In this work we make use of over-segmentations, otherwise known as superpixel segmentation. The aim of over-segmentation is not to segment entire objects like traditional segmentation methods but instead to produce perceptually meaningful connected regions which group pixels similar in colour or other features. Superpixels were initially introduced by Ren and Malik [176]. In this section, we provide a definition of over-segmentations as well as reviewing the SLIC [1] and M-SLIC [140] algorithms from which SGL draws inspiration.

3.3.1 Over-segmentation

For a given RGB image, we give a general definition of what it means to over-segment an image and provide a visual example of an over-segmentation in Fig 3.3

Definition 3.1. Over-segmentation: An RGB image of integer width w and integer height h is a function $I : X \to \Omega$, where Ω is the $\mathbb{R}^3$ image colour domain and $X = [w] \times [h] \subset \mathbb{Z}^2$. An over-segmentation is a partition $X = \{R_i\}_{i=1}^{n}$ of X into disjoint sets $R_i \cup R_j = \emptyset \ \forall i \neq j$. which are path-connected on $\mathbb{Z}^2$. Each R_i is often referred to as a superpixel.

Existing superpixel techniques differ in how they partition the image and for a detailed survey on the varied approaches to superpixels, we refer to the reader to [198]. Our

approach is an extension to the Manifold SLIC approach [140] which itself is an extension to the SLIC approach [1]. Therefore, we give a concise overview of both SLIC and M-SLIC.

3.3.2 SLIC Superpixels

Simple linear iterative clustering (SLIC) is a fast and effective clustering based superpixel approach for RGB images. For a given RGB image I, each element $x \in X$ is an individual pixel in the two dimensional image grid $X \subset \mathbb{Z}^2$ with co-ordinates $x = (x_1, x_2)$. SLIC represents each pixel x as the combination of its colour $I(x)$ in the CIELAB colour space $I(x) = (l_x, a_x, b_x)$ and its two dimensional co-ordinate (x_1, x_2) in the data space X.

Definition 3.2. CIELAB color space: The LAB color space L_{ab} mathematically describes all perceivable colors in three dimensions represented by l for lightness and a and b for the green–red and blue–yellow colour components. For l the range of possible values is from 0 to 100 whilst a and b take values in a bounded interval with the bound depending on the convention used.

From this SLIC creates a metric space (X, D) where the metric function for two pixels $x, y \in X$ is given by

$$D(x, y) = d_c + \frac{m}{S} d_s, \tag{3.1}$$

where m is a constant controlling the compactness of superpixels, $S = \sqrt{N/K}$ and $d_s = ||x - y||^2$ and $d_c = ||I(x) - I(y)||^2$ are spatial and colour distances respectively. Given a user defined target number of superpixels K, the algorithm initialises K seeds $s \in \mathbb{Z}^2$ in the grid at regular intervals given by N/K where, N is the total number of pixels in the image. SLIC then partitions X using a Voronoi diagram.

Definition 3.3. Voronoi Diagram: Let (X, d) be a metric space with distance function d. Let $\{s_i\}_{i=1}^K$ be an ordered collection of K single sites in the space X. The Voronoi region R_i associated with the site s_i is given by

$$R_i = \{x \in X \mid d(x, s_i) \leq d(x, s_j) \ \forall \ j \neq i\} \tag{3.2}$$

and contains all points in X whose distance to s_i is not greater than their distance to any other site s_j. We denote the size of a Voronoi region by $|R_i|$. The Voronoi regions $\{R_i\}_{i=1}^K$ partition the space X and these Voronoi regions are the superpixels we will use in later tasks.

In our case X is the image grid and the metric is given by $D(x, y)$. The Voronoi diagrams are then iteratively updated by Lloyd's algorithm [141], also known as Voronoi iteration. Lloyd's algorithm iteratively improves the Voronoi diagrams by moving the seeds $\{s_i\}_{i=1}^K$ to their centre of masses $\{c_i\}_{i=1}^K$ and then recalculating the Voronoi diagram. In general, for a continuous $X \subset \mathbb{R}^2$ and given some density function $p : X \to \mathbb{R}^+$ the centre of mass of a Voronoi region is given by

$$c_i = \frac{\int_{x \in R_i} x p(x) dx}{\int_{x \in R_i} p(x) d(x)}. \tag{3.3}$$

SLIC uses a uniform density function and in the case of a discrete X we see that the centre of mass is simply given by the mean position of all pixels within the region.

$$c_i = \frac{1}{|R_i|} \sum_{x \in R_i} x \tag{3.4}$$

Note that a Voronoi diagram for which the seeds coincide with their centre of masses is termed a *Centroidal Voronoi Diagram* , which our diagrams are by construction. Aside from Voronoi iterations, SLIC makes one important choice which sets itself apart from the similar K-means clustering algorithm and greatly improves the computation performance. Instead of calculating the distance between every seed s and every pixel $x \in X$ it instead limits the search range to a square window of size $2\sqrt{\frac{N}{K}} \times 2\sqrt{\frac{N}{K}}$ centred on the seed. Therefore, SLIC has a time complexity of $O(N)$, with no dependence on the number of superpixels.

3.3.3 Manifold Content Sensitivity

Whilst SLIC achieved state-of-the-art performance alongside fast computational speeds, it failed to include content sensitivity due to its choice of a uniform density function. Content sensitivity is the notion that where you have high content value in an image you would want more superpixels, and where there is lower content value you would want fewer. *This is important for satellite images where large parts of the image are plain and small objects can hold high importance.* Therefore, in the work of Liu et al [140] they proposed an extension to SLIC to include content sensitivity.

In this work they define a stretching map $\Phi : X \to \mathbb{R}^5$ which sends pixels to a 2-manifold $\mathcal{M}$ embedded in the $\mathbb{R}^5$ space given by $\Phi(x) = (x_1, x_2, l_x, a_x, b_x)$. From this the area of a pixel can be calculated in the following way. For a pixel x , denote by $\square_x$ the unit square in X around x. Let, p_1, p_2, p_3, p_4 be the four corners of this unit square. $\Phi(\square_x)$ is made of two curved triangles $\Phi(\triangle_{p_1 p_2 p_3}) \cup \Phi(\triangle_{p_1 p_3 p_4})$, see Fig 3.4 for a visualisation.

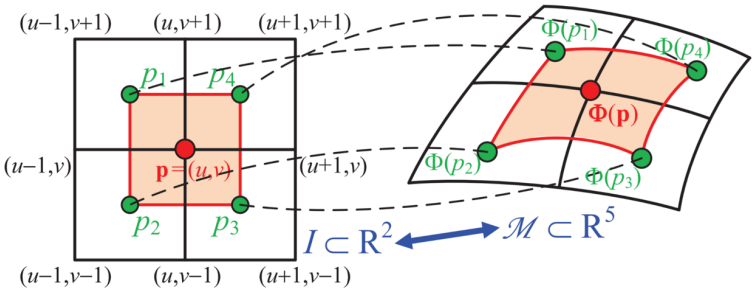


Fig. 3.4 Visualisation of the embedding process from the M-SLIC algorithm. A unit square in the image plane X is warped by the embedding to a 2-Manifold in $\mathbb{R}^5$ space. We can use the approximate area of the curved square on the manifold to act as a measure of the amount of content. Taken from Liu et al [140]

The authors approximated the areas of the curved triangles, and thus the area of $\Phi(\square_x)$ by using the areas of the planar triangles $\triangle_{\Phi(p_1)\Phi(p_2)\Phi(p_3)} \cup \triangle_{\Phi(p_1)\Phi(p_3)\Phi(p_4)}$. From this we can define the area of a Voronoi region by $A(R) = \sum_{x\in R} \Phi(\square_x)$ and the area of the whole image by $A(I) = \sum_{x\in I} \Phi(\square_x)$. The authors propose using the area of a pixel as a direct measure of its content as the higher the variation of colour the greater the area of the embedded square.

Using these pixel areas, the authors make a series of heuristic choices to promote content sensitivity. We note that whilst the original authors formulate their process as Voronoi iteration on the manifold, we find an equivalent formulation can be found by an iteration on the image grid itself which is easier to transmit. The three heuristic choices made by the authors are splitting Voronoi regions that have a large area, merging Voronoi regions which have low areas and an area adjusted search range. We use similar choices in our approach and so review the methodology of these choices below.

Content Adjusted Search Region

For a given image I and K seeds $\{s_i\}_{i=1}^K$, the expected area of each Voronoi region is given by

$$\mathbb{E}[A(R)] = \frac{1}{K}\sum_{i=1}^{K} A(R_i) = \frac{1}{K}\sum_{x\in X} \Phi(\square_x) = \frac{A(I)}{K} \tag{3.5}$$

Defining Ω_{s_i} to be the square of size $2\sqrt{\frac{N}{K}} \times 2\sqrt{\frac{N}{K}}$ centred on the seed s_i. The scaling factor λ_i for seed s_i is given by

$$\lambda_i = \sqrt{\frac{\mathbb{E}[A(R)]}{A(\Omega_{s_i})}}. \tag{3.6}$$

Thus the scaling factor for high content regions is $\lambda_i \leq 1$ whilst the scaling factor for low content regions is $\lambda_i \geq 1$. The factor λ_i is used in the splitting operation, which is explained later, and in the Voronoi iteration. Unlike in SLIC which limits the search range of each seed to a square window Ω_{s_i} around each seed, for MSLIC the search window is given by $\Omega'_{s_i} = \lambda_i \Omega_{s_i}$. Intuitively, this allows high content superpixels to focus on a small area whilst forcing low content superpixels to take up a larger area.

Voronoi Region Splitting

In order for more superpixels to be present in high content regions, the authors proposed Voronoi splitting. For each seed s_i if the corresponding scale factor $\lambda_i < 0.5$ and the area of the corresponding Voronoi region $A(R_i) > \mathbb{E}[A(R)]/4$, the seed is split. The area condition prevents seeds from continuous splitting. To split a seed the authors propose to replace the seed s_i at location $[x_1, x_2]$ with four seeds at locations half way between s_i and the corners of the square Ω'_{s_i} centred at s_i. The Voronoi regions for these seeds are then calculated and the algorithm continues.

Voronoi Region Merging

To ensure the number of superpixels stays close to the user defined target number K, seeds with low content are merged together in two ways. Firstly, if the area of a Voronoi region $A(R_i) \leq \mathbb{E}[A(R)]/8$ the region R_i is merged randomly with a connected Voronoi region. Note, connectivity is defined with respect to the grid $X \subset \mathbb{Z}^2$. Secondly, if the combined area of two connected regions R_i, R_j $A(R_i) + A(R_j) \leq \mathbb{E}[A(R)]/5$, then the two regions are merged together. Note the merger limit is picked to be higher than the split limit or else merged regions would be re-split immediately. When two seeds s_i, s_j are merged the postilion of the new seed s_{ij} is calculate as follows.

$$(x_{ij_1}, x_{ij_2}) = \frac{A(R_i)(x_{i_1}, x_{i_2}) + A(R_j)(x_{j_1}, x_{j_2})}{A(R_i) + A(R_j)} \tag{3.7}$$

3.3.4 Alternative methods for Content Sensitivity

In our work we used the manifold area of pixels to produce a content sensitive superpixel map. However, there are alternative methods to produce such mappings. One method, originally used for astrophysical image restoration is the **pixon** approach [168]. The pixon is derived from a Bayesian approach to image restoration. Given an image I, model M (which includes the image grid) and data D, the generalised image cells known as pixons are generated by maximising the probability $p(I|M)$, and can be thought of representing the minimum set of degrees of freedom necessary to describe the information in the image. Similar to the above superpixel approaches, the number of pixons produced is far smaller than the original number of pixels and speeds up later computational tasks.

The original pixon approach kept the shapes of the pixons constant but allowed the sizes to vary, a marked difference to superpixel methods. The pixons were generated by using a local convolution operation between a defined kernel function and a pseudo image. Yang and Jiang [239] proposed an extension to the pixon approach using Markov random fields, allowing for both shape and size to be varied simultaneously. Furthermore, in the work of [242] the authors demonstrated the adaption of the pixon approach for hyperspectral imaging. In this paper, the authors demonstrated that applying a pixon segmentation approach to regularise the classification output of a SVM yielded a large improvement in performance.

Despite, the success of the above approaches, pixon based approaches have not been a popular method for hyperspectral images as a whole. The improvement gained by the pixon approach of [242] compared very poorly to the similar superpixel based approach of [59]. As a Bayesian based approach, the computational time to compute the pixon mapping is much greater than the simpler superpixel algorithms used in this chapter. Furthermore, incorporating high level features into superpixel approaches can be done easily via the distance function, which can be harder to do for pixon methods where we must change several formula.

3.4 Methodology

This section explains in detail our proposed framework, SGL, which tackles the problem of HSI classification. Our task is to find an accurate classification prediction for a large amount of unlabelled data given an extremely small amount of labelled data. In order to do so we tackle the classification task under the transductive SSL paradigm. Transductive [44] in the sense we learn to classify specific pixels, those in the image, and do not learn general rules for any pixel.

Definition 3.4. Transductive Semi-supervised Classification Task. We are given a set of labelled observations $Z_l = \{x_i, y_i\}_{i=1}^{n_l}$, and unknown observations $Z_U\{x_i\}_{i=n_l+1}^{n_l+n_u}$ which are drawn from the same distributions $\mathscr{X} \times \mathscr{Y}$. Additionally, we have a pre-defined label set $\mathscr{L} = \{1, .., c\}$ where $\{y_i\}_{i=n_1}^{l} \in \mathscr{L}$. We seek to find a function $f : \mathscr{X} \mapsto \mathscr{Y}$, which utilises the labelled and unlabelled data such that f gives a good prediction for the unknown labels $\{y_i\}_{i=n_l+1}^{n_l+n_u}$

SGL can be split into three major tasks: *over-segmentation*, *graph construction* and *label propagation*. The explanation to each of these will be given in turn as the result of one feeds into the next.

Firstly, we will explain the dimensionality reduction used in this paper. Given the closeness of neighbouring hyperspectral bands there is a large correlation between them. Therefore, we applied principal component analysis (PCA) [116] to the hyperspectral data to extract non-correlated features and improve the computational efficiency. For PCA we performed dimensionality reduction using singular value decomposition of a matrix where each row contained the spectral information of an individual pixel. Denoting a hyperspectral image as $I : X \rightarrow \mathbb{R}^b$ where b is number of hyperspectral bands and $X = [w] \times [h] \subset \mathbb{Z}^2$, we produce a reduced image $\hat{I} : X \rightarrow \mathbb{R}^a$ where $a \ll b$. To select the dimensionality a, we required that the principal components must explain at least 99.8% of the variance present in the original data.

3.4.1 Superpixel Generation

In order to extract spatial features for use in spectral-spatial models, it is important to be able to define good local regions. Whilst setting a fixed size window (e.g. [48]) has shown good results, a fixed size does not allow for the full exploitation of spatial context. Using superpixels as adaptive regions [71, 110–113] has been shown to produce discriminative information for hyperspectral image classification. Cui et al. [59] demonstrated this by using a superpixel based random walker to optimise an SVM probability map to great effect. Furthermore, Cui et al. additionally demonstrated that a superpixel spectrum is more stable and less affected by noise that an individual pixel spectrum. Therefore by using superpixels we become more resistant to noise present in the data.

The vast majority of methods which seek to produce superpixels for hyperspectral data simply feed the first three principal components of HSI into RGB designed superpixel algorithms. However, we take a different approach and argue that by throwing away such a large amount of spectral information we limit the produced over-segmentation accuracy. Therefore, to ensure that our algorithm extracts effective information from hyperspectral

data, we took an existing state-of-the-art approach for RGB and modified it to work well with higher dimensional data.

As the base for our algorithm, we began with Manifold SLIC (MSLIC) [140]. MSLIC has two features that make it highly useful for our purpose. Firstly it produces content sensitive superpixels by mapping the image I to a two dimensional manifold $\mathscr{M}$ and measuring the area of Voronoi cells on $\mathscr{M}$. Secondly, the number of superpixels will change from the initial selection to fit the content structure in the image, thereby lowering the chances of a poor initial choice of K greatly reducing the final accuracy. However, out of the box MSLIC produces unsatisfactory performance on hyperspectral data as it fails to utilise more than three spectral dimensions. Therefore, we designed and implemented several key modifications to MSLIC to allow it to produce accurate results for hyperspectral data. We name this extension Hyper Manifold SLIC (HMS). HMS involves three major changes over MSLIC.

1. **High dimensional adaption:** We alter the MSLIC algorithm to take image data with any number of bands b. Therefore, the image function is viewed as $I : X \to \mathbb{R}^b$ and the colour information for each pixel is b dimensional.

2. **Hyperspectral clustering distance:** We design a more effective "colour" distance as a combination of the Euclidean spectral distance and Log-Euclidean distance (LED) [10] of a covariance matrix representation [209]. For each pixel $x \in X$ we construct a covariance matrix C_x using the same methodology as Fang et al [70] and use the LED metric to calculate the distances between these matrices. Therefore, the distance between two pixels $x, y \in X$ is given by:

$$d(x,y) = ||\text{logm}(C_x) - \text{logm}(C_y)||_F + ||\hat{I}(x) - \hat{I}(y)||^2 + \frac{m}{S}||x-y||^2. \tag{3.8}$$

   where logm is principal matrix logarithm. From (3.8), the parameter m controls the compactness of superpixels, as it weights between the spatial and spectral terms, whilst S scales the spatial distance and, for a image with N pixels, we take $S = \sqrt{N/K}$ as in the SLIC algorithm.

3. **Spectral Merging** In the original MSLIC algorithm, when the area of a region R_i is below a threshold it is randomly merged with a neighbouring region R_j. However, in our implementation we instead choose the neighbouring region which satisfies:

$$j = \underset{R_j \in \mathscr{N}}{\text{argmin}} ||\vec{f}_i^m - \vec{f}_j^m||^2. \tag{3.9}$$

where $\vec{f}_i^m$ is the average spectral information of the region R_i which is fully defined in the graph construction section and $\mathcal{N}$ is the set of neighbouring regions on the grid X. We choose to merge superpixels which are most similar in their spectral properties as this yields a better form of adaptation to hyperspectral data.

Additionally in the MSLIC algorithm, if two regions R_i, R_j have a combined area $A(R_i) + A(R_j) \leq \mathbb{E}[A(R)]/5$, then the two regions are merged together. However, we found that changing this rule to say that if two regions R_i, R_j have a combined area $A(R_i) + A(R_j) \leq \mathbb{E}[A(R)]/5$ and $A(R_i) + A(R_j) \leq A(R_i) + A(R_k)$ for any other region R_k connected to R_i then merge the regions. This prevents the merger of high content and low content regions and forces the algorithm to keep similar hyperspectral pixels in the same region.

These proposed changes allow HMS to produce accurate superpixels for HSI and we demonstrate through detailed experimentation that our approach produces much better over-segmentations than either the original SLIC or MSLIC algorithms.

3.4.2 Graph Construction

Now we seek to extract meaningful features from the extracted superpixels which can be used to construct the graphical representation. From each superpixel R_i we extract three different features and in this section we explain each of them in turn. To extract localised spatial information we apply a mean filter to each superpixel to produce a mean feature f_i^m which is defined as .

$$f_i^m = \frac{\sum_{x \in R_i} \hat{I}(x)}{|R_i|}. \tag{3.10}$$

Using a weighted combination of the mean feature of a superpixel's connected neighbours, we can obtain a measure of the spatial information between superpixels. Note that adjacency is defined based on connectivity on the image grid. For each superpixel R_i, we store the adjacent superpixels indices in the set $Z_i = \{z_1, z_2..\}$. From this, we construct the weighted feature vector f_i^w which reads:

$$f_i^w = \frac{\sum_{z_j \in Z_i} w_{R_i,R_{z_j}} f_{z_j}^m}{\sum_{z_j \in Z_i} w_{R_i,R_{z_j}}}, \tag{3.11}$$

where the weight between adjacent superpixels w_{R_i,R_j} is defined as:

$$w_{R_i,R_j} = \exp\left(-||\vec{f}_j^m - \vec{f}_i^m||^2/r\right), \tag{3.12}$$

with r is a scalar parameter. Finally, we propose to extract the centroidal location of each superpixel f_i^p which we calculate as:

$$f_i^p = \frac{\sum_{x \in R_i} x}{|R_i|}. \tag{3.13}$$

We now explain how we used these extracted features to create a undirected weighted graph-representation. However, first we give some background into challenges associated with the computational implementation of graph-based methods and how superpixels can be used to overcome some of these.

As noted by Camps-Valls et al [38], many graphical algorithms rely on calculating and manipulating large kernel matrices formed by the labelled and unlabelled data. As an example, for an image with N pixels the associated graph Laplacian is a matrix of size $N \times N$. If we seek to inverse the graph Laplacian via singular value decomposition then the computational complexity would be $O(N^3)$, greatly extending the computational time. Approximation methods do exist to speed up such matrix inversions. One commonly used technique is the Nyström extension [232] and it is regularly used to speed up matrix calculations [38] [25]. However, the Nyström extension has several drawbacks, a major one being that it is unsuitable for sparse applications as the Nyström extension acts as an approximation for complete matrices.

In this paper, we implement a novel solution to increase the speed and reduce the complexity of graphical classifiers applied to HSI. Instead of having a graphical representation where each node represents a pixel, we instead use our segmented superpixels as the node set. This greatly reduces the size of our node set and allows us to perform matrix inversion and other calculations without approximations such as the Nyström extension. Furthermore, a superpixel representation should help to boost the classification accuracy as we are defining a strong prior on our classification map, given that we are constraining the classification map to be constant across superpixel regions.

Therefore, from these previously discussed features and our superpixel node set, a weighted, undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ can be created. The weight w_{ij} between two superpixels R_i and R_j is constructed based on two Gaussian kernels, one for encoding spectral information and another for encoding spatial information, and is given as

$$w_{ij} = s_{ij} l_{ij}, \tag{3.14}$$

where

$$s_{ij} = \exp\left(\frac{(\beta - 1)||f_i^w - f_j^w||^2 - \beta||f_i^m - f_j^m||^2}{\sigma_s^2} \right), \tag{3.15}$$

$$l_{ij} = \exp\left(\frac{-||f_i^p - f_j^p||^2}{\sigma_l^2}\right). \tag{3.16}$$

where β balances the influence between the mean and weighted features and σ_s, σ_l determine the width of the Gaussian kernels. Note that the weight w_{ij} is limited in value between $[0,1]$ with 1 implying most similar. We then sparsify the edge set by using k-nearest neighbours. Therefore, in the symmetric weight matrix W_{ij} the edge weights are given by

$$W_{ij} = \begin{cases} w_{ij}, & \text{if } i \text{ is one of the } k \text{ nearest neighbours of } j, \\ & \text{or vice versa.} \\ 0 & \text{otherwise.} \end{cases} \tag{3.17}$$

from this weight matrix W we construct the degree matrix $D := \text{diag}(W\mathbb{1}_n)$ which gives a measure of how connected each node is across the graph.

3.4.3 Label Propagation

We now describe how the initial label information is propagated across our weighted graph representation to obtain the classification labels for the unlabelled points. Following standard protocol [39, 84, 59], we randomly sample a set of labelled spectral pixels from the original HSI. The initial label of each superpixel y_{R_i} is taken as the average initial label of its corresponding set of pixels .

$$y_{R_i} = \frac{1}{|R_i|} \sum_{x_i \in |R_i|} y_i \tag{3.18}$$

If no pixel within a superpixel is initially labelled then the superpixel is initially unlabelled. After the superpixel labels are created there is no further use of the pixel labels themselves. The labelling information for the superpixels are specified using a matrix $Y \in \mathbb{R}^{K \times c}$, where c is the number of classes present and K is the number of superpixels and Y_{vl} specifies the value of class l for node v.

The weight matrix and the initial labelling are then passed into the Local and Global Consistency (LCG) algorithm [248]. LCG is a graph-based SSL approach that formalises the smoothness and clustering assumptions of semi-supervised learning by designing a classification function which is smooth upon the graphical structure generated by all the data. The final labelling is specified using a matrix $F \in \mathbb{R}^{K \times c}$. The cost function associated with the matrix F is given by

$$Q(F) = \frac{1}{2}\sum_{i,j=1}^{n} W_{ij}\left\|\frac{F_i}{\sqrt{D_{ii}}} - \frac{F_j}{\sqrt{D_{jj}}}\right\|^2 + \frac{\mu}{2}\sum_{i=1}^{n}||F_i - Y_i||^2, \tag{3.19}$$

where $\mu > 0$ is a regularisation parameter and F_i denotes the ith row of the matrix F. The labelling matrix minimises this cost function and is given by $F^* = \text{argmin}_{F\in\mathscr{F}} Q(F)$ where $\mathscr{F}$ is the set of all $\mathbb{R}^{K\times c}$ matrices. The first term in the cost function is the smoothess constraint, which encourages connected nodes to have similar labellings, whilst the second term encourages the final prediction matrix to fit to the initial label data. Balance between these constraints is set by the parameter μ. The above cost function has a closed form solution which reads:

$$F^* = \beta\left(I - \alpha D^{-\frac{1}{2}} W D^{-\frac{1}{2}}\right) Y, \tag{3.20}$$

where $\beta = \frac{\mu}{1+\mu}$ and $\alpha = 1-\beta$. The final labelling of the superpixel nodes is then computed as: $y_{R_i} = \text{argmax}_{j\leq c} F_{ij}$. Each pixel within a superpixel is then assigned the label y_{R_i} thus giving the pixel wise classification map.

3.5 Numerical Results

In this section, we detail the experiments that were conducted to validate the proposed approach. In turn we will describe the chosen datasets, the evaluation protocol, parameter selection and finally present numerical and visual results for our approach.

3.5.1 Dataset Description

We use three standard benchmark hyperspectral imaging datasets to evaluate our approach, which have the following descriptions.

- **Indian Pines.** This image was collected by an airborne visible/infrared imaging spectrometer (AVIRIS) sensor over an agricultural site in Indiana, USA and has 16 classes. The classes cover a range of crops, woodlands and materials. The data set consists of 145 × 145 pixels, 200 spectral channels, a spectral range of 0.4 to 2.5μm and a spatial resolution of 20m.

- **Salinas.** This image was also collected by the AVIRIS sensor over Salinas Valley, California, and contains 16 classes, which are almost entirely agricultural in nature. The image size is 512 × 217 pixels and identical to Indian Pines has 200 spectral channels over 0.4 to 2.5μm. The data has a spatial resolution of 3.7m per pixel.

- **University of Pavia.** This dataset was acquired by the reflective optics system imaging spectrometer (ROSIS). The image (610 × 340 pixels) covers the Engineering School at the University of Pavia and has 9 classes, which are mainly material based such as bitumen, asphalt and soil etc. The image contains 115 spectral channels from 0.43 to 0.86μm and has a has a spatial resolution of 1.3m.

3.5.2 Evaluation Protocol

To evaluate the performance of each HSI classifier, we use three commonly implemented evaluation criteria *overall accuracy (OA)*, *average accuracy (AA)* and the *Kappa coefficient (Kappa)*. For each experiment in this chapter, each one is repeated ten times and the average and standard deviation are provided for each criteria. Note that each repeat uses a different randomly sampled subset of the initial labels.

To compare the performance of our proposed classification framework SGL, we implemented several state-of-the-art HSI classification methods to act as comparisons. These are local co-variance matrix representation (LCMR) [70], superpixel-based classification via multiple kernels (SC-MK) [71], the edge preserving filter based method (EPF) [118], local binary patterns (LBP) [133], an SVM method [149] and image fusion and recursive filtering (IFRF) [117]. For the compared methods the parameters were set using the default values provided in the demo code or referenced in the papers themselves. The SVM method was implemented using the LIBSVM [42] library and uses a Gaussian kernel and five-fold cross validation.

3.5.3 Parameter Selection

In our approach we have eight hyperparameters that come from the following tasks

1. **Superpixel construction**: The number of superpixels $K \in \mathbb{R}$ and the compactness of those superpixels $m \in \mathbb{R}$

2. **Feature Extraction and Graph Construction**: Exponential temperature $r \in \mathbb{R}$ for the weighted spectral feature. The widths $\sigma_s, \sigma_l \in \mathbb{R}_+$ of the spectral and spatial Gaussian Kernels respectively and finally $\beta \in \mathbb{R}$ which weights between the spatial and spectral features.

3. **Graph Propagation:** $k \in \mathbb{Z}_+$ sets the number of nearest neighbours for each point in the graph and $\mu \in \mathbb{R}_+$ which balances the normalised $p = 2$ graphical Laplacian and the data fidelity term.

FIXED PARAMETERS		
Parameter	Description	Value
m	Controls the compactness of superpixels	10.0
r	Weighted filtering kernel	15.0
σ_s	Kernel parameter for constructing s_{ij}	0.20
k	k-NN construction	8
μ	Weighting in the LGC classifier	{0.1,0.15}

DATA-BASED PARAMETERS			
Parameter	Indian Pines	Salinas	Pavia University
β	0.9	0.9	0.1
σ_l	{0.4,0.5}	{3.2.4.0}	{17,20}
K	1200	1400	2400

Table 3.1 The parameter values used for all experiments in this chapter. Note that $\{a_1, a_2\}$ signifies a random uniform distribution between a_1 and a_2.

The parameter values used in the experiments are given in Table 3.1. For the superpixels construction step, we enforce that the ratio of the number of pixels N to the number of superpixels K must $\frac{N}{K} > 15$. If this ratio is low, then the small number of pixels per superpixel prevents the superpixel contraction from actually improving the computational time for graph propagation. The value for the parameters m, h, σ_s, k and μ were constant across all datasets used. These values were found using empirical testing in a coarse to fine search method. The other three parameters, σ_l, β and K, change value depending on the HSI used and were also set by a fine to coarse search method. We note that that the parameters m, β and k had a very small effect on the final classification accuracy. The reasoning for the different value of σ_l and K is due to the different sizes of the input images. A larger image will need a larger number of superpixels to have a similar ratio of pixels to superpixels and will need a wider spatial kernel to reflect the larger pixel distances.

Superpixel Parameter Sensitivity

Given that the approach uses a superpixel level classification decision, it is critically important to understand how the number of superpixels K effects the accuracy. This is especially true when it is unclear what value of K to pick for a given image. To investigate the effect of changing the parameter K, we classified all three HSI using a varying number of superpixels and a small number of randomly selected labels for each class and plotted the classification accuracy against the superpixel number. The results for this analysis are given in Fig. 3.5. In general the classification accuracy increases with the number of superpixels, due to the underlying over-segmentation being more accurate. However,

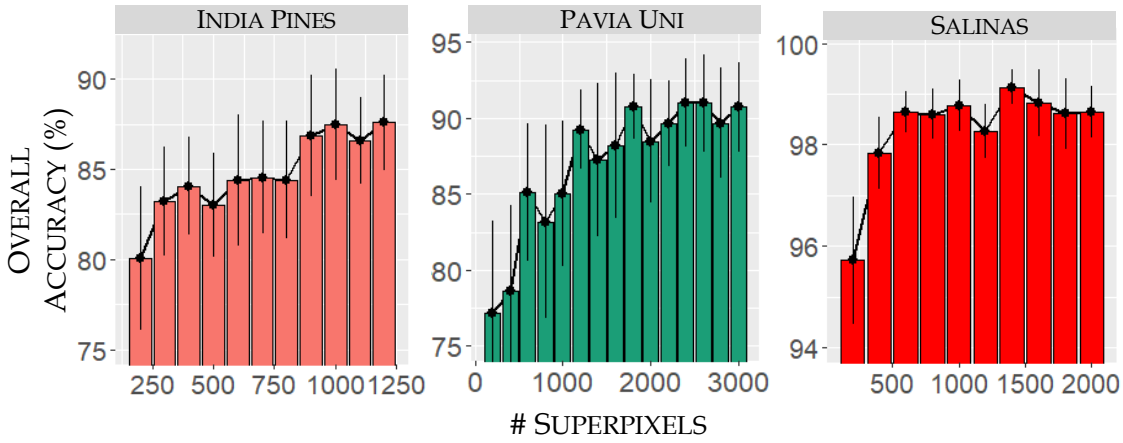


Fig. 3.5 Sensitivity analysis of the number of superpixels K for (a) Indian Pines, (b) Pavia University and (c) Salinas. Each data point is the accuracy average of ten repetitions using seven random labels for each class. The error bars reflect one standard deviation. For all three data sets the accuracy initally increases with increasing values of K. However, once the number of superpixels is high enough to accurately over-segment the image, there are limited returns for further increasing the number of superpixels and the accuracy flattens out.

once the image is accurately over-segmented, there are limited to no returns for further increasing the superpixel number. Combined with the fact that increasing the number of superpixels increases the size of the graph and thus the running time, we used the smallest number of superpixels that reliably gave a good classification accuracy for each HSI.

3.5.4 Experimental Results

In this section, the experiments are split into several parts. Firstly, the classification accuracy of the proposed framework is compared with the comparison classifiers detailed above. Secondly, we test our novel HMS algorithm for hyperspectral over-segmentation. Thirdly, we produce visual classification maps to understand and explain the performance difference of our classifier to relation to the others. Finally, we compare the computational time of our approach to the other classification approaches.

Firstly, we evaluate the overall accuracy (OA) of our method against the state-of-the-art when using a reduced amount of labelled data for training ($\{3,5,7,10,15,20\}$ randomly selected samples per class). The accuracy of the different classifiers for the three benchmark datasets are given in Table 3.2 and the graphical representation of the results is shown in Fig. 3.6. We see that the accuracy produced by the SGL framework is, by a significant margin, the best of any classifier considered in this paper. The SGL framework

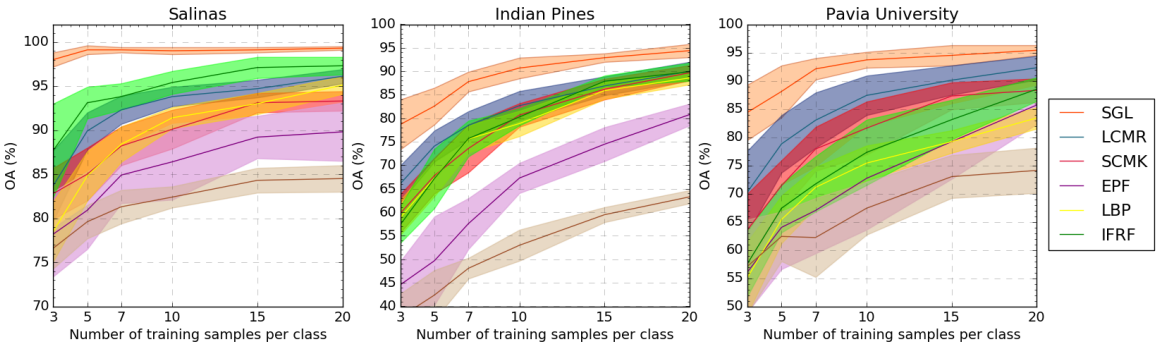


Fig. 3.6 Comparison of the classification accuracy of different methods with varying number of training samples. The methods used are LCMR [70], SC-MK [71], EPF [118], LBP [133], IFRF [117], SVM [149] and the proposed SGL method. The solid lines represent the average accuracy of each method whilst the shaded area covers one standard deviation from the mean.

produces the best accuracy for all three benchmark images for each differing amount of labelled data. For the agricultural Salinas scene our approach was able to generate great classification performance from only 5 labelled samples of each class. The average difference in OA between SGL and its nearest competitor LCMR [70], across the three datasets, was 9% when using 5 samples per class and was 13.7% when using 3 samples per class. These results demonstrate that including unlabelled data into the learning framework by means of semi-supervised learning can be a very powerful tool for overcoming poor generalisation due to limited labelled data in hyperspectral image classification

Furthermore, we sought to demonstrate that the modifications we made to produce our HMS superpixel algorithm for spectral data had had a positive effect on the accuracy of the over-segmentations when compared to the unchanged algorithm and algorithms commonly used in other approaches. To demonstrate this, we evaluated the OA produced by our model when using different superpixel algorithms in the over-segmentation step. We compared our approach to the unchanged Manifold SLIC [140] and additionally to entropy rate superpixels ERS [137] and SLIC superpixels [1]. For these comparative methods the parameters were kept the same as in the cited papers, the number of superpixels were set using Table I and three spectral bands, extracted by PCA were used as input. Note that unlike these comparison methods, HMS is being fed several ten's of spectral bands, extracted using the previously described dimensional reduction method, as it can cope with input images of any dimension. The accuracy comparison is given in Fig 3.7 and it clearly shows that the results produced by HMS are substantially more accurate. Thus highlighting the extra information that can be obtained by using more spectral bands and high dimensional features such as covariance matrices.

SALINAS							
SAMPLES PER CLASS	**OURS**	LCMR [70]	SC-MK [71]	EPF [118]	LBP [133]	IFRF [117]	SVM [149]
3	98.0 ± 0.8%	83.7 ± 4.0%	82.8 ± 2.9%	78.2 ± 4.8%	78.5 ± 3.4%	87.7 ± 5.3%	76.6 ± 2.3%
5	99.1 ± 0.5%	89.9 ± 2.1%	85.0 ± 3.0%	80.9 ± 4.4%	84.8 ± 2.8%	93.1 ± 1.8%	79.6 ± 1.9%
7	99.1 ± 0.3%	92.3 ± 1.6%	88.2 ± 2.2%	84.9 ± 3.2%	88.4 ± 2.0%	93.8 ± 1.5%	81.3 ± 1.9%
10	99.0 ± 0.4%	93.8 ± 1.1%	90.1 ± 2.2%	86.4 ± 4.3%	91.4 ± 1.2%	95.4 ± 1.3%	82.4 ± 1.2%
15	99.1 ± 0.3%	94.7 ± 1.0%	93.1 ± 1.1%	89.2 ± 2.4%	93.0 ± 1.2%	97.1 ± 1.2%	84.3 ± 1.4%
20	99.3 ± 0.2%	96.1 ± 0.8%	93.3 ± 1.1%	89.8 ± 3.3%	95.1 ± 1.1%	97.3 ± 1.0%	84.5 ± 1.5%

INDIANA PINES							
SAMPLES PER CLASS	**OURS**	LCMR [70]	SC-MK [71]	EPF [118]	LBP [133]	IFRF [117]	SVM [149]
3	78.7 ± 5.3%	66.1 ± 3.9%	59.8 ± 4.1%	44.6 ± 5.0%	58.9 ± 3.7%	57.4 ± 3.9%	37.7 ± 5.0%
5	82.6 ± 3.9%	74.1 ± 3.3%	67.8 ± 3.8%	49.7 ± 9.4%	67.3 ± 3.9%	67.2 ± 6.3%	42.4 ± 5.3%
7	87.8 ± 2.1%	78.5 ± 3.0%	73.6 ± 5.1%	57.6 ± 5.4%	75.6 ± 2.9%	75.7 ± 3.8%	48.1 ± 2.2%
10	90.7 ± 2.2%	82.7 ± 3.1%	80.7 ± 2.5%	67.3 ± 3.2%	78.9 ± 2.7%	80.3 ± 1.8%	53.0 ± 3.3%
15	92.9 ± 0.9%	86.9 ± 2.0%	86.2 ± 2.2%	74.5 ± 3.6%	85.9 ± 1.8%	87.9 ± 1.2%	59.5 ± 1.6%
20	94.4 ± 1.4%	90.0 ± 2.0%	89.7 ± 1.6%	80.8 ± 2.3%	88.6 ± 1.4%	89.9 ± 1.9%	63.3 ± 1.4%

UNIVERSITY OF PAVIA							
SAMPLES PER CLASS	**OURS**	LCMR [70]	SC-MK [71]	EPF [118]	LBP [133]	IFRF [117]	SVM [149]
3	84.4 ± 4.9%	70.3 ± 7.3%	63.6 ± 6.4%	56.1 ± 7.1%	55.1 ± 6.4%	57.6 ± 5.8%	57.1 ± 8.3%
5	88.1 ± 4.6%	78.8 ± 5.1%	71.4 ± 4.5%	64.0 ± 7.4%	65.4 ± 4.2%	67.4 ± 4.3%	62.4 ± 4.4%
7	92.1 ± 1.9%	83.0 ± 4.9%	77.9 ± 3.9%	67.0 ± 7.6%	71.1 ± 3.6%	71.7 ± 4.9%	62.2 ± 7.0%
10	93.7 ± 1.4%	87.4 ± 3.5%	81.6 ± 4.7%	72.7 ± 9.1%	75.4 ± 3.1%	77.3 ± 5.8%	67.4 ± 4.7%
15	94.5 ± 1.8%	90.1 ± 2.6%	87.3 ± 2.4%	79.2 ± 6.6%	79.2 ± 2.0%	83.1 ± 3.5%	73.0 ± 3.8%
20	95.4 ± 0.9%	92.3 ± 2.1%	88.3 ± 2.1%	85.7 ± 3.4%	83.4 ± 1.9%	88.5 ± 2.1%	74.1 ± 4.0%

Table 3.2 OA (%) of Ten Repeated Experiments with Differing Numbers of training samples per class for our approach and the compared classifiers over the Indian Pines, Salinas and University of Pavia datasets.

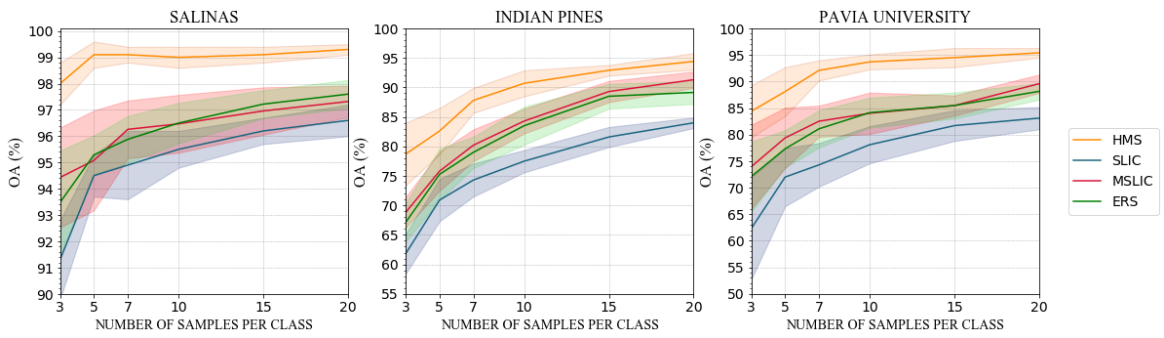


Fig. 3.7 Comparison of the classification accuracy obtained when using different superpixel algorithms for over-segmentation. Hyperspectral MSLIC (HMS) is the new modification proposed in this paper whilst SLIC[1], MSLIC[140] and ERS[137] are three commonly used superpixel methods which have been used for hyperspectral images.

In Figs 3.8, 3.9 and 3.10 we provide additional visual results for the HMS algorithm. We provide over-segmentations with differing numbers of superpixels highlighting the content sensitivity of the algorithm. In particular, note that Indian Pines and Salinas are easily

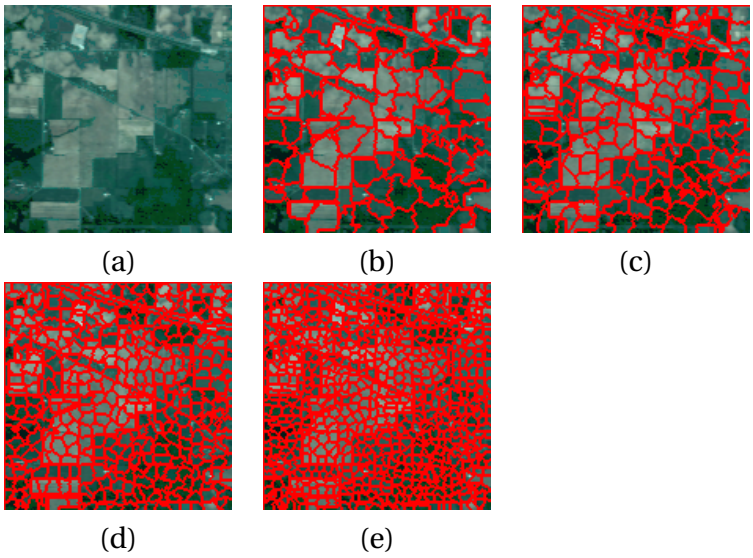


Fig. 3.8 Superpixel over-segmentations on the Indian Pines scene generated by the HMS extension. From left to right: (a) the composite RGB image, (b)-(e) superpixel segmentations with 129, 207, 434 and 791 superpixels respectively.

over-segmented using a small number of superpixels whilst the more complex structure of Pavia University requires more superpixels to achieve an accurate over-segmentation.

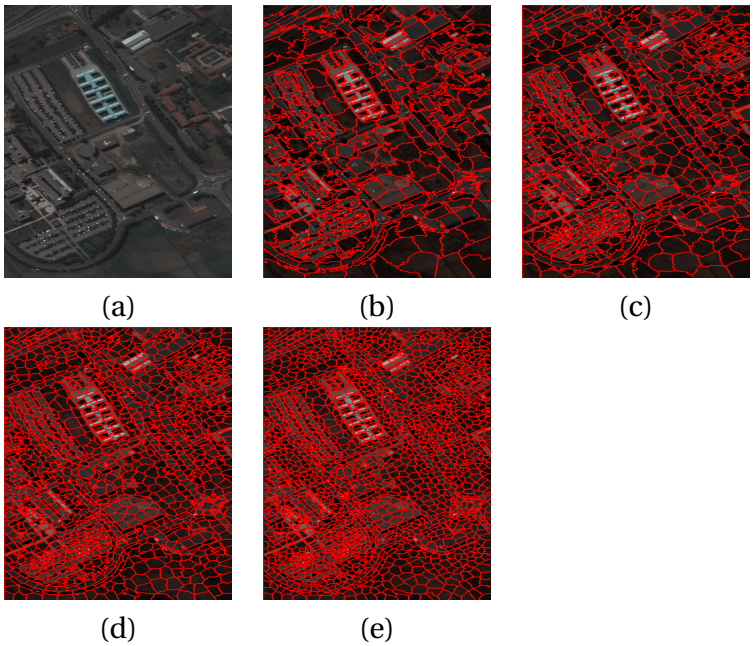


Fig. 3.9 Superpixel over-segmentations on the University of Pavia scene generated by the HMS extension. From left to right: (a) the composite RGB image, (b)-(e) superpixel segmentations with 948, 1286, 1662 and 1963 superpixels respectively.

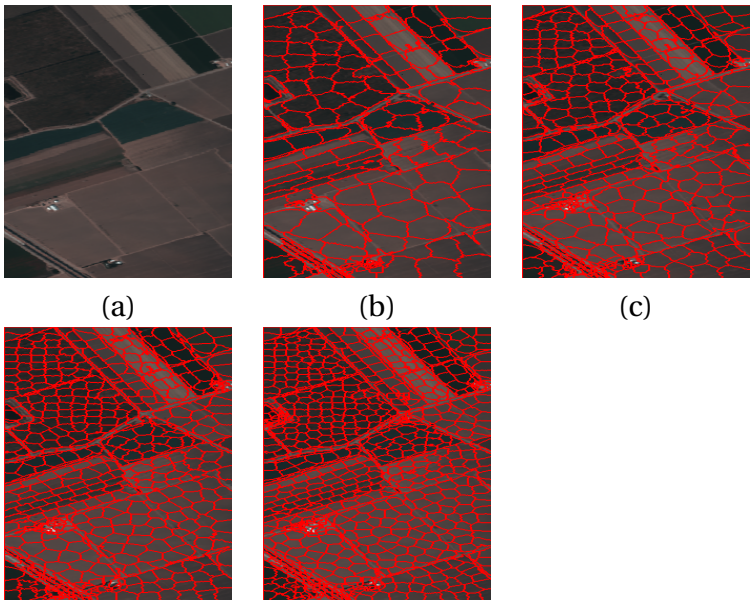


Fig. 3.10 Superpixel over-segmentations on the Salinas scene generated by the HMS extension. From left to right: (a) the composite RGB image, (b)-(e) superpixel segmentations with 244, 465, 639 and 919 superpixels respectively.

Visual Classification Maps

In our next evaluation we seek to gain an understanding about how each classifier was performing and the explanation for the large increase in classification accuracy obtained by SGL. In order to do this we produce visual classification maps for all methods over the three datasets. For each classification map, each colour corresponds to a different unique class. For each HSI we use ten labelled samples per class and ran the methods again to calculate the overall accuracy (OA), average accuracy (AA), the Kappa coefficient, a full class by class breakdown and full classification maps. The numerical results for this experiment are reported in Table 3.3 whilst Figs 3.11, 3.12 and 3.13 give the ground truth image and final classification maps for the seven considered methods for the three datasets used. Examining the OA, AA and Kappa coefficient of the differing methods, we observe that SGL is again the best performing method with an average improvement of OA +12.3%, AA +9.5% and Kappa +11.6% in the Indian pines scene, OA +15.4%, AA +13.8% and Kappa +17.3% in the Pavia University scene and OA +7.8%, AA +4.3% and Kappa +8.4% in the Salinas scene compared to the other classifiers, excluding the baseline SVM.

To provide an explanation for the performance of SGL compared to the other methods we examine the classification maps. The poorest performing classifier was the SVM which only uses spectral information and as a result produces very noisy classification maps. The EPF method seeks to optimise the SVM classification map with an edge preserving filter to smooth out some of this noise and from these results we can see it successfully does so. However, the poor performance of the underlying SVM classification prevents the EPF method from achieving good classification. The LBP and IFRF methods produce over-smooth classification results when only a limited amount of data is available. This causes poor performance in the more complicated Indian Pines and Pavia University images. The LCMR and SCMK methods are the closest competitors to the SGL method with LCMR slightly outperforming the SCMK method due to a slightly higher amount of smoothing. Both of these methods manage to preserve edges and boundaries whilst producing smooth classification maps. This is due to the inclusion of spatial information via local neighbouring pixel construction and superpixel based kernels respectively.

What sets SGL apart from the other methods considered is that the classification map has been intelligently smoothed with near complete preservation of edges and boundaries. Primarily, this is due to the use of superpixels as the node set in our graph. The superpixels produce by HMS have accurately preserved the edges and boundaries in the image. Therefore, when we assign labels to each superpixel, rather than each pixel, we smooth our classification map across the homogeneous superpixels whilst retaining boundaries.

Indiana Pines							
Class	SGL	LCMR [70]	SC-MK [71]	EPF [118]	LBP [133]	IFRF [117]	SVM [149]
1	98.04 ± 0.65%	99.44 ± 1.17%	98.33 ± 1.43%	53.78 ± 28.30%	100.0 ± 0%	53.51 ± 25.90%	20.53 ± 5.10%
2	76.06 ± 9.19%	75.99 ± 10.18%	78.82 ± 6.80%	55.88 ± 11.32%	70.94 ± 7.05%	70.35 ± 10.65%	43.14 ± 8.51%
3	84.27 ± 6.04%	70.78 ± 8.08%	77.65 ± 9.18%	60.19 ± 17.99%	70.28 ± 12.13%	67.99 ± 7.80%	39.15 ± 8.39%
4	96.67 ± 2.73%	93.88 ± 10,78%	86.39 ± 12.62%	34.57 ± 12.83%	97.93 ± 3.23%	79.37 ± 12.24%	21.25 ± 3.97%
5	92.30 ± 7.48%	90.32 ± 10.14%	82.33 ± 10.56%	93.39 ± 5.30%	82.92 ± 7.94%	79.40 ± 13.64%	59.34 ± 11.59%
6	98.71 ± 0.60%	91.40 ± 4.18%	89.54 ± 7.43%	86.32 ± 10.34%	90.36 ± 5.49%	93.63 ± 4.96%	83.29 ± 3.83%
7	100.0 ± 0.00%	100.0 ± 0.00%	100.0 ± 0.00%	70.92 ± 39.13%	100.0 ± 0.00%	39.65 ± 23.77%	24.85 ± 11.06%
8	100.0 ± 0.00%	99.68 ± 0.23%	97.09 ± 9.19%	98.41 ± 3.83%	100.0 ± 0.00%	99.97 ± 0.07%	93.12 ± 4.02%
9	100.0 ± 0.00%	100.0 ± 0.00%	100.0 ± 0.00%	59.44 ± 26.70%	100.0 ± 0.00%	28.81 ± 21.34%	12.59 ± 8.36%
10	88.94 ± 6.52%	76.46 ± 7.31%	71.32 ± 10.49%	61.19 ± 11.43%	79.90 ± 5.05%	75.95 ± 9.75%	36.93 ± 10.25%
11	91.04 ± 7.49%	71.40 ± 6.33%	69.22 ± 12.69%	81.05 ± 8.72%	73.78 ± 6.78%	93.81 ± 4.23%	61.50 ± 3.96%
12	90.05 ± 4.14%	90.50 ± 3.66%	78.47 ± 17.31%	44.31 ± 14.00%	70.58 ± 6.99%	74.08 ± 10.25%	28.20 ± 5.78%
13	99.56 ± 0.15%	99.33 ± 0.25%	99.90 ± 0.22%	98.34 ± 3.38%	98.31 ± 2.97%	75.32 ± 15.02%	80.12 ± 6.41%
14	100.0 ± 0.00%	98.18 ± 3.54%	88.14 ± 2.69%	95.15 ± 4.04%	91.32 ± 5.03%	98.31 ± 1.34%	88.22 ± 4.03%
15	97.69 ± 6.92%	91.62 ± 10.39%	90.96 ± 11.42%	64.91 ± 23.49%	90.59 ± 10.07%	77.14 ± 11.42%	39.10 ± 8.79%
16	100.0 ± 0.00%	98.67 ± 3.79%	97.59 ± 1.50%	84.46 ± 7.54%	98.43 ± 1.14%	92.62 ± 14.18%	87.71 ± 20.71%
OA	90.89 ± 2.98%	82.74 ± 2.32%	79.91 ± 2.60%	68.95 ± 2.01%	80.52 ± 2.03%	80.86 ± 3.76%	51.20 ± 3.92%
AA	92.16 ± 6.77%	90.48 ± 1.56%	87.86 ± 1.53%	71.39 ± 3.49%	88.46 ± 1.29%	74.99 ± 3.16%	51.19 ± 3.22%
Kappa	87.5 ± 3.33%	80.51 ± 2.59%	77.31 ± 2.93%	65.02 ± 2.25%	78.09 ± 2.23%	78.45 ± 4.16%	45.41 ± 4.11%

University of Pavia							
Class	SGL	LCMR [70]	SC-MK [71]	EPF [118]	LBP [133]	IFRF [117]	SVM [149]
1	86.64 ± 4.39%	79.29 ± 7.09%	72.48 ± 13.89%	94.80 ± 4.52%	59.64 ± 5.07%	68.30 ± 7.67%	94.09 ± 5.44%
2	95.87 ± 3.17%	87.67 ± 8.05%	80.05 ± 8.03%	89.55 ± 6.61%	69.72 ± 8.12%	94.90 ± 2.19%	85.59 ± 2.55%
3	85.37 ± 10.53%	90.96 ± 4.45%	76.84 ± 9.10%	62.03 ± 23.65%	79.52 ± 7.30%	53.78 ± 10.49%	42.74 ± 13.46%
4	87.44 ± 3.77%	95.10 ± 3.40%	94.77 ± 2.74%	57.08 ± 11.72%	66.44 ± 7.33%	66.44 ± 22.53%	59.85 ± 10.48%
5	95.84 ± 2.91%	97.03 ± 6.17%	99.66 ± 0.08%	91.20 ± 5.64%	89.91 ± 12.78%	99.63 ± 1.10%	93.69 ± 5.78%
6	99.92 ± 0.19%	95.37 ± 2.36%	76.24 ± 6.62%	49.32 ± 13.91%	89.33 ± 4.03%	82.47 ± 9.46%	39.38 ± 9.21%
7	96.59 ± 1.10%	92.58 ± 8.13%	76.06 ± 14.92%	66.86 ± 13.46%	89.15 ± 8.91%	63.32 ± 12.76%	42.26 ± 10.38%
8	94.03 ± 5.63%	84.67 ± 5.57%	79.79 ± 3.85%	75.57 ± 10.98%	80.78 ± 16.55%	55.33 ± 7.28%	73.22 ± 5.74%
9	97.55 ± 0.43%	93.80 ± 3.55%	100.0 ± 0.00%	98.48 ± 1.70%	59.40 ± 7.01%	49.33 ± 9.07%	99.87 ± 0.10%
OA	93.70 ± 1.35%	88.29 ± 4.06%	80.23 ± 4.06%	73.92 ± 7.06%	72.66 ± 4.29%	76.36 ± 3.81%	67.40 ± 4.66%
AA	93.25 ± 5.03%	90.72 ± 1.67%	83.99 ± 2.15%	76.10 ± 5.06%	75.99 ± 2.71%	70.39 ± 3.24%	70.08 ± 2.48%
Kappa	91.71 ± 1.73%	84.91 ± 4.89%	74.63 ± 4.60%	67.40 ± 8.23%	72.66 ± 4.25%	69.70 ± 4.55%	59.38 ± 4.88%

Salinas							
Class	SGL	LCMR [70]	SC-MK [71]	EPF [118]	LBP [133]	IFRF [117]	SVM [149]
1	100.0 ± 0.00%	99.95 ± 0.06%	99.93 ± 0.13%	100.0 ± 0.00%	97.97 ± 2.64%	95.77 ± 6.65%	97.54 ± 2.53%
2	100.0 ± 0.00%	93.21 ± 5.19%	98.66 ± 1.82%	99.87 ± 0.29%	96.57 ± 2.58%	100.0 ± 0.00%	99.10 ± 0.49%
3	100.0 ± 0.00%	99.56 ± 0.42%	96.94 ± 4.04%	93.84 ± 2.01%	98.59 ± 2.03%	99.32 ± 0.78%	86.62 ± 3.31%
4	99.71 ± 0.01%	100.0 ± 0.00%	98.79 ± 0.77%	97.70 ± 0.79%	97.84 ± 2.98%	87.42 ± 8.54%	96.92 ± 0.73%
5	98.09 ± 0.00%	96.88 ± 1.09%	95.63 ± 1.92%	99.48 ± 0.98%	92.37 ± 4.34%	99.92 ± 0.08%	97.57 ± 2.25%
6	99.93 ± 0.02%	98.53 ± 0.67%	99.53 ± 0.81%	99.98 ± 0.02%	92.14 ± 4.40%	100.0 ± 0.00%	99.97 ± 0.05%
7	99.48 ± 0.98%	97.57 ± 1.96%	94.22 ± 5.79%	97.92 ± 2.40%	92.68 ± 6.81%	98.88 ± 1.14%	97.68 ± 1.84%
8	99.38 ± 0.62%	87.84 ± 4.77%	74.47 ± 11.72%	84.19 ± 7.87%	85.27 ± 6.12%	96.83 ± 4.43%	70.82 ± 3.92%
9	100.0 ± 0.00%	96.90 ± 2.63%	99.40 ± 0.81%	99.47 ± 0.19%	93.05 ± 2.72%	98.82 ± 0.18%	98.84 ± 0.90%
10	96.72 ± 2.92%	93.71 ± 7.73%	88.34 ± 7.20%	86.13 ± 5.46%	93.65 ± 3.28%	99.21 ± 8.00%	79.64 ± 4.15%
11	95.882 ± 2.19%	99.94 ± 0.05%	97.03 ± 3.65%	91.81 ± 8.68%	97.83 ± 3.26%	98.96 ± 0.45%	83.29 ± 6.77%
12	99.90 ± 0.00%	99.60 ± 1.14%	97.50 ± 6.22%	99.42 ± 0.56%	89.96 ± 4.07%	98.19 ± 1.15%	94.42 ± 1.60%
13	98.80 ± 0.00%	98.65 ± 0.73%	95.36 ± 4.40%	96.32 ± 2.87%	91.59 ± 6.07%	92.20 ± 8.00%	88.15 ± 8.68%
14	95.38 ± 1.48%	95.23 ± 2.86%	90.29 ± 6.73	95.27 ± 11.66%	88.18 ± 6.84%	87.05 ± 14.28%	84.51 ± 17.07%
15	99.28 ± 0.07%	88.12 ± 7.50%	84.36 ± 6.15%	56.02 ± 5.52%	82.42 ± 12.45%	86.51 ± 8.12%	49.19 ± 2.71%
16	100.0 ± 0.00%	94.46 ± 6.47%	96.17 ± 3.18%	98.67 ± 4.09%	97.96 ± 3.91%	99.77 ± 0.56%	92.51 ± 8.59%
OA	99.24 ± 0.16%	93.90 ± 1.29%	90.38 ± 2.42%	86.53 ± 1.99%	90.68 ± 1.35%	95.87 ± 1.62%	82.42 ± 1.15%
AA	98.9 ± 1.51%	96.26 ± 1.02%	94.16 ± 1.11%	93.51 ± 0.91%	93.00 ± 1.03%	96.24 ± 1.43%	88.55 ± 0.99%
Kappa	99.15 ± 0.17%	93.22 ± 1.44%	89.33 ± 2.663%	85.10 ± 2.15%	90.68 ± 1.36%	95.41 ± 1.80%	80.53 ± 1.25%

Table 3.3 OA(%) AA(%), Kappa and a class by class breakdown obtained by different classifiers with ten training samples per class. The best results are highlighted in green.

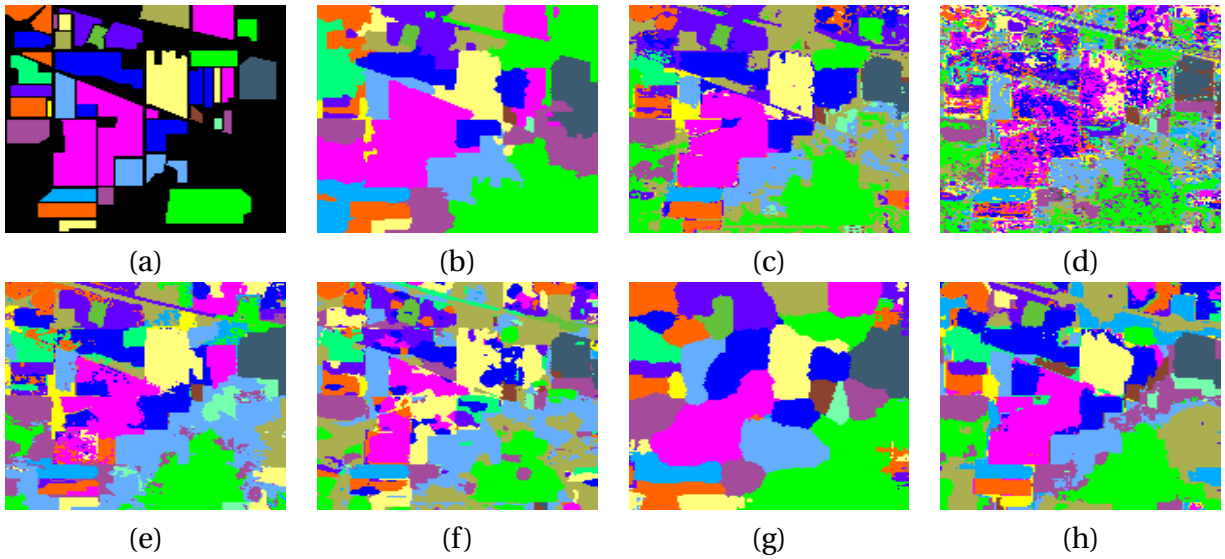


Fig. 3.11 Visual Classification maps for the Indian Pines dataset. (a) Ground truth. (b)-(h) are classifications maps produced using 10 labelled samples for each class. The methods used were: (b) the proposed SGL , (c) LCMR [70], (d) SVM [149] , (e) SC-MK[71], (f) EPF [118], (g) LBP [133] and (h) IFRF [117]

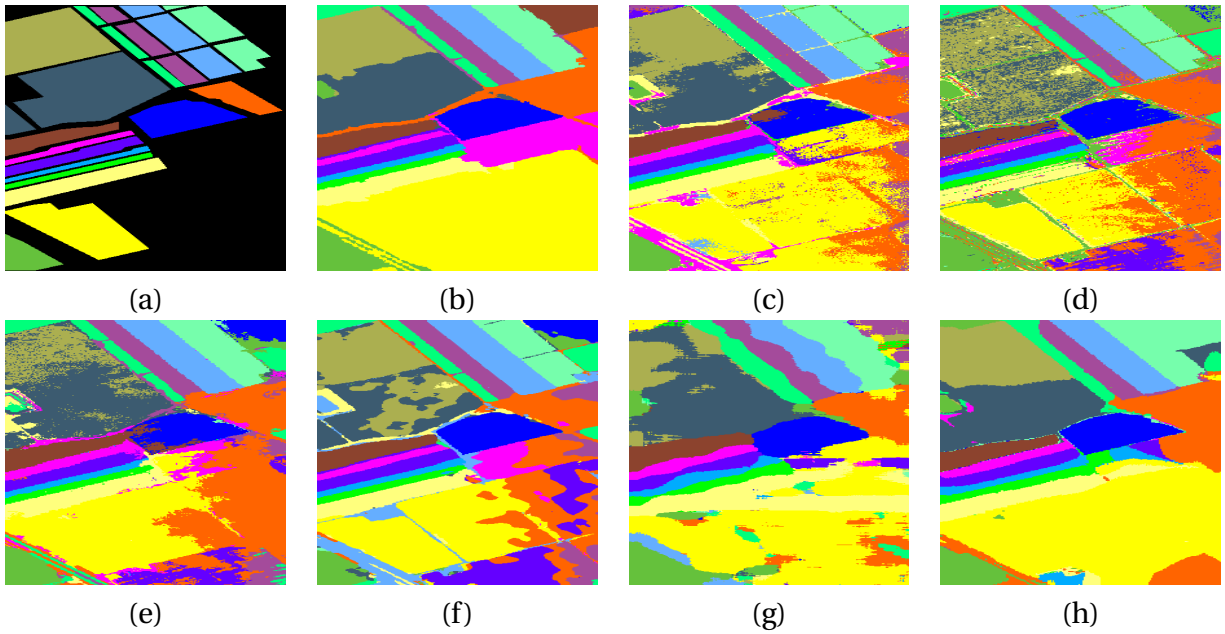
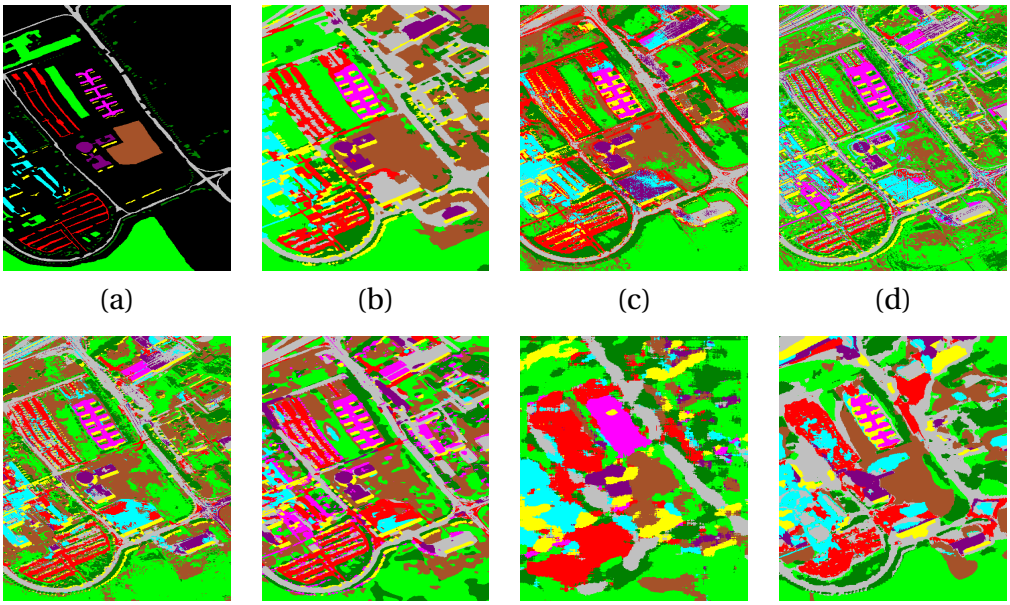


Fig. 3.12 Salinas data set. (a) Ground truth. (b)-(h) are classifications maps produced using 10 labelled samples for each class. The methods used were: (b) the proposed SGL , (c) LCMR [70], (d) SVM [149] , (e) SC-MK[71], (f) EPF [118], (g) LBP [133] and (h) IFRF [117]



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 3.13 Pavia University dataset. (a) Colour composite image. (b) Ground truth. (c)-(h) are classifications maps produced using 10 labelled samples for each class. The methods used were: (c) the proposed SGL , (d) LCMR [70], (e) SVM [149] , (f) SC-MK[71], (g) EPF [118], (h) LBP [133] and (i) IFRF [117]

TECHNIQUE	**OURS**	LCMR [70]	SC-MK [71]	EPF [118]	LBP [133]	IFRF [117]	SVM [149]
Indian Pines	16.9	9.6	2.8	5.2	41.1	2.1	5.8
Salinas	72.5	51.7	10.2	10.5	174.5	3.9	9.5
Pavia University	137.6	91.8	10.7	6.7	268.1	4.6	4.5

Table 3.4 Comparison of Computational Time(Seconds) With Ten Labelled Samples for Each Class

For our final experiment, we compare the computing time of the methods when applied to the three HSI reported in Table 3.4. We ran all test under the same conditions using an Intel Core i5-4670 processor and 16GB of RAM. As we can see, our full method (OURS) is one of the slower methods considered in this analysis. This is due to the time required to extract and manipulate the co-variance matrices. If we exclude the time needed to produce the covariance matrices the average computational time is instead 6.6s, 22.2s and 24.6s for the three datasets. As the covariance matrices are only used in the over-segmentation step, the implementation of a new faster over-segmentation approach, which preserves performance would allow our framework to achieve very fast performance in line with the compared methods.

3.6 Conclusion

In this chapter, we have presented a novel superpixel contracted graphical framework for semi-supervised hyperspectral image classification termed "Superpixel Graph Learning". We demonstrate through extensive experiments on benchmark datasets that our approach produces state-of-the-art results, particularly when the number of available labelled samples is small. Our results highlight the power of semi-supervised learning for hyperspectral image classification.

We created a novel hyperspectral modification to a popular over-segmentation algorithm which included adaptively changing to multi-band data, adding a hyperspectral distance and hyperspectral based merging. We demonstrate that not only is our over-segmentation approach much better than the base algorithm, but it greatly outperforms other over-segmentation algorithms commonly used in the field.

Our superpixel approach brought several benefits to our framework. Firstly, the size of the superpixel graph is much smaller than a pixel-based graph allowing for computational reasonable run times without the need for matrix approximations. Secondly, constraining labels to be constant across superpixels intelligent smooths the output classification maps with excellent preservation of edges and boundaries.

We utilised the normalised graph $p = 2$ Laplacian to propagate label information across this constructed superpixel graph to obtain final classification outputs. By examining the full classification maps, we show that our approach offers a great blend of label smoothing and edge preservation which outperforms the current state-of-the-art.

Whilst our approach performs well there are several properties of which could be improved by further research. In particular, the feature spaces used for both superpixel segmentation and graphical propagation are handcrafted. The local covariance matrices are particularly computationally expensive to calculate. Such hand-crafted feature spaces are often a bottleneck in performance. Therefore, transitioning to a learned feature space should increase the generalisation.

Furthermore, there are several parameters in this approach which are dataset dependent and must be tuned to new datasets. This is a rather unsatisfactory approach as often we were not have labelled data to tune to. Therefore, we can either seek to learn these parameter values during training or configure a way to eliminate them entirely from our framework.

Chapter 4

Pseudo-Labelling Approaches for Natural Image Classification

4.1 Introduction

Everyday in our modern lives we interact with a increasing number of cameras: CCTV, mobile phones, hand-held cameras, facial-recognition systems and an increasing number of cameras on autonomous machines such as self-driving cars. The amount of imaging data produced by these avenues is ever increasing. The images taken by these devices are termed natural images due to the fact they are similar to what we would see with our own eyes. Natural images form the dominant domain for computer vision research with tasks including image classification, object detection and image segmentation and a visual representation of these tasks is presented in Figure 4.1.

Fig. 4.1 Different computer vision tasks in the field of natural images. From left to right the task is image classification and object detection, object classification and instance segmentation.

In recent years the performance of computer vision systems has skyrocketed in many computer vision tasks. The development of deep learning frameworks have been fundamental in this performance increase and we can observe this across multiple vision tasks including classification [195, 127, 94, 102, 220], object detection e.g. [175, 85, 174] and image segmentation [142, 180, 46]. Training these deep learning models relies upon access to large amounts of labelled training data that form a representative sample of the data. The requirement of large representative labelled datasets is at odds with the fact that we often find that labels are scarce, expensive to collect, prone to errors (high uncertainty) and might require expert knowledge. *Therefore, relying on a well-representative dataset to achieve good performance is a major limitation for the practical deployment of deep learning methods.* These issues have motivated the development of approaches which combine the benefits of semi-supervised learning [44] with the powerful generalisability that neural networks possess which will be referred to as deep semi-supervised learning (DSSL).

In the last few years, DSSL approaches have reached unprecedented performance e.g. [197, 236], and the gap between supervised and semi-supervised models is much smaller then it ever has been, with semi-supervised methods surpassing certain supervised techniques in the natural image domain whilst requiring a fraction of the labels. However, the driving factor behind this performance improvement has been an increased model complexity, data augmentation, costly optimisation schemes involving multiple loss terms and technical tricks rather than a better understanding of the semi-supervised paradigm itself. Over-costly computational approaches and arbitrary technical tricks, make it hard to evaluate what tools or approaches are important for improved generalisation and make it difficult to use SSL methods in realistic settings as overly complex models often have to be fine tuned to the problem at hand due to hyper-parameter sensitivity.

Therefore, in this chapter we present three related research problems built around improving performance for DSSL for image classification whilst decreasing complexity or by adding theoretical understanding. These problems are as follows:

1. We explore the use of cluster-based regularisation as an alternative to perturbation based consistency regularisation. Rather than defining and applying local data perturbations we propose a global clustering approach which removes the need for creating domain specific perturbations.

2. We investigate the use of strong data augmentation in DSSL frameworks and propose, theoretically justify and experimental validate a multi-sampling approach.

3. We consider the advantage that graph-based propagation has over neural network predictions for estimating labels for unlabelled data points in pseudo-label approaches.

Contributions

The contributions of this research are contained in two proposed approaches for deep semi-supervised image classification: CycleCluster [188] and LaplaceNet [189]. This work was done collaboratively with Angelica I. Aviles-Rivero and Carola-Bibiane Schönlieb who supervised both projects and offered great insight and direction into the research presented in this chapter. These methods share major similarities in that both are built around graph-based pseudo-labeling but they differ in how they approach the problem. Note that in pseudo-labeling we estimate the true label for unlabelled data points and subsequently use them in training the model. In short, CycleCluster uses clustering as an additional learning task whilst LaplaceNet focuses on using data augmentation. The main contributions of the two approaches are as follows

1. **CycleCluster** We propose an approach named CycleCluster [188] which uses a direct implementation of the cluster assumption as an alternative to consistency regularisation approaches. We demonstrate that with clustering regularisation we do not need to design domain specific data perturbations. We demonstrate through rigorous experiments on benchmark datasets that clustering regularisation is a strong viable alternative to current state-of-the-art approaches.

2. **LaplaceNet** We propose a graph based pseudo-label approach for semi-supervised image classification which we name LaplaceNet [189]. In this work we propose, theoretically justify and experimentally demonstrate that a multi-sample averaging approach to augmenation not only improves generalisation but reduces the sensitivity of the model's output to data augmentation. Additionally, we show that using an energy-based graphical model for pseudo-label generation produces more accurate pseudo-labels, with a small computational overhead, than using the network's predictions directly. We demonstrate through extensive testing, that LaplaceNet produces state-of-the-art results on benchmark datasets CIFAR-10, CIFAR-100 and Mini-ImageNet without needing several technical tricks that are currently thought to be essential for pseudo-label methods and substantially reducing overall model complexity.

The remainder of this Chapter is structured in the following way. In Section 3.2 we cover the topic of data augmentation in both its theoretical justification and modern im-

plementation. In Section 3.3 we explore DSSL approaches and focus on the techniques of consistency regularisation and pseudo-labelling. In Section 3.4 we detail the methodology, implementation and results obtained using CycleCluster [188] and cluster regularisation . In Section 3.5 we then detail the methodology, implementation and results obtain using LaplaceNet [189] which also covers results relating to data augmentation and graphical pseduo-labels. For conciseness we do not repeat methodology which is shared between the two approaches and instead refer to the relevant passages. In Section 3.5 we conclude the chapter by summarising our findings and offering directions for further work.

4.2 Preliminaries

In this Chapter we make use of several different data augmentation schemes. As a preliminary to data augmentation we give a brief overview of its use, explore its theoretical foundation before focusing in on modern methods for neural networks which use basic image manipulations and image mixing. However, there are a large number of other data augmentations approaches used in the computer vision domain and we would refer to the review by Shorten and Khoshgoftaar [194] for a thorough overview of modern methods.

4.2.1 Data Augmentation

In statistics and specifically in the training of neural networks, we often deal with the concept of over-fitting. Over-fitting generally refers to producing a model that fits too closely to a particular set of data, and thus generalises poorly to unseen data. This can be commonly seen when training neural networks when the loss on the training set continues to decrease whilst the loss on the validation set begins to rapidly increase. Many methods have been proposed to reduce the problem of overfitting [194] and increase the model's generalisation to unseen data, data augmentation being one of them.

Vicinal Risk Minimisation

The motivation for data augmentation is formulated by Chapelle's et al work [43] on vicinal risk minimization (VRM). For some joint data and label distribution $\mathscr{X} \times \mathscr{Y}$, the standard learning framework is to search for a function f from a set of possible hypotheses F that minimises the risk

$$R(f) = \int l(f(\mathbf{x}), y) dP(\mathbf{x}, y), \tag{4.1}$$

where l is some loss function which measures the error of the model's response. This risk typically cannot be computed as we do not have access to the distribution $dP(\mathbf{x}, y)$. However, from a discrete sampling of this joint distribution we are provided with a dataset $\{x_i, y_i\}_{i=1}^n$ from which we can evaluate the empirical risk

$$R_{emp}(f) = \frac{1}{n}\sum_{i=1}^{n} l(f(x_i), y_i). \tag{4.2}$$

Therefore, we can see that minimising $R_{emp}(f)$ is equivalent to minimising $R(f)$ if we take the empirical distribution to be equal to

$$dP(\mathbf{x}, y) = \frac{1}{n}\sum_{i=1}^{n} \delta_{x_i}(\mathbf{x})\delta_{y_i}(y). \tag{4.3}$$

However, Chapelle et al pointed out that we can quite naturally consider other distribution estimates and in fact proposed replacing the delta function with a local estimate in the vicinity of the point x_i , $P_{x_i}(\mathbf{x})$. Thereby defining the vicinal risk to be

$$R_{vic}(f) = \frac{1}{n}\sum_{i=1}^{n} \int l(f(x_i), y) dP_{x_i}(\mathbf{x}). \tag{4.4}$$

Modern Implementations

Most data augmentation methods can be described either as a data warping or over-sampling approach, although both of these methods can be used in combination. Data warping approaches seek to apply label-preserving transformations to each image, thereby increasing the size of the available dataset, and use these transformations to create an estimate of the probability distribution. Given a potentially infinite set of label preserving transformations $A = \{a_1, .., a_m\}$, where $a : \mathscr{X} \to \mathscr{X}$, we take the vicinity estimate of the probability distribution at a point to be

$$dP_{x_i}(\mathbf{x}) = \frac{1}{m}\sum_{j=1}^{m} \delta_{a_j(x_i)}(\mathbf{x}), \tag{4.5}$$

which is a set of Dirac deltas located at all the possible transformations of x_i. Due to the computational infeasibility of using every possible transform, it is common to only sample from the vicinity distribution estimate once for each point such that the vicinal loss is given by

$$R_{vic}(f) = \frac{1}{n}\sum_{i=1}^{n} l(f(x_i), y_i)\delta_{a_j(x_i)}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} l(f(a_j(x_i)), y_i). \tag{4.6}$$

Fig. 4.2 Examples of image deformation techniques used in data augmentation. The original image on the left is colour bloomed and sheared using the RandAugment technique [58] to give the middle image and image erasing [58] is used to give the image on the right.

Often the transformation used is a composite transformation made by chaining several different transformations together. The simplest type of transformations used are colour and geometric transformations, where the image is cropped or flipped for example, which are informally referred to as *weak augmentations*. More severe transformations such as warping or colour transformations are also used and these are referred to as *strong augmentations*. Alternatively, image erasing techniques [65] remove a section of the image. We give a visual representation of both these effects in Fig 4.2. In slightly more complex works the transformations may be generated in an adversarial manor by another learning model [151]. Whilst the set of transformations is often kept constant during the training of the network, there are a group of approaches [57] which seek to optimise the set of transformations used during training by up-weighting transformations which lead to a large change in model output.

Alternatively , oversampling approaches seek to produce synthetic instances which can either be added to the training set in case of generative models [27] or used instead of the original images as is the case with image mixing [245]. As image mixing is used extensively in semi-supervised learning we further explore the detail of the methodology. Image mixing methods such as the widely used Mixup [245] are not label preserving and instead use a vicinity distribution estimate in both the label and data space given by

$$dP_{x_i}(\mathbf{x}, y_i|\lambda) = \sum_{j=1}^{n} \delta_{\lambda x_i + (1-\lambda)x_j}(\mathbf{x}) \delta_{\lambda y_i + (1-\lambda)y_j}(y). \tag{4.7}$$

In this equation $\alpha \in (0, \infty)$, $\lambda \in [0,1]$, and $\lambda \sim \text{Beta}(\alpha, \alpha)$ and x_j is another image from the dataset with label y_j. In the original MixUp paper the value of α was set using the validation dataset. Again, for computational reasons, typically only one sample is made from the estimated vicinity distribution. Note that these deltas are at linear interpolations between two images from the dataset.

For both data warping and oversampling methods such as MixUp the computational overhead from using data augmentation is very small and as is discussed in the related work, data augmentation has become a staple in deep semi-supervised methods.

4.3 Related Work

Whilst this Chapter focuses on deep learning, it is important to note that semi-supervised learning for natural image classification has been extensively investigated from the classical perspective which includes manifold learning [19], graph-based approaches [254, 249] and other techniques including manifold embeddings [19, 18, 88, 122]. Typically in classical methods one seeks to minimise a given energy functional that exploits the assumed relationship between labelled and unlabelled data [44]. However, classical approaches tended to rely on hand-crafted features that limited their performance and generalisation capabilities. With the popularisation of deep learning and its ability to learn generalisable feature representations, many techniques have incorporated neural networks to mitigate problems of generalisation. These modern deep learning methods are dominated by two approaches, consistency regularisation and pseudo-labelling.

4.3.1 Consistency Regularisation Techniques

One of the fundamental assumptions that allows semi-supervised learning to help performance is the *cluster assumption*, which states that points in the same cluster are likely to be in the same class. This can be seen to be equivalent to the *low-density assumption* which states that the decision boundaries of the model should lie in low-density regions of the data distribution. Following from the above assumptions, if we have access to some labelled data $Z_l = \{x_i, y_i\}_{i=1}^{n_l}$ and a large amount of unlabelled data $Z_u = \{x_i\}_{i=n_l+1}^{n_l+n_u}$, we should seek to move our decisions boundaries to be in low density regions of the joint labelled and unlabelled data distributions.

Consistency regularisation seeks to implement the low-density assumption by encouraging the trainable model f_θ with parameters θ to be invariant to perturbations δ to the unlabelled data x whilst accurately classifying the labelled data. As a result the decision boundaries are pushed away from the data points to low-density regions. Given some data perturbing function $u : \mathscr{X} \to \mathscr{X}$, such that $u(x) = x + \delta$, consistency based approaches seek to minimise some loss L in the general form of

$$L_{con} = \frac{1}{n_l} \sum_{i=1}^{n_l} l_{ce}(f_\theta(x_i), y_i) + \frac{1}{n_u} \sum_{i=n_l+1}^{n_u+n_l} l_s(f_\theta(u(x)), f_\theta(x)), \tag{4.8}$$

where l_{ce} is the cross entropy loss and l_s is some similarity loss function, for example cosine similarity or the l_2 norm loss. The first term is the supervised loss over labelled samples whilst the second term is the consistency loss. In order to prevent the consistency loss from causing the the neural network to collapse to the trivial solution of $f_\theta(x) = c \; \forall x$ the stop gradient operator is applied to the $f_\theta(x)$ term as show in Fig 4.3.

Modern Implementations

The downside of consistency regularisation techniques is the vagueness in choosing an appropriate perturbation δ which is often domain specific. This vagueness is reflected in the wide range of perturbations which have been used in the field. For example, Virtual Adversarial Training uses adversarial training to learn an effective δ for each image. Another example of this approach is Interpolation Consistency Training (ICT) [216] which combines the representations of unlabelled points. So for two unlabelled data points x_i, x_j ICT minimises

$$L_{\text{ICT}} = l_s(\lambda f_\theta(x_i) + (1-\lambda) f_\theta(x_j), f_\theta(\lambda x_i + (1-\lambda) x_j)), \tag{4.9}$$

where $\lambda \in (0,1)$. In recent works, several approaches have shown that using a data augmentation approach to consistency regularisation leads to large improvements in accuracy. The authors of [236] demonstrated that replacing flip and crop transformations or Gaussian noise perturbations for stronger augmentation perturbations such as warps and colour jitters (eg, RandAugment [58] or CTAugment [22]) led to a substantial performance improvement. Strong augmentation approaches often have to carefully construct their optimisation scheme as some approaches have shown common model divergences [197].

Perturbations are not only limited to the data. In the work of Mean Teacher [205] the authors decided to apply a perturbation to the model parameters rather than the input data. The authors use an exponential moving average (EMA) of the model parameters θ_{EMA} instead of data perturbations to generate a second representation as shown in Figure 4.4. Note that the parameters of the EMA model are not trained but updated at each training step t by the rule $\theta_{ema} = \alpha\theta_{ema} + (1-\alpha)\theta_t$ for $\alpha \in (0,1)$, where θ_t is the model's parameters at time step t.

Although these techniques have demonstrated great performance, it is unclear how best to set the perturbations δ and how best to incorporated them in learning frameworks, whether these be model or data perturbations. In our work, we avoid using model based perturbations and learnt data perturbations and instead focus on the the application of strong data augmentation which is common in supervised learning for natural images.

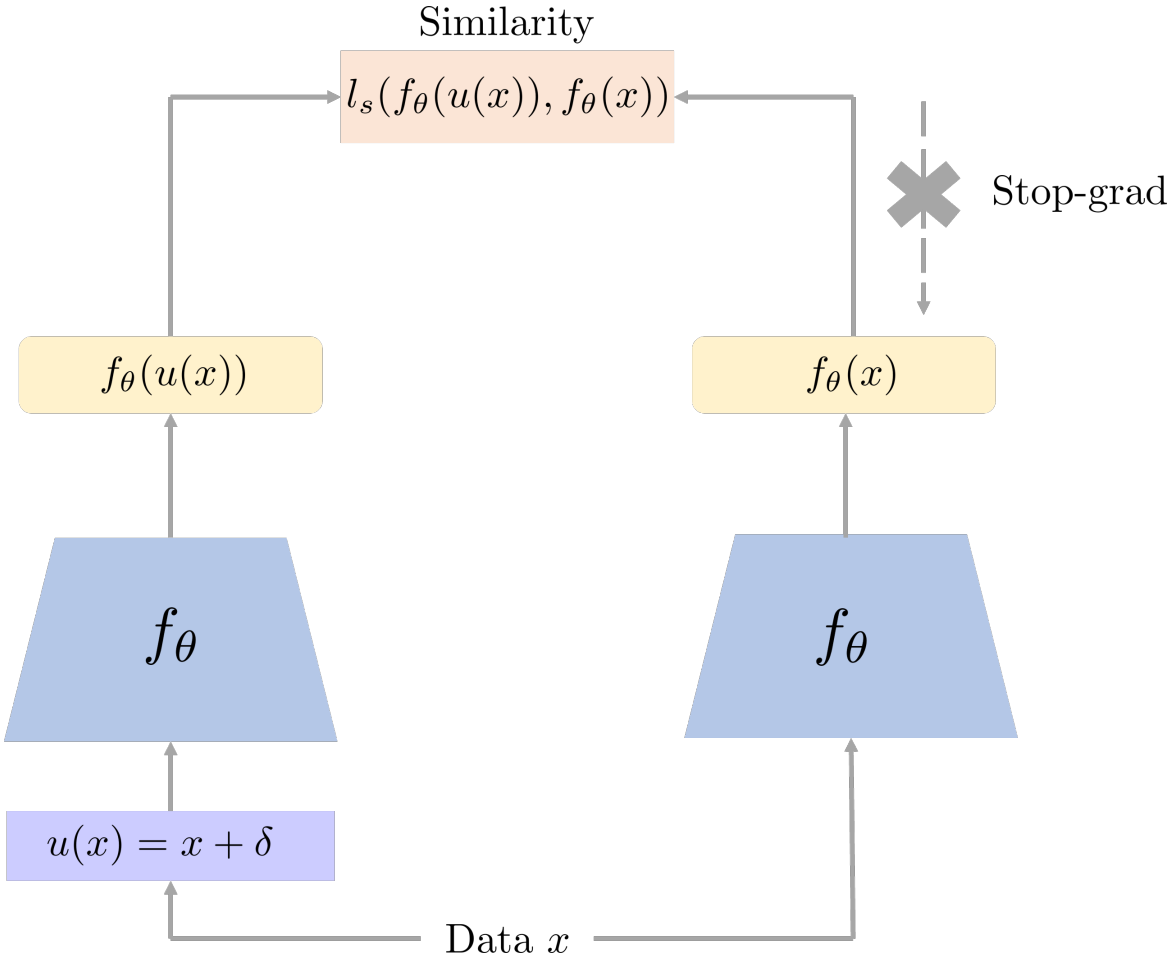


Fig. 4.3 Visual illustration of consistency loss. The initial unlabelled input image x and a perturbed version $u(x)$ are fed into the same neural network model f_θ to produce two representations $f_\theta(u(x)), f(x)$. In the case of classification, these representations are the produced probability distributions of the linear classification layer. These representations are then used to form a similarity loss for unlabelled data. Note the stop-grad operator on the gradients related to the original image x to prevent representation collapse.

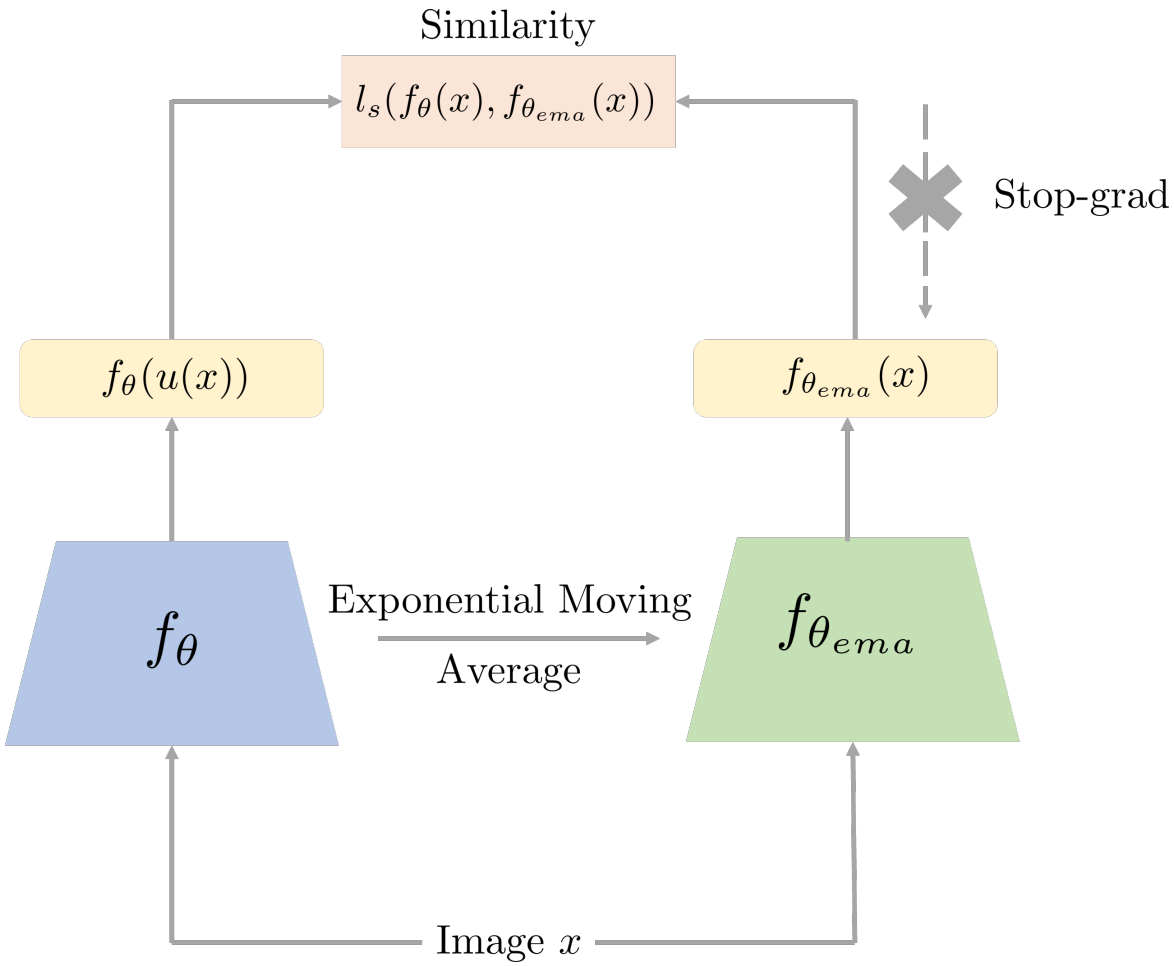


Fig. 4.4 Visualisation of model perturbations for consistency regularisation. The initial unlabelled input image x is passed into the model f_θ and an exponential moving average of the model parameters which is given by $f_{\theta_{ema}}$. This process forms two representations of the image which can then be used to create a similarity loss between them.

4.3.2 Pseudo-Labelling Techniques

Pseudo-label approaches focus on estimating labels $\hat{Y} = \{\hat{y}_{n_l+1}, ..., \hat{y}_{n_l+n_u}\}$ for unlabelled data points $Z_u = \{x_i\}_{i=n_l+1}^{n_l+n_u}$ and then using them in a modified loss function. Forcing the network to make predictions on unlabelled points minimises the entropy of the unlabelled predictions [44] and moves the decision boundaries to low-density regions. Additionally we increase the amount of labelled data the model has access to and reduce overfitting to the initally small label set. However, unless the produced pseudo-labels are perfectly generated, which they will not be in practical settings, the pseudo-labels will be noisy with many incorrect predictions. If the pseudo-labels are not carefully managed we encounter *confirmation bias* [8] where the model overfits to incorrect predictions.

The two most common ways to incorporate pseudo-labels into the loss function are to either use a composite loss function with a specific loss term for the pseudo-labels [197, 103] or by using composite batches made of part labels, part pseudo-labels [106, 8]. For a batch of labelled data points $\{(x_b, y_b) : b \in (1, ..., b_l)\}$ and a batch of unlabelled data points $\{(x_b, \hat{y}_b) : b \in (1, ..., b_u)\}$ the composite loss minimises the following objective

$$\hat{L}_{cl} = \frac{1}{b_l} \sum_{i=1}^{b_l} l_s(f_\theta(x_i), y_i) + \eta \frac{1}{b_u} \sum_{i=1}^{b_u+n_l} l_s(f_\theta(x_i), \hat{y}_i) + ..., \tag{4.10}$$

where η is a balancing parameter. The first term is over the initially labelled data whilst the second term is over the unlabelled data and there may potentially be several other terms which incorporate entropy regularisations [8] or consistency regularisation. For a composite batch approach the unlabelled and labelled batches are concatenated into one larger batch of size $b = b_l + b_u$ which is then used in loss objectives with the general form of

$$\hat{L}_{cb} = \frac{1}{b} \sum_{i=1}^{b} l_s(f_\theta(x_i), y_i) + ..., \tag{4.11}$$

where the first term represents the joint optimisation over the dataset and again there maybe other terms reflecting further regularisation.

Modern Implementations

In the literature, by far the most common approach to producing pseudo-labels is to use the output of the neural network itself [22, 23, 197, 8]. For an unlabelled data point x_i, the output probability distribution of the neural network is given by $f_\theta(x_i)$. This can be

used directly as a *soft pseudo-label* across all the classes or a hard pseudo-label can be generated by taking $\hat{y}_i = \arg\max f_\theta(x_i)$. Note that when the pseudo-label is given directly from the neural network, one can see that the mathematics of pseudo-labelling is very close to that of consistency regularisation.

The first application of this idea to the deep learning setting was presented by Lee [131]. Lee used a composite loss approach and took the maximum of the output of the neural network to give hard pseudo-labels. The pseudo-labels are recalculated every-time the unlabelled data is passed through the network. As an alternative Berthelot et al [23] used the full output probability distribution as a soft pseudo-label. However, due to the fact that the Shannon entropy of the full distribution is often high, it was found that sharpening the probability distribution led to better entropy minimisation. Sharpening a pseudo-label $\hat{y}_i$ amounted to the operation

$$\text{Sharpen}(\hat{y}_i) := \hat{y}_i^{\frac{1}{T}} / \sum_j \hat{y}_{ij}^{\frac{1}{T}} \tag{4.12}$$

where T is some temperature parameter. Unfortunately, it was found that the model was very sensitive to the parameter T, highlighting the importance of entropy minimisation for generalisation.

As pointed out by Arazo et al [8] there is a potential pitfall in this style of approach. Networks are often wrong and by using the network's prediction as estimates of the ground truth the network can overfit to its own incorrectly guessed pseudo-labels in a process termed *confirmation bias*. Several different, and often conflicting, approaches have been proposed to counter the negative effects of confirmation bias. Arazo et al proposed using a series of technical tricks in MixUp augmentation [245], soft labels and a composite batch approach with a minimum amount of labelled data per batch to reduce the effect of confirmation bias.

An alternative approach is built around the concept of confidence estimation. For each pseudo-label $\hat{y}_i$ a confidence score $c(\hat{y}_i) \in [0,1]$ is calculated. As examples, the work of [197, 106] used the entropy H of the probability distribution to give $c(\hat{y}_i) = H(\hat{y}_i)$ whilst [192] used a distance metric in feature space between the labelled and unlabelled data. These approaches then use the confidence scores in the loss function to weight confident pseudo-labels more strongly than non-confident ones. Works such as [106, 192] weight the loss function as

$$\hat{L}_{clc} = \frac{1}{b_l}\sum_{i=1}^{b_l} l_s(f_\theta(x_i), y_i) + \eta \frac{1}{b_u}\sum_{i=1}^{b_u} c(\hat{y}_i) l_s(f_\theta(x_i), \hat{y}_i) + ... \tag{4.13}$$

whilst approaches such as [197] instead reject all pseduo-labels whose confidence is below some threshold τ

$$\hat{L}_{clc} = \frac{1}{b_l}\sum_{i=1}^{b_l} l_s(f_\theta(x_i), y_i) + \eta\frac{1}{b_u}\sum_{i=1}^{b_u} \mathbb{1}_{\mathbb{C}(\hat{\mathbb{y}}_\mathbb{i})\geq\tau} l_s(f_\theta(x_i), \hat{y}_i) + ... \tag{4.14}$$

These confidence based approaches rely upon the assumption of network calibration, which is that there is some correlation between the confidence of a neural network prediction and the accuracy of that prediction. However, recent work [91] has shown that modern neural networks are often poorly calibrated which weakens the argument for using the confidence as a measure of estimate accuracy.

4.3.3 Graphical Techniques

Instead of using the output of the neural network as the label estimate, one can re-imagine the task of pseudo-labelling as a node classification task where we train a model or use a classical energy function to predict the labels of the unlabelled nodes. This is particularly useful since it has been shown from a classical perspective [254] that energy-based models on graphs are well suited to the task of node classification. However, a prerequisite to using a graphical technique is to first construct a weighted undirected graphical representation $G = (V, E, W)$ from the available data. Several works [106, 132] have shown that combining a feature representation generated from the parameters of the neural network and a carefully chosen metric function, produces a suitable graphical representation. Often the graph is iteratively updated by using the new parameters of the network generated after each epoch of training. The output of the node classification task at each epoch is then kept fixed and used to train the network in a semi-supervised manner.

Several different node classification methods have been used to produce the estimated pseudo-labels. The work of Iscen et al [106] focused on implementing "Learning with Local and Global Consistency" from Zhou et al [249] for deep learning networks which uses the $p = 2$ graphical Laplacian to propagate information over the graph. In the work of [13] the authors demonstrate the superiority of the $p = 1$ graphical Laplacian for producing accurate pseudo-labels at the cost of a greater computational time. Finally, in the work of Li et al [132] the authors proposed to use the neighbourhood information of each point in the graph to improve their feature embeddings.

4.4 CycleCluster: Modernising Clustering Regularisation

In this section, we present the CycleCluster [188] approach which uses a novel alternative to δ-based approaches based around direct implementation of the cluster assumption. The model iteratively trains a neural network using both an unsupervised cluster based task and a semi-supervised pseudo-label task on a shared architecture. Using the cluster assumption we are able to use global information from the unsupervised task to learn better decision boundaries which then allows for the generation of more meaningful pseudo-labels. Our modelling hypothesis is that by carefully combining our clustering regularisation approach to pseudo-label approaches we can greatly boost performance without needing to specify δ-perturbations which are domain specific. We demonstrate through rigorous experiments on benchmark datasets that this is the case and that clustering regularisation is a strong viable alternative to δ-perturbation techniques. Furthermore, we perform cluster based ablation experiments and show that the common problem of choosing the number of clusters is not a problem in our framework.

The rest of this section is structured as follows. Firstly, we detail the methodology of CycleCluster, focusing on the two learning tasks and how they are combined on the shared architecture. We then detail the experiments and results generated to compare clustering regularisation against the current state-of-the-art for δ-approaches.

4.4.1 Methodology

In this section, we detail both the clustering and pseudo-label learning tasks our model uses. We additionally give a high level overview of our model in Fig 4.5 and start by explicitly defining the problem at hand.

Problem Statement. From a joint distribution $\mathscr{X} \times \mathscr{Y}$ we have a dataset Z of size $n = n_l + n_u$ comprised of a labelled part of joint samples $Z_l = \{x_i, y_i\}_{i=1}^{n_l}$ and a unlabelled part $Z_u = \{x_i\}_{i=n_l+1}^{n}$ of single samples on $\mathscr{X}$. The labels come from a discrete set $y \in \{1, 2, .., C\}$ of size C . Our task is to train a classifier $f_\theta : \mathscr{X} \to \mathscr{Y}$, modelled by a neural network with parameter vector θ, which can accurately predict the labels of unseen data samples from the same distribution $\mathscr{X}$. The classifier f can be viewed as the composition of two functions t_θ and g_θ such that $f_\theta(x) = g_\theta(t_\theta(x))$. $t_\theta : \mathscr{X} \to \mathbb{R}^{d_p}$ is the embedding function mapping our data input to some d_p dimensional feature space and $g_\theta : \mathbb{R}^{d_p} \to \mathscr{Y}$ is a fully connected linear classifier projecting from the feature space to the classification space.

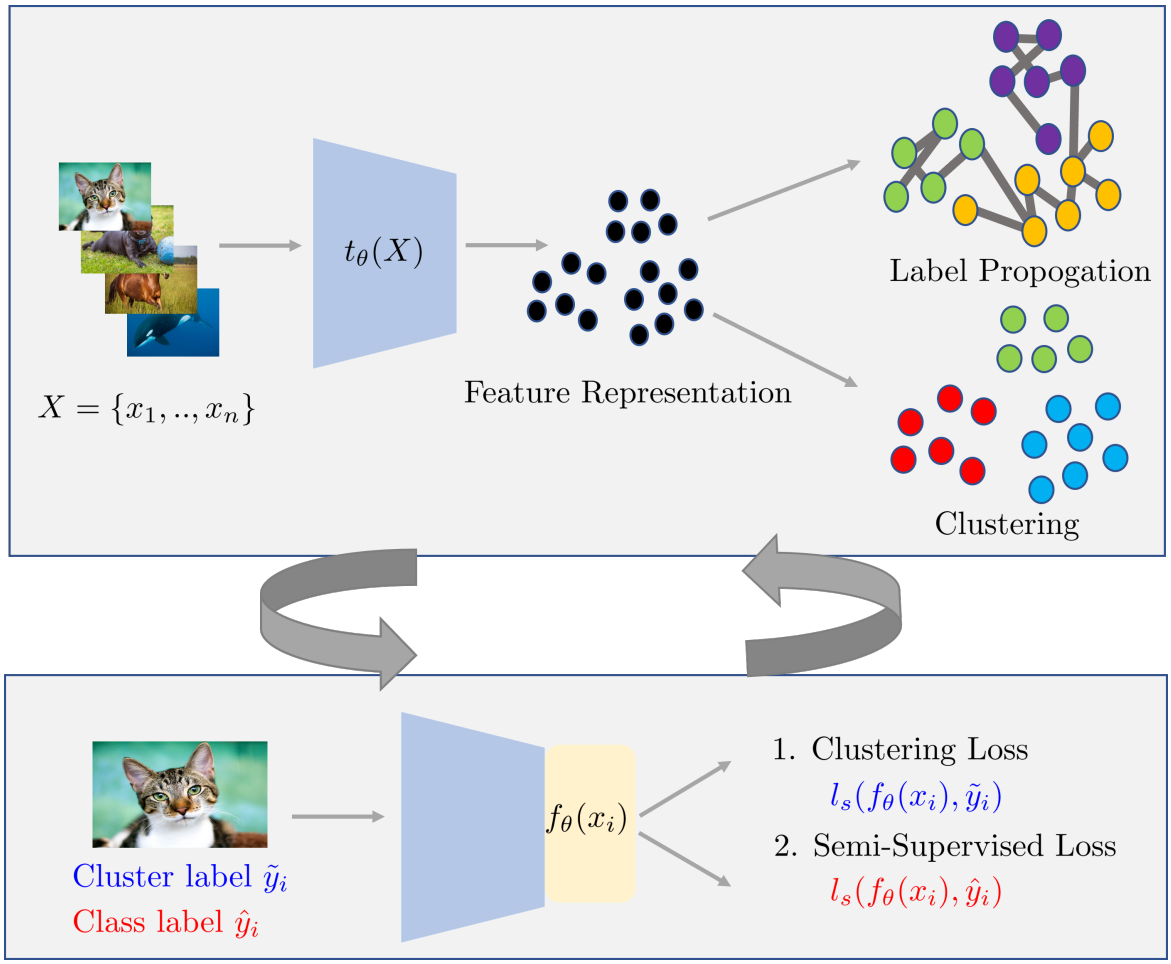


Fig. 4.5 Our approach named CycleCluster iterates between two phases. In the first phase, shown in the upper box, we extract a feature representation of our dataset $X = \{x_1, .., x_n\}$ by passing it through the network's feature extraction layers $t_\theta(X)$. From this representation we compute clustering labels $\tilde{Y}$ and classification pseudo-labels $\hat{Y}$ by using K-means clustering and graph-based label propagation respectively. In the second phase, shown in the lower box, we separately train our model on both a clustering loss and a classification loss for each image using the shared model architecture, $f_\theta(x)$. This updated model is then used to produce an updated feature representation and subsequently update labels. The algorithm iterates between these two phases for a given number of optimisation steps.

Clustering Regularisation

In this paper, we revert back to the original *clustering assumption* of SSL [44], data points in the same cluster are likely to be of the same class, which motivates our first learning task: *clustering regularisation*.

Using a clustering based approach we first need to cluster our data and then extract meaningful labels from which we can train our neural network. In order to cluster large-scale datasets we need a fast yet powerful clustering algorithm. One of the most popular algorithms is Lloyd's K-means [141] algorithm which has been used in several unsupervised deep learning approaches e.g. [40], and due to its performance and low computational burden we use K-means clustering in our approach. Additionally, we take inspiration from an observation in [177, 2] that *over-segmentation increases discriminative information* and allow the number of clusters K to be greater than the number of classes C which has not been investigated for the semi-supervised learning paradigm.

More precisely, given a fixed input feature representation $T \in \mathbb{R}^{n \times d_p}$ on n points in a d_p dimensional space, Lloyd's algorithms partitions T into K clusters with each cluster being characterised by a centroid. In our work we take the feature representation of each data point x_i to be the output of the feature embedding layer of a neural network $t_i = t_\theta(x_i)$. With this feature representation in hand, we now seek to solve a joint optimisation over the centroid matrix $M \in \mathbb{R}^{d_p \times K}$ and the cluster assignments $\tilde{Y} = \{\tilde{y}_1, .., \tilde{y}_n\}$ by solving the optimisation problem

$$\min_{M \in \mathbb{R}^{d_p \times K}} \sum_{i=1}^{n} \left[\min_{\tilde{y}_i \in \{0,1\}^k} ||M\tilde{y}_i - t_\theta(x_i)||^2 \right], \tag{4.15}$$

for a certain number of iterations. Due to $\tilde{Y}$ being model produced, we often encounter the scenario where the model produces trivial solutions where some subsets of the cluster are empty. Therefore, at the end of every iteration we reassign empty clusters by moving the corresponding empty centroid m_i to the nearest non-empty cluster centroid m_j plus a small randomly generated displacement.

After the cluster psuedo-labels are produced we then use them to perform unsupervised training by minimising the predictive loss between the cluster label and the output of the neural network

$$\theta \leftarrow L_C(X, \tilde{Y}; \theta) := \frac{1}{n} \sum_{i=1}^{n} l_{ce}(f_\theta(x_i), \tilde{y}_i), \tag{4.16}$$

where the pseudo-labels $\tilde{Y}$ are fixed and l_{ce} is the cross entropy loss. The cluster labels $\tilde{Y}$ are recalculated after every epoch of training.

Pseudo-Label Semi-Supervised Learning

In this section, we discuss our second learning task, semi-supervised learning with pseudo-labels and how it is connects to the cluster regularisation. As well as the clustering assumption, the ability for SSL to yield increases in performance also relies on the *smoothness assumption*, in that *if two points x_1, x_2 are close then the corresponding outputs y_1, y_2 should also be close.* In the context of neural networks we rewrite this as, *if two feature representations $z(x_1), z(x_2)$ are close then their outputs y_1, y_2 should also be close.* To enforce this constraint we use the approach of label propagation (LP) [248] by minimising the $p = 2$ graphical Laplacian.

Given the feature representation of our data $T = \{t_i\}_{i=1}^n$ we construct a weighted graphical representation of our data $\mathscr{G} = (\mathscr{V}, \mathscr{E}, \mathscr{W})$ with adjacency matrix $W \in \mathbb{R}^{n\times n}$. The edge weightings are produced using the normalised inner product metric so that

$$w_{ij} = \left\langle \frac{t_i}{||t_i||^2}, \frac{t_j}{||t_j||^2} \right\rangle \tag{4.17}$$

The weight matrix W is then sparsified by using K-nearest neighbours. From W we construct the degree matrix $D = \text{diag}(W\mathbb{1}_n)$, where $\mathbb{1}_n$ is the one vector of length n. Finally we construct the initial label matrix $Y \in \mathbb{R}^{n\times c}$ where $y_{ij} = 1$ if the ith image was provided with label j. We extract the final prediction matrix F by minimising the objective

$$Q(F) = \frac{1}{2}\sum_{i,j=1}^{n} W_{ij}\left|\left|\frac{F_i}{\sqrt{D_{ii}}} - \frac{F_j}{\sqrt{D_{jj}}}\right|\right|^2 + \frac{\mu}{2}\sum_{i=1}^{n}||F_i - Y_i||^2, \tag{4.18}$$

using a conjugate gradient approach. In this equation μ is a balancing term between the normalised $p = 2$ graph Laplacian and a data fidelity term. For CyleCluster we use hard pseudo-labels so the pseudo-labels for the unlabelled points are extracted by taking $\hat{y}_i = \arg\max_j F_{ij}$. Additionally, in CycleCluster we counter the problems of confirmation bias and class balancing by using a *class weight* $\zeta_i \in (0,1)\ \forall 1 \leq i \leq C$ to account for unbalanced pseudo-labels and an *entropy weight* $w_i \in (0,1)$ to encode pseudo-label confidence. We take the same approach as suggested by Iscen et al [106] so the class weight ζ_i is given by

$$\zeta_j := \left(\sum_{i=1}^{n_l}\mathbb{1}_{y_i=j} + \sum_{i=n_l+1}^{n}\mathbb{1}_{\hat{y}_i=j}\right)^{-1}, \tag{4.19}$$

and the entropy weight is given by

$$w_i := 1 - \frac{H(F_i)}{\log(C)}, \tag{4.20}$$

where $H(\cdot)$ is the Shannon entropy and the value $0 \leq w \leq 1$ is bounded due to the maximum of the Shannon entropy being equal to the logarithm of the number of classes. Therefore, the total semi-supervised loss is given by

$$\theta \leftarrow L_W(Z, \hat{Y}; \theta) := \frac{1}{n_l} \sum_{i=1}^{n_l} \zeta_{y_i} l_{ce}(f_\theta(x_i), y_i) + \frac{1}{n_u} \sum_{i=n_l+1}^{n} \zeta_{\hat{y}_i} \omega_i l_{ce}(f_\theta(x_i), \hat{y}_i) \tag{4.21}$$

where the pseudo-labels $\hat{Y}$ are kept fixed during the optimisation and l_{ce} is the cross entropy loss. The classification pseudo-labels are updated at the end of each epoch of training, just as the clustering pseudo-labels are.

Shared Framework

We now detail our overall optimisation. We first extract some baseline level of knowledge on the dataset by minimising a supervised loss over the initially labelled data Z_l. The supervised loss is given by

$$L_s(Z_L; \theta) := \frac{1}{n_l} \sum_{i=i}^{l} l_{ce}(f_\theta(x_i), y_i). \tag{4.22}$$

and we use stochastic gradient descent with a batch size b and optimise for 100 passes through the labelled data Z_l.

We then combine the two learning tasks defined in (5.16) and (4.16). Given that the cluster labels are vectors in $\mathbb{R}^K$ whilst the class labels are vectors in $\mathbb{R}^C$, combining them together is not trivial. In our approach we expand the dimensionality of our linear classifier to $g_\theta : \mathbb{R}^{d_p} \rightarrow \mathbb{R}^K$ as $K > C$. From this we use the full output of $g_\theta(x)$ to contrast against the cluster pseudo-label and the first C elements of $g_\theta(x)$ as the classification.

Given that our model can now be trained against both losses we detail our optimisation scheme. We use an iterative approach as shown in Fig 4.5. Firstly, we extract cluster and classification pseudo-labels $\tilde{Y}, \hat{Y}$. We then use stochastic gradient descent with a batch size b to sequentially optimise L_C and L_W. For L_C we sample uniformly from the entire dataset and train for one pass through the dataset. For L_W we use batches with b_l labelled images and b_u unlabelled images and train for one pass through the unlabelled data Z_u. Note that this means each labelled image is used multiple times per pass. We iteratively cycle through updating the pseudo-labels and training the model for a total of E epochs. For full clarity we give a high-level full algorithm for CycleCluster in Algorithm 1.

Algorithm 1 Training CycleCluster

Input Dataset Z with labelled samples $Z_l = \{x_i, y_i\}_{i=1}^{n_l}$ with C total classes and unlabelled samples $Z_u = \{x_i\}_{i=n_l+1}^{n}$, Model f_θ of composite functions t_θ, g_θ
Parameters: Number of epochs E, Batch size b, labelled batch size b_l, unlabelled batch size b_u.
for i = 1,2,....,100 **do**
  for $j = 1, .., \lfloor \frac{n_l}{b} \rfloor$ **do** ▷ Initial Supervised Baseline
    Batch $B_L = \{x_i, y_i\}_{i=1}^{b} \subset Z_l$
    $\theta \leftarrow L_s = \frac{1}{b} \sum_{i=1}^{b} l_{ce}(f_\theta(x_i), y_i)$
  end for
end for
for i = 1,..,E **do**
  $T = \{t_1, .., t_n\} = t_\theta(X)$ where $X = \{x_1, .., x_n\}$ ▷ Extract Feature Embeddings
  Perform K-means clustering and extract $\tilde{Y}$
  Construct Graph with Adjacency Matrix W
  Construct Degree and Labelling matrices D, Y
  Propagate Information via $Q(F)$
  $\hat{y}_i = \arg\max F_i \ \forall \ n_l + 1 \leq i \leq n$
  for $n_l + 1 \leq i \leq n$ **do**
    Calculate entropy weight $w_i := 1 - \frac{H(F_i)}{log(C)}$ ▷ H being Shannon Entropy
  end for
  for $1 \leq j \leq C$ **do**
    Calculate class weight $\zeta_j = \left(\sum_{i=1}^{n_l} \mathbb{1}_{y_i = j} + \sum_{i=n_l+1}^{n} \mathbb{1}_{\hat{y}_i = j} \right)^{-1}$
  end for
  for $i = 1, .., \lfloor \frac{n}{b} \rfloor$ **do** ▷ Clustering Regularisation
    Batch $B_C = \{x_i, \tilde{y}_i\}_{i=1}^{b} \subset \{X, \tilde{Y}\}$
    $\theta \leftarrow \frac{1}{b} \sum_{i=1}^{b} l_{ce}(f_\theta(x_i), \tilde{y}_i)$
  end for
  for $i = 1, .., \lfloor \frac{n - n_l}{b} \rfloor$ **do** ▷ Semi-Supervised Learning
    Batch $B_L = \{x_i, y_i\}_{i=1}^{b_l} \subset \{Z_l\}$, $B_U = \{x_i, \hat{y}_i\}_{i=1}^{b_u} \subset \{Z_u, \hat{Y}\}$
    $\theta \leftarrow \frac{1}{b_l} \sum_{i=1}^{b_l} \zeta_{y_i} l_{ce}(f_\theta(x_i), y_i) + \frac{1}{b_u} \sum_{i=1}^{b_u} \zeta_{\hat{y}_i} \omega_i l_{ce}(f_\theta(x_i), \hat{y}_i)$
  end for
end for

4.4.2 Results and Discussion

In this section, we detail the experiments that we conducted to test the performance of CycleCluster as well as the ablation results focusing on the nature of cluster regularisation. In turn we describe the chosen datasets, the evaluation protocol, parameter selection and finally present numerical and visual results.

Dataset Description

We evaluate our approach using three benchmarking datasets:

1. **CIFAR-10 / CIFAR-100** [126] is composed of 50k training images and 10k test images, which represent ten different image classes. Each image is an RGB image of resolution 32×32. We perform experiments using 50, 100, 200 and 400 labels per class which corresponds to 500, 1k ,2k and 4k total labels. CIFAR-100 is a more complicated version of CIFAR-10 in that it contains 100 different classes but has the same number of training and test images. We perform experiments using 40 and 100 labels per class, corresponding to 4k and 10k labels in total.

2. **Mini-ImageNet** [217]. Mini-ImageNet is a version of the popular ImageNet dataset [63] which is designed for the purpose of SSL [217]. It comprises 100 different classes each having 600 images of resolution 84×84 of which 500 are assigned to the training set and 100 to the test set. We perform experiments using 4K and 10K total labels.

For each dataset, we use the official data partition. Throughout this chapter we use the Top-1 error rate as the evaluation metric. The Top-1 error rate is defined as the percentage of test images for which the model's most confident estimate of the label is not correct. For the Cifar and Mini-Imagenet datasets this is a more useful comparison than the Top-5 error rate, the percentage of test images for which none of the model's five most confident estimates of the label are correct, as almost all methods achieve very high Top-5 error rates due to the small number of classes present in the datasets. As is standard practice in the area, we quote the mean error rate and standard deviation over different randomly generated splits: ten for CIFAR-10 and five for CIFAR-100 and Mini-ImageNet.

Evaluation Protocol

The goal of CycleCluster was to directly compare the effect of clustering regularisation against δ perturbation approaches. Therefore, to extract a clean comparison, we initially do not use heavy-data augmentation [197] or other added complexities such as MixUp

augmentation [245]. Instead we only use standard flip and crop augmentation and compare against approaches which share this same approach to data augmentation. We compare our work against: Ladder Networks [171], VAT [152], SSL-GAN [186], TSSDL [192], MT [205], LPDSSL [106] and ICT [216]. Subsequently, we demonstrate that augmentation can be easily combined with our approach to boost performance and combine our approach with RandAugment [58].

In addition to this comparison, we perform ablation experiments relating to the implementation of clustering regularisation including its full removal. We also experiment with a combination of our approach and MT, where we utilises MT only for optimising the semi-supervised loss, and use the Mean Teacher code provided by the original Mean Teacher approach [205].

Implementation Details

Architectures: For the CIFAR-10 and CIFAR-100 dataset we used a "13-layer CNN" network that has been used in previous works [130]. For Mini-Imagenet we use the ResNet-18 architecture [95]. We add an l_2 normalisation layer before the final fully connected layer and set the dropout rate to zero. All code was written in using the PyTorch library. For all experiments, we use 1 Nvidia P100 GPUs to train our models.

Parameters We split our parameters into several groups: clustering, semi-supervised learning and training parameters. *Clustering*: For all datasets clustering was performed for 1000 iterations using the K-means clustering algorithm. The value of K was dataset dependent and is specified in each experiment. *Semi-supervised*: For graph construction we use K-NN with $K = 50$ and $\mu = 0.01$. *Training*: For all datasets we train using stochastic gradient descent with momentum and l_2 weight regularisation with momentum = 0.9 and weight decay 2×10^{-4}. We train for 180 epochs with $l_0 = 0.05$ and an annealing finishing point of 210. For CIFAR-10 we use a batch size $b = 100$ with sub-batch sizes of $b_l = 50, b_u = 50$. For CIFAR-100 and MiniImagenet we use a batch size of $b = 128$ with pseudo-label batch sizes of $b_l = 88, b_u = 40$.

Results and Discussion

In this section we present the experimental results and complementary visualisations generated. Firstly we begin by comparing CycleCluster against several different δ-perturbation models which only use flip and crop augmentation. These methods offer a wide variety of the δ-perturbations used in the field. We test these approaches on the CIFAR-10, CIFAR-100 and MiniImageNet datasets. Note for this section we set the number of clusters

CIFAR-10				
	# LABELS			
METHOD	500	1k	2k	4k
SUPERVISED BASELINE	48.93±0.80	39.18±0.88	28.23±0.49	21.20±0.46
PERTURBATION BASED APPROACHES				
Ladder Networks [171]	–	–	–	20.40±0.47
VAT [152]	–	–	–	11.36±0.34
SSL-GAN [186]	–	21.83±2.01	19.61±2.09	18.63±2.32
TSSDL [192] †	–	21.13± 1.17	14.65± 0.33	10.90 ± 0.23
MT [205]	27.45 ± 2.64	21.55±1.48	15.73±0.31	12.31±0.2
ICT [216] †	–	19.56±0.56	14.35±0.15	11.19±0.14
LPDSSL [106] †	32.40 ± 1.80	22.02 ± 0.88	15.66±0.35	12.69±0.29
LPDSSL + MT [106] †	24.02 ± 2.44	16.93 ± 0.70	13.22±0.29%	10.61±0.28
LGA [107] †	–	–	–	12.91±0.15
LGA + VAT [186] †	–	–	–	12.06 ± 0.19
CLUSTERING BASED APPROACHES				
CycleCluster	**19.35 ± 2.52**	**14.76± 0.34**	**12.11 ± 0.40**	**10.52 ± 0.45**

Table 4.1 Comparison of CycleCluster to perturbation based DSSL methods on CIFAR-10. The error rate is reported. We denote by † error rates obtained by previous works.

.

CIFAR-100		
	# LABELS	
METHOD	4k	10k
SUPERVISED BASELINE	55.59 ± 0.91	40.84 ± 0.34
LDPSSL † [106]	46.20 ± 0.76	38.43 ± 1.88
MT † [205]	45.36 ± 0.49	36.08 ± 0.51
LDPSSL + MT † [106]	**43.73 ± 0.20**	35.92 ± 0.47
CycleCluster	45.19 ± 0.34 %	35.65 ± 0.50
CycleCluster+MT	44.34 ± 0.26	**34.98 ± 0.38**

MINI IMAGENET		
	# LABELS	
METHOD	4k	10k
SUPERVISED BASELINE	74.59 ± 0.90 %	60.17 ± 0.50
LDPSSL † [106]	70.29 ± 0.81	57.58 ± 1.47
MT † [205]	72.51 ± 0.22	57.55 ± 1.11
LDPSSL + MT † [106]	72.78 ± 0.15	57.35 ± 1.66
CycleCluster	**69.12 ± 1.05**	**54.27 ± 0.71**
CycleCluster+MT	**63.30 ± 0.29**	**53.47 ± 0.17**

Table 4.2 Comparison of CycleCluster to perturbation based approaches with DSSL methods on CIFAR-100 and Mini-ImageNet. The error rate is reported. We denote by † error rates obtained by previous works.

K equal to ten times the numbers of classes for CIFAR-10 and equal to the number of classes for CIFAR-100 and MiniImageNet so that each experiment used the same number of clusters, one hundred.

We present the comparison results for CIFAR-10 in Table 4.1, where we used all comparison methods and the "13-CNN" architecture. We see that out of all methods considered CycleCluster is the most accurate method across all label amounts. Furthermore, we can see by comparing our approach to LPDSSL, a purely graphical pseudo-labelling approach, that the inclusion of clustering based regularisation to a graphical approach offers great performance increases, especially at low label amounts where the small number of initial labels is a poor representation of the dataset.

CIFAR-10		
	# LABELS	
Method	1k	4k
Fully Supervised	39.189 ± 0.91	40.84 ± 0.34
CycleCluster N-RA	14.76 ± 0.34	10.52 ± 0.45
CycleCluster RA	8.52 ± 0.29	6.58 ± 0.18

MINIIMAGENET		
	# LABELS	
Method	1k	4k
Fully Supervised	74.59 ± 0.90	60.17 ± 0.50
CycleCluster N-RA	69.12 ± 1.05	57.82 ± 1.01
CycleCluster RA	56.36 ± 0.49	45.40 ± 0.37

Table 4.3 The effect of including strong augmentations in the form of one RandAugment [58] sample. The error rate is reported for CycleCluster without RandAugment (N-RA) and with RandAugment (RA). The experimental parameters used were the same as in the prior experiments. We report results for both CIFAR-10 and MiniImageNet and see a large increase in performance upon the inclusion of RandAugment.

For CIFAR-100 and MiniImageNet, we use a smaller number of comparison models and present those results in Table 4.2. For CIFAR-100 all methods used the "13-CNN" architecture and for MiniImageNet all methods use a ResNet-18 . For CIFAR-100 we find that our approach again performs well producing the lowest error rate for 10k labels but slightly below the performance of LDPSSL+MT on 4K labels. For Mini-ImageNet our method is by some margin the best method considered, showing that clustering can extract valuable information from even complex datasets. We would like to highlight the performance that our method combined with MT has on the Mini-ImageNet dataset. CycleCluster + MT achieves error rates of 9.48 and 3.88 better than LDPSSL+MT for 4k and 10k labels respectively. Comparing our results on CIFAR-100 and MiniImageNet it would appear that the benefit of clustering regularisation compared to consistency regularisation is very dataset dependent.

As recent approaches [197, 23] have shown, the inclusion of stronger augmentation techniques can lead to a dramatic performance increase in the semi-supervised setting. The inclusion of augmentation to our framework is trivial and can be done separately for both the clustering and classification loss. In our case we choose to add one sampled augmentation from RandAugment [58] in addition to our standard flip and crop augmentation whilst keeping all parameters the same. We give results on both CIFAR-10 and Mini-ImageNet for our augmented version in Table 4.3 and compare that to the baseline model. We see that the inclusion of data augmentation greatly increases the performance of CycleCluster across all datasets and label numbers, demonstrating that data augmentation can be successfully combined with clustering based regularisation.

Ablation Study

For clustering methods, the number of clusters K has to be provided as a prior parameter for most methods and choosing the number of clusters may be non-trivial. In this section

CIFAR-10				
	# LABELS			
METHOD	500	1k	2k	4k
SUPERVISED BASELINE	48.93 ± 0.80	39.18 ± 0.88	28.23 ± 0.49	21.20 ± 0.46
LP	32.21 ± 1.56	22.31 ± 0.78	15.63±0.45	12.63±0.32
$K = 10$	21.58 ± 1.73	15.86 ± 0.83	13.00 ± 0.30	10.73 ± 0.36
$K = 100$	20.94 ± 2.19	15.52 ± 0.88	12.79 ± 0.35	10.79 ± 0.45
$K = 300$	21.36± 0.99	16.98 ± 0.90	13.43 ± 0.66	11.28 ± 0.39
CIFAR-100				
	# LABELS			
METHOD	4k		10k	
SUPERVISED BASELINE	55.59 ± 0.91		40.84 ± 0.34	
LP †	46.20 ± 0.76		38.43 ± 1.88	
Clusters=100	45.19 ± 0.34 %		35.65 ± 0.52%	
Clusters=300	45.18 ± 0.49%		35.72 ± 0.21%	

Table 4.4 Ablation study on how changing the number of clusters K effects the final classification accuracy on the CIFAR-10 and CIFAR-100 dataset.

we investigated how dependent the performance of the model is on the selection of the number of clusters. Can a bad choice of K lead to non-convergence or is the model relatively invariant to the choice of K? To investigate this we ran an experiment where, for the CIFAR-10 and CIFAR-100 datasets, we used a range of K values and compared the change in Top-1 error rate. Additionally we created another variant of our model where we dropped the clustering task all together to create a purely graphical label propagation based model which we denote by LP. The results for this ablation are reported in Table 4.4.

Firstly we see that for all values of K, CycleCluster performs much better than a method which only uses graphical pseudo-labelling, demonstrating that clustering regularisation extracts meaningful information for a large range of K values. For both CIFAR-10 and CIFAR-100, we found that the performance increase was not dependent upon K. Showing that in general, the improvement in performance from using over-clustering regularisation is very robust to the value of K and *choosing the value of K is not a major problem in this framework.*

Whilst CycleCluster provided an alternative to the implementation of domain dependent and complex data perturbations there are still several areas for improvement. The choice of having one classification layer g_θ for both tasks may be a bottleneck in performance and alternate neural network architectures, such as having a classification layer for each task, may further improve performance. Secondly, considering a more complex joint

optimisation model may be superior to the current sequential approach. Of course the downside of such an approach being an increased computational time.

However, as will be further explained in the following section, consistency regularisation approaches have been superseded in favour of high-performance pseudo-label methods [197] and it is this domain we turn to next.

4.5 LaplaceNet

There are two dominant families of approaches in deep semi-supervised learning: *consistency regularisation* and *pseudo-labelling*. Due to its ease of use and superior performance, pseudo-labelling methods have become the go-to-approach. Many recent state-of-the-art approaches to deep pseudo-labelling have focused on costly optimisation schemes [22] involving multiple loss terms or large numbers of optimisation steps. Several have focused upon the application of strong data augmentation [197, 236] to improve performance. Other works have included numerous technical tricks [8] to reduce confirmation bias.

However, we see that by comparing the first work on deep semi-supervised learning [131] to the latest works that the underlying learning task has been kept almost identical. There are two major drawbacks to this style of improvement: it is hard to say which changes in methodology have contributed to increased performance and *it becomes harder to deploy deep semi-supervised learning*. For example, in the popular work of [197] it was found that slightly changing the data augmentation led to model collapse and in the work of [24] slight changes to pseudo-labelling hyperparameters led to massive drops in performance. Deploying semi-supervised learning to complex domains, where labelled data is harder to acquire, becomes exponentially harder as we add layers of complexity to our models.

In this section we present LaplaceNet [189], a state of the art approach for semi-supervised image classification. The core objective of LaplaceNet was to achieved state-of-the-art results whilst minimising model complexity. We aimed to use the rigour and strength of graph based pseudo-labels to remove the need for several technical tricks. Additionally, we wanted to mathematically explore the use of data augmentation and see if we could come up alternative strategies that led to greater performance and reduced sensitivity. The major highlights of this research are

1. The exploration, implementation and validation of a multi-sampling approach to data augmentation.

2. A demonstration of the benefits of graphical based pseudo-labelling over naive neural network predictions.

3. State-of-the-art results on several benchmark datasets whilst greatly reducing model complexity.

The rest of the section is structured as follows. We firstly explore the methodology of LaplaceNet. We then discuss the evaluation protocol we use to evaluate our approach before presenting the experiment results and discussion which includes a detailed ablation on data augmentation and graph-based pseudo-labels.

4.5.1 Methodology

This section details the methodology of our proposed semi-supervised method LaplaceNet. We firstly cover the generation of pseudo-labels before then discussing the optimisation of the model and our multi-sample augmentation approach. Additionally we present a full algorithm of LaplaceNet for completeness. Note that for notation we use exactly the same problem statement as was given in the section on CycleCluster.

Pseudo-labels Generation

The generation of pseudo-labels in LaplaceNet is very similar to the method used in CycleCluster. Both the metric and sparsification method are identical, leading to the same Graph $\mathscr{G} = (\mathscr{V}, \mathscr{E}, \mathscr{W})$. Additionally, we degree normalise the weight matrix and optimise over the same objective (4.18) using a conjugate gradient approach to give the prediction matrix F. However, unlike in CycleCluster, we take a different approach to both the problem of confirmation bias and class imbalances.

For confirmation bias we found that entropy weighting as used in CycleCluster actually harmed performance if used in conjunction with data augmentation. Instead we found that no weighting was the best weighting and treated all pseudo-labels equally. For class imbalances, we found that the weighting approach of Iscen et al actually made the performance of the model worse than leaving the predictions as is. An alternate approach to counter class imbalance is distribution alignment [22], which enforces the distribution of the pseudo-label predictions to match some given prior distribution. The implementation of this idea by ReMixMatch worked by re-weighting the network prediction at inference time which is unsuitable for graphical methods where the labels for the entire dataset are produced in one go.

Algorithm 2 Smooth Distribution Alignment

Input Pseudo-label Prediction $F \in \mathbb{R}^{n \times C}$, Prior Distribution $D \in \mathbb{R}^{C}$, labelled and unlabelled indexes $L = \{l_i\}_{i=1}^{n_l}$ and $U = \{u_i\}_{i=1}^{n_u}$ and max iteration T
Output Adjusted Pseudo-label Prediction F.
for $t_i = 1$, t_i++, while $t_i < T$ **do**
    Initialise $D_U \in \mathbb{R}^{C}$ to zero.
    for $u_i \in U$ **do**
        $D_U[\text{argmax}_j F[u_i]]$ += $\frac{1}{n_u}$ ▷ Obtain class distribution of pseudo-labels
    end for
    $R = D/D_u$
    $R[R > 1.01] = 1.01$ and $R[R < 0.99] = 0.99$ ▷ Clip values to smooth deformation
    for $c_i = 1$, c_i++, while $c_i < C$ **do**
        $F[U, c_i]$ *= $R[c_i]$ ▷ Promote the prediction of under-sampled classes
    end for
    for $i = 1$, i++, while $i \leq n$ **do**
        $F[i] = F[i] / \sum_{j=1}^{C} F[i, j]$
    end for
end for

Instead, we propose a novel smoother version of distribution alignment which we apply after minimising (4.18). We give a full algorithm for this in Algorithm2. The algorithm is an iterative approach which smoothly deforms the pseudo-label predictions F by the ratio R between the prior distribution D and the pseudo-label distribution of the unlabelled points D_U. Thereby promoting the prediction of underrepresented classes and vice versa. To ensure the deformation is smooth we clip the range of R values to be close to one. We show in the experimental section that this approach improves the performance of the model.

Optimising the Model

For initialisation purposes, we quickly extract some baseline knowledge from the dataset by minimising a supervised loss L_{sup}, for one hundred passes through the labelled set Z_l. This supervised loss reads:

$$L_{\text{sup}} = \frac{1}{b} \sum_{i=1}^{b} l_{ce}(f(x_i), y_i), \tag{4.23}$$

where b is the batch size and l_{ce} is the cross entropy loss. We then begin our main learning loop which alternates between updating the pseudo-label predictions and minimising

the semi-supervised loss L_{ssl} for one epoch, where we define one epoch to be one pass through the unlabelled data Z_u.

In the deep semi-supervising setting, particularly in the current state-of-the-art (SOTA) [197] [23], several works seek to minimise a semi-supervised loss $\hat{L}_{ssl}$ composed of two or more terms, one each for the labelled and unlabelled data points and potentially others covering entropy minimisation etc. A typical loss of this form is given below.

$$\hat{L}_{ssl} = \frac{1}{n_l}\sum_{i=1}^{n_l} l_s(f(x), y) + \eta\frac{1}{n_u}\sum_{i=1}^{n_u} l_s(f(x), \hat{y}) + ..., \tag{4.24}$$

where η is a balancing parameter between the terms. For our approach we wanted to strip away as much complexity from the loss function as possible in an effort to see what elements are required for good performance. We move away from using a composite loss and instead only use the standard supervised loss which has worked so well in supervised image classification. To include our unlabelled data we use composite batches of size b which are made up of b_l labelled samples and b_u unlabelled samples to which we have assigned a pseudo-label $\hat{y}$. Our semi-supervised loss L_{ssl} for each batch is given by:

$$L_{ssl} = \frac{1}{b}\sum_{i=1}^{b} l_{ce}(f(x_i), y_i). \tag{4.25}$$

Note that in (4.25) y_i may be a ground truth label or a pseudo-label. What is remarkable about this loss is its simplicity. There is no confidence thresholding of the pseudo-labels, additional weighting parameters, no consistency based terms or other regularisations. Instead we rely upon the energy-based graphical approach to pseudo-label estimation.

The cycle of label estimation and optimisation runs for a total of S optimisation steps and the fully trained model is then tested on the relevant testing set. Note that we do use Mixup [245] on both L_{sup} and L_{ssl} with a beta distribution parameter α. In Algorithm 3, we give an overview of training our model for S optimisation steps.

Multi-sampling augmentation

Since the initial work of Xie [236], several approaches have used strong augmentation to boost performance in the semi-supervised setting [103, 22, 197] , with each work having a different way of including augmentation to their framework. Very recent works [103, 22] have begun using multiple augmented versions of the same unlabelled image. As yet there is no motivation for why this multiple sampling idea is preferable to alternatives such as larger batch sizes or running the code for more steps.

Algorithm 3 Training LaplaceNet

Input Dataset Z with labelled samples $Z_l = \{x_i, y_i\}_{i=1}^{n_l}$ with C total classes and unlabelled samples $Z_u = \{x_i\}_{i=n_l+1}^{n}$, Model f_θ of composite functions t_θ, g_θ
Parameters: Number of optimisation steps S, Batch size b, labelled batch size b_l, unlabelled batch size b_u.
for i = 1,2,....,100 **do**
  for $j = 1,..,\lfloor \frac{n_l}{b} \rfloor$ **do** ▷ Initial Supervised Baseline
    Batch $B_L = \{x_i, y_i\}_{i=1}^{b} \subset Z_l$
    $\theta \leftarrow L_{sup} = \frac{1}{b}\sum_{i=1}^{b} l_{ce}(f_\theta(x_i), y_i)$
  end for
end for
Set current step $s_i = 0$.
while $s_i < S$ **do**
  $T = \{t_1,..,t_n\} = t_\theta(X)$ where $X = \{x_1,..,x_n\}$ ▷ Extract Feature Embeddings
  Construct Graphical Representation
  Propagate Information via $Q(F)$
  Distributed Alignment on F
  $\hat{y}_i = \arg\max F_i \ \forall \ n_l + 1 \leq i \leq n$
  for $i = 1,..,\lfloor \frac{n-n_l}{b} \rfloor$ **do** ▷ Semi-Supervised Learning
    Batch $B_L = \{x_i, y_i\}_{i=1}^{b_l} \subset \{Z_l\}$, $B_U = \{x_i, \hat{y}_i\}_{i=1}^{b_u} \subset \{Z_u, \hat{Y}\}$
    $\theta \leftarrow L_{ssl} = \frac{1}{b}\sum_{i=1}^{b} l_{ce}(f(x_i), y_i)$
  end for
end while

In this section we offer a theoretical motivation for why multi-sampling improves generalization along with a mathematical bound on its performance gain. With this knowledge in mind we provide a simple generic method for including augmentation averaging into our SSL framework and demonstrate this approach increases accuracy and reduces the sensitivity of the model to data augmentation. We will view a given data augmentation strategy as a set of transformations $A = \{a_1,..,a_m\}$ where each transformation is a label preserving function $a : \mathscr{X} \rightarrow \mathscr{X}$. Doing so produces an estimated vicinity distribution of

$$dP_{x_i}(\mathbf{x}) = \frac{1}{m}\sum_{j=1}^{m} \delta_{a_j(x_i)}(\mathbf{x}), \tag{4.26}$$

which is a set of Dirac deltas located at all the possible transformations of x_i. Due to the computational in-feasibility of integrating over $P_{x_i}(\mathbf{x})$, almost all methods only sample once from this vicinity for each point

$$R_{vic}(f) = \frac{1}{n}\sum_{i=1}^{n} l(f(x_i), y)\delta_{a_j(x_i)}(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n} l(f(a_j(x_i)), y). \tag{4.27}$$

However, we argue that such a simple implementation might not extract the full information present in the vicinity distribution generated by the augmentation. If we want to encourage our model output to be invariant over $dP_{x_i}(\mathbf{x})$, and as a consequence produce a more generalisable model, we need to perform a multi-sample approach. To justify this, we consider the full vicinal risk:

$$R_{vic}(f) = \frac{1}{n}\sum_{i=1}^{n}\int l(f(x_i), y_i) dP_{x_i}(x) = \frac{1}{n}\frac{1}{m}\sum_{i=1}^{n} l(f(x_i), y_i)\sum_{j=1}^{m}\delta_{a_j(x_i)}(\mathbf{x}), \tag{4.28}$$

such that the full vicinal risk is

$$R_{vic}(f) = \frac{1}{n}\frac{1}{m}\sum_{i=1}^{n}\sum_{j=1}^{m} l(f(a_j(x_i), y_i) = \frac{1}{n}\sum_{i=1}^{n} R_{vic}(f(x_i)). \tag{4.29}$$

$R_{vic}(f(x_i))$ is the vicinity loss for each data point x_i over the entire set of transformations. If we want to minimise $R_{vic}(f)$ then we should minimise $R_{vic}(f(x_i))$ for each data point. However, does the current strategy of one sampling from $R_{vic}(f(x_i))$ achieve this? To see instead that a multi-sample approach helps us we use Hoeffding's inequality [101]. Hoeffding's inequality provides us with a probability bound that the sum of bounded independent random variables deviates from its expected value by more than a certain amount. Let $Z_1, .., , Z_{n_a}$ be a sequence of i.i.d random variables. Assume that $\mathbb{E}[Z] = \mu$ and $\mathbb{P}[a \leq Z_i \leq b] = 1$ for every i. Then, by Hoeffding's inequality, for any $\epsilon > 0$, one has:

$$\mathbb{P}\left[\left|\frac{1}{n_a}\sum_{i=1}^{n_a} Z_i - \mu\right| > \epsilon\right] \leq 2\exp(-2n_a\epsilon^2/(b-a)^2). \tag{4.30}$$

We can rewrite (4.30) in context of the previously defined vicinity distributions, assuming for now the model output is i.i.d for different augmentations, by replacing $Z_1, .., , Z_{n_a}$ with n_a samples from $A: f(a_1(x)), f(a_2(x)), .., f(a_{n_a}(x))$ then we have the following concentration bound

$$\mathbb{P}\left[\left|\frac{1}{n_a}\sum_{j=1}^{n_a} l(f(a_j(x_i)), y_i) - R_{vic}(f(x_i))]\right| > \epsilon\right] \leq 2\exp(-2n_a\epsilon^2/b^2). \tag{4.31}$$

As we increase n_a, we converge in probability to the desired estimate $R_{vic}(f(x_i))$ for each data point and subsequently we should obtain a lower value of $R_{vic}(f)$ meaning that the model output will fluctuate less over the augmentation set A. Furthermore, we can see

Labelled Transform	Unlabelled Transform
Random Horizontal Flip Random Crop and Pad	
RandAugment Sample -	RandAugment Sample RandAugment Sample
CutOut Normalisation	

Table 4.5 The augmentation transformations used for labelled and unlabelled data. For normalisation we use the official normalisation parameters that are provided alongside the dataset.

that the probability is bounded by an exponent whose power is $\propto -n_a$. Therefore, as we increase n_a the rate of decrease for the bound also decreases, making the first few samples far more important than later ones. This result explains prior behaviours reported but not reasoned in past papers such as [22]. When using n_a samples the computational complexity increases as $O(n_a)$ but as there should be diminishing returns for increasing n_a it should only be necessary to use n_a values slightly above one.

Augmentation Implementation

With this result in mind, we now detail how augmentation is implemented in LaplaceNet to create the set of possible transformations A. Similarly to other approaches we use two different augmentation strategies: one for labelled data and another for unlabelled data. In Table 4.5 we give the full workflow of transformations used in LaplaceNet.

Note that the sequence of augmentations shown in Table 4.5 randomly generates one transformation a from the set of all possible augmentations A. Therefore, in our augmentation averaging approach, when we want to use n_a transformations for each data point we repeat the whole sequence of augmentations n_a times every time an image is called.

Strong Augmentations

Unlike prior works [197] which reported model divergences, we apply strong augmentations to both labelled and unlabelled data. For strong augmentations we use RandAugment [58], and CutOut augmentation [65]. Our implementation of RandAugment differs slightly to the original paper but closely aligns to other semi-supervised implementations. In our implementation, we predefined a set of suitable transformations with each having a range of magnitude values. Rather than optimise a specific magnitude parameter, we

Transformation	Description	Range
RandAugment Transformations		
Autocontrast	Maximises the image contrast by setting the darkest (lightest) pixel to black (white)	—
Brightness	Adjusts the brightness of the image. $B = 0$ returns a black image $B = 1$ returns the original image	$B \in [0.05, 0.95]$
Color	Adjusts the colour balance of the image. $C_l = 0$ returns a black and white image. $C_l = 1$ returns the original image.	$C_l \in [0.05, 0.95]$
Contrast	Controls the contrast of the image. $C_o = 0$ returns a gray image. $C_o = 1$ returns the original image.	$C_o \in [0.05, 0.95]$
Equalise	Equalises the image histogram.	—
Identity	Returns the original image.	—
Posterise	Reduces each pixel to B bits.	$B \in [4, 8]$
Rotate	Rotates the image by θ degrees.	$\theta \in [-30, 30]$
Sharpness	Adjusts the sharpness of the image, where $S = 0$ returns a blurred image $S = 1$ returns the original image.	$S \in [0.05, 0.95]$
Shear X	Shears the image along the horizontal axis with rate R.	$R \in [-0.3, 0.3]$
Shear Y	Shears the image along the vertical axis with rate R	$R \in [-0.3, 0.3]$
Solarize	Inverts all pixels above a threshold value of T	$T \in [0, 1]$
Translate X	Translates the image horizontally by (λ×image width) pixels.	$\lambda \in [-0.3, 0.3]$
Translate Y	Translates the image vertically by (λ×image height) pixels	$\lambda \in [-0.3, 0.3]$
CutOut Augmentation		
CutOut	Sets a random square patch of side-length (L×image width) pixels to grey	$L \in [0, 0.5]$

Table 4.6 List of Transformations used in our application of RandAugment as well their description and magnitude range. Additionally, we list the CutOut transformation used at the end of RandAugment sampling.

sample from this range when each respective transformation is called. In Table 4.6 we detail this pool of transformations. When RandAugment is called upon an image, n_r transformations are sampled and applied sequentially to the image. For unlabelled images we use $n_r = 2$ whilst for labelled images we use $n_r = 1$. After RandAugment is used we perform Cutout augmentation as a final step.

4.5.2 Implementation and Evaluation

In this section we detail the implementation of LaplaceNet, including hyper-parameter values and training schemes, and the evaluation protocol we use to measure our model's performance and compare against the current state-of-the-art.

Dataset Description

We use three image classification datasets: CIFAR-10 and CIFAR-100 [126] and Mini-ImageNet [217] to benchmark our approach. Following standard protocol, we evaluate our method's performance on differing amounts of labelled data for each datasets. We refer back to the dataset description in Section 4.4.2 for further information.

PARAMETER	CIFAR-10	CIFAR-100	Mini-ImageNet
α	1.0	0.5	0.5
μ	0.01	0.01	0.01
k	50	50	50
S	2.5×10^5	2.5×10^5	2.5×10^5
b	300	100	100
b_l	48	50	50
l_r	0.03	0.03	0.1
n_m	0.9	0.9	0.9
ω	5×10^{-4}	5×10^{-4}	5×10^{-4}
n_a	3	3	3

Table 4.7 List of hyperparameters used for LaplaceNet across the CIFAR-10/100 and Mini-Imagenet datasets.

Implementation Details

Architectures For a fair comparison to older works we use the "13-CNN" architecture [205] and for comparison to recent state-of-the-art works we use a WideResNet (WRN) 28-2 and a WRN-28-8 [241] architecture. We additional use a ResNet-18 [220] for Mini-Imagenet. For all models we set the drop-out rate to 0. For the "13-CNN" we add a l_2-normalisation layer to the embedding function. **Training Details**: We train with stochastic gradient descent (SGD) using Nesterov momentum n_m with value 0.9 and weight decay ω with value 0.0005. We use an initial learning rate of $l_r = 0.3$ and use $S = 250000$ optimisation steps in total. We utilise a cosine learning rate decay such that the learning rate decays to zero after 255000 steps. We do not make use of any EMA model averaging. **Parameters** We list the parameter values used in Table 4.7. Most parameter values are common parameter settings from the deep learning field and are not fine-tuned to our application. Being able to work with reasonably generic parameters is well suitable to the task of SSL where using fine-tuning over validation sets is often impossible in practical applications.

Comparison Methods

We evaluate the performance of LaplaceNet on the CIFAR-10/CIFAR-100 and Mini-Imagenet datasets and compare against the current SOTA models for semi-supervised learning. For ease of comparison, we split the current SOTA into two groups.

1. Methods which used the 13-CNN architecture [205]: Π-Model [130], Mean Teacher(MT) [205], Virtual Adversarial Training (VAT) [152], Label Propagation for Deep Semi-Supervised Learning (LP) [106], Smooth Neighbors on Teacher Graphs (SNTG) [145], Stochastic Weight Averaging(SWA) [11], Interpolation Consistency Training (ICT) [216], Dual Student [120], Transductive Semi-Supervised Deep Learning(TSSDL)

Dataset	CIFAR-10				CIFAR-100	
Method	500	1000	2000	4000	4000	10000
Supervised Baseline	37.12 ± 0.89	26.60 ± 0.22	19.53 ± 0.12	14.02 ± 0.10	53.10 ± 0.34	36.59 ± 0.47
Consistency Based Approaches						
Π-Model	-	-	-	12.36 ± 0.31	-	39.19 ± 0.36
MT†	27.45 ± 2.64	21.55 ± 1.48	15.73 ± 0.31	12.31 ± 0.20	45.36 ± 0.49	36.08 ± 0.51
VAT	-	-	-	11.36 ± 0.34	-	-
MT-LP	24.02 ± 2.44	16.93 ± 0.70	13.22 ± 0.29	10.61 ± 0.28	43.73 ± 0.20	35.92 ± 0.47
SNTG	–	18.41±0.52	13.64±0.32	9.89±0.34	–	37.97±0.29
MT-fast-SWA	-	15.58 ± 0.12	11.02 ± 0.12	9.05 ± 0.21	-	34.10 ± 0.31
MT-ICT	-	15.48 ± 0.78	9.26 ± 0.09	7.29 ± 0.02	-	-
Dual Student	–	14.17±0.38	10.72±0.19	8.89±0.09	–	32.77±0.24
Pseudo-Labelling Approaches						
TSSDL†	-	21.13 ± 1.17	14.65 ± 0.33	10.90 ± 0.23	-	-
LP†	32.40 ± 1.80	22.02 ± 0.88	15.66 ± 0.35	12.69 ± 0.29	46.20 ± 0.76	38.43 ± 1.88
DAG	9.30 ± 0.73	7.42 ± 0.41	7.16 ± 0.38	6.13 ± 0.15	37.38 ± 0.64	32.50 ± 0.21
Pseudo-Label Mixup	8.80 ± 0.45	6.85 ± 0.15	-	5.97 ± 0.15	37.55 ± 1.09	32.15 ± 0.50
LaplaceNet †	**5.68 ± 0.08**	**5.33 ± 0.02**	**4.99 ± 0.12**	**4.64 ± 0.07**	**31.64 ± 0.02**	**26.60 ± 0.23**

Table 4.8 Top-1 error rate on the CIFAR-10/100 datasets for our method and other methods using the 13-CNN architecture. We denote with † experiments we have ran.

[192], Density-Aware Graphs (DAG) [132] and Pseudo-Label Mixup [8]. Unfortunately, due to the natural progress in the field, each paper has different implementation choices which are not standardised. Despite this, comparisons to this group are still useful as a barometer for model performance.

2. Recent methods which used the WRN [241] (MixMatch [23], FixMatch (RandAugment variant) [197] and UDA [236]). To guarantee a fair comparison to these techniques, and as suggested by [159], we used a shared code-base for UDA and FixMatch which reimplemented the original baselines. Additionally we then ensured UDA and FixMatch used the same model code, the same optimiser with the same parameters, the same number of optimisation steps and the same RandAugment implementation as our approach.

For each dataset we use the official train/test partition and use the Top-1 error rate as the evaluation metric. For each result we give the mean and standard deviation over five label splits.

4.5.3 Results and Discussion

In this section, we discuss the experiments we performed to evaluate and compare LaplaceNet against the current state-of-the-art (SOTA) in deep semi-supervised learning. Additionally, we detail several ablation experiments which explore the benefits of graph-based pseudo-labels versus typical network produced pseudo-labels and studies the effect

DATASET	CIFAR-10			CIFAR-100	
METHOD	500	2000	4000	4000	10000
OTHER METHODS					
MixMatch	9.65 ± 0.94	7.03 ± 0.15	6.34 ± 0.06	—	—
SAME CODEBASE					
UDA †	6.88 ± 0.74	5.61 ± 0.16	5.40 ± 0.19	36.19 ± 0.39	31.49 ± 0.19
FixMatch(RA) †	5.92 ± 0.11	5.42 ± 0.11	5.30 ± 0.08	34.87 ± 0.17	30.89 ± 0.18
LaplaceNet †	**5.57 ± 0.60**	**4.71 ± 0.05**	**4.35 ± 0.10**	**33.16 ± 0.22**	**27.49 ± 0.22**

Table 4.9 Top-1 error rate for CIFAR-10/100. All methods, except MixMatch, are tested using the same code-base and use the same model code, the same optimiser (SGD) with the same optimisation parameters, the same number of optimisation steps and the same RandAugment implementation. We denote with † experiments we have ran.

METHOD	4000	10000
Supervised Baseline	66.04 ± 0.32	52.89 ± 0.33
Consistency Regularisation Methods		
MT	72.51 ± 0.22	57.55 ± 1.11
MT-LP	72.78 ± 0.15	57.35 ± 1.66
Pseudo-Label Methods		
LP	70.29 ± 0.81	57.58 ± 1.47
Pseudo-Label Mixup	56.49 ± 0.51	46.08 ± 0.11
LaplaceNet	**46.32 ± 0.27**	**39.43 ± 0.09**

Table 4.10 Top-1 error rate for Mini-ImageNet. We compare against methods which have used an identical ResNet-18 architecture.

of a multiple-sample augmentation averaging approach on both network performance and network sensitivity.

Firstly, we test our model on the less complex CIFAR-10 and CIFAR-100 datasets. In Table 4.8, we compare LaplaceNet against the first group of approaches which used the 13-CNN network. Our approach, by some margin, produces the best results on CIFAR-10 and CIFAR-100 and represents a new SOTA for pseudo-labels methods. We obtain a lower error rate using 500 labels than the recent work of Arazo et al [8] obtain using 4000 labels. For CIFAR-100 LaplaceNet is a full 6% more accurate than any other approach and the first method to achieve an error rate below 30% on CIFAR-100 using 10k labels. In Table 4.9 we compare against the second group of methods using the WRN-28-2 network. LaplaceNet is again the best performing method, outperforming the recent works of UDA [236] and FixMatch [197]. In particular we find a significant increase in performance on the more complex CIFAR-100 dataset and beat the other considered methods by more than 3% with 10k labels.

DATASET	CIFAR-10		CIFAR-100	
MODEL	500	4000	4000	10000
WRN-28-2	5.57 ± 0.60	4.35 ± 0.10	33.16 ± 0.22	27.49 ± 0.22
WRN-28-8	3.81 ± 0.37	2.87 ± 0.18	26.61 ± 0.10	22.11 ± 0.23

Table 4.11 The effect on Top-1 error rate by scaling up the neural network in size from a WRN-28-2 to a WRN-28-8 on the CIFAR-10/100 datasets.

To test the performance of LaplaceNet on a more complex dataset, we evaluate our model on the Mini-ImageNet dataset, which is a subset of the well known ImageNet dataset and in Table 4.10 we compare our results against all others methods which have used this dataset. Once again, we find our method performs very well, producing an error rate a 10% and 7% better than any other method on 4k and 10k labelled images respectively. Demonstrating our approach can be applied to complex problems in the field. We are more than 20% more accurate that the nearest graphical approach (LP).

To measure how performance scaled with increasing network size we used an WRN-28-8(26 million parameters) architecture and and compared that to the WRN-28-2(1.6 million parameters) architecture in Table 4.11. Unsurprisingly, we achieved a large performance improvement using a WRN-28-8 on both CIFAR-10 and CIFAR-100, with an 2.87 error rate on CIFAR-10 using 4k labels and an 22.11% error rate on CIFAR-100 using 10k labels.

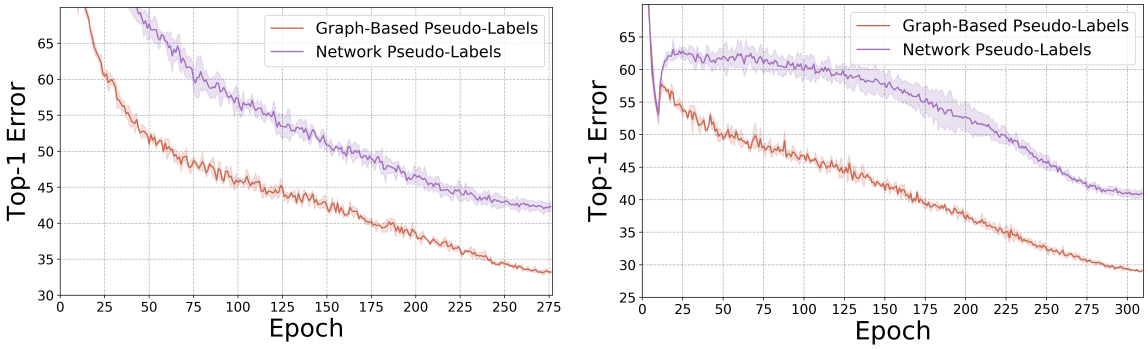
Graph Based Pseudo-Labels

Many pseudo-label based techniques [197] [8] have produced state-of-the-art results using pseudo-labels generated directly by the network rather than using an energy-based approach such as label propagation on a constructed graph, which is computationally more complex but has more mathematical reasoning and intuitive regularisation. Therefore, in this section we examine whether there is an advantage in using a graph based approach? To test the importance of graph based pseudo-labels, we created two variants of LaplaceNet, both without distribution alignment and with $n_a = 1$.

1. The pseudo-labels are generated directly from the network predictions.
   $\hat{y}_i = \text{argmax}\, f(x_i) \;\forall\; i > l$

2. The pseudo-labels are generated from the graph, as in (4.18).
   $\hat{y}_i = \text{argmax}_j\, F_{ij} \;\forall\; i > l$

We then compared the Top-1 error rate of these two variants on the CIFAR-100 dataset, see Fig 4.6. The graph-variant greatly outperformed the direct prediction variant, emphasising the clear advantage that graphically produced pseudo-labels have. What is



(a) (b)

Fig. 4.6 Experimental comparison of the effect of using pseudo-labels produced in a graphical framework versus pseudo-labels generated by the neural network on the Top-1 error rate on the CIFAR-100 dataset ((a) 4k and (b) 10k labelled images) with the 13-CNN network. Using graphically produced pseudo-labels we achieve a much higher accuracy than using the network predictions

contributing to this advantage? As an energy-based approach, propagation on the graph incorporates information on the global structure of the data, whilst the network is making a purely local decision at each point. Arazo et al [8] showed that naive network based pseudo-label approach could not generate an accurate solution for the "two moons" toy dataset, despite the fact that this problem has been solved by graphical methods for some time [254]. Thus demonstrating that purely local decisions are detrimental to accuracy when the global structure of data isn't taken into account.

Augmentation Averaging

In this Chapter we theoretically considered an augmentation averaging approach to further improve semi-supervised models. In this section, we present the experimental verification of our theoretical predictions as well as comparing to alternative techniques. To test the effect of augmentation averaging we ran our approach on the CIFAR-100 dataset using the 13-CNN network for a range of values $n_a = [1,3,5]$. Additionally we compared the changed caused by augmentation averaging to the more common approaches of scaling the batch size b and labelled batch size b_l by $[1,3,5]$. To quantify the effect of a given change we use two measures: the common choice of Top-1 error and an augmentation invariance measure which we define in this paper. Augmentation invariance measures the extent to which the classifier's performance changes under data augmentation. Given an augmenation function $u : \mathscr{X} \to \mathscr{X}$ and a classifier f_θ the augmentation invariance V with respect to a dataset Z made up of n point-label pairs $Z = \{x_i, y_i\}_{i=1}^n$ is given by

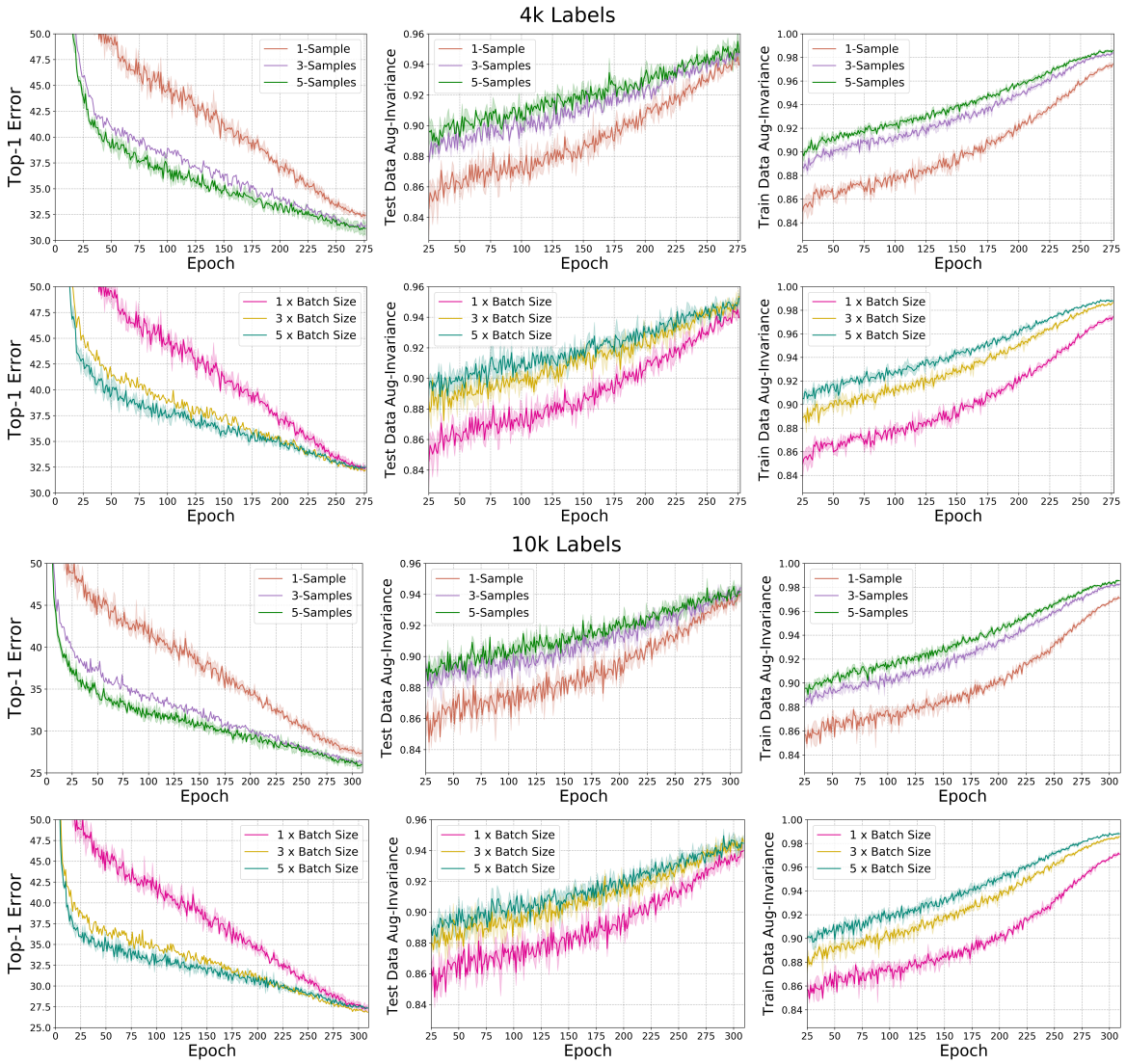


Fig. 4.7 A comparison on the effect of increasing batch size versus increasing the number of augmentation samples on Top-1 error rate, test data augmentation invariance and training data augmentation invariance for the CIFAR-100 dataset for both 4k and 10k labelled images. Increasing the amount of augmentation averaging decreased the error rate whilst also decreasing the sensitivity of the model's output predictions to augmented data. Increasing the batch size had a similar effect on the model's sensitivity, but it offered no improvement to model accuracy.

$$V_Z = \frac{\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{\operatorname{argmax} f_\theta(u(x_i))=y_i}}{\frac{1}{n}\sum_{i=1}^{n} \mathbb{1}_{\operatorname{argmax} f_\theta(x_i)=y_i}}, \tag{4.32}$$

which can be viewed as the performance ratio with and without data augmentation and as such we would expect $V_Z \in [0,1]$. We consider both the augmentation invariance

	CIFAR-100	
MODEL	4k	10k
Baseline	32.41 ± 0.25	27.37 ± 0.20
COMPONENT REMOVED		
RandAugment	44.43 ± 0.66	34.75 ± 0.23
Distribution Alignment	33.26 ± 0.24	29.07 ± 0.07
MixUp	33.74 ± 0.26	28.02 ± 0.20

Table 4.12 The effect of removing individual components from the baseline model on Top-1 error rate for CIFAR-100 on the 13-CNN network.

of our model with respect to the fully labelled training and test data in order to give a full picture of the model's invariance. In Fig 4.7 we present our findings. As previously theorised, we find that increasing the number of augmentation samples decreased the sensitivity of the model's predictions to augmentation on both the training and test data. An almost identical effect was found by scaling the batch size. However, the major difference between the two is their effect on Top-1 error rate. We found scaling the batch size offered no improvement to Top-1 error, in-fact the largest batch size offered the worst outcome, whilst increasing the number of augmentation samples noticeably improved the model's accuracy. Additionally as theorised, we see that the gain in performance from $n_a = 1 \rightarrow 3$ is much greater than $n_a = 3 \rightarrow 5$, supporting our statements regarding the exponential bound in probability. These results suggest that scaling the number of augmentation samples could be a great option for semi-supervised models using suitable strong augmentations.

4.5.4 Component Evaluation

As LaplaceNet combines several different techniques, we test the importance of strong augmentation, distribution alignment and MixUp to the overall accuracy of the model. We create a baseline model ($n_a = 1$) and then remove each component one at a time and test the performance on the CIFAR-100 dataset, see Table 4.12. Whilst the removal of each component decreased the performance of the model, it is clear the most crucial component to model performance is strong augmentation and removing it drastically reduces model accuracy. However, unlike other works [8] we find that whilst MixUp [245] offers a small advantage, is it not critical for composite batch pseudo-label approaches. This may be due to the advantages of graph-based approaches overcoming the flaws of naive neural network predictions.

Model	Computational Time (Hours)
Baseline	7.52 ± 0.04
Component Removal	
No Distribution Alignment	6.18 ± 0.01
No Strong Augmentation	5.84 ± 0.03
No Graphical propagation	6.32 ± 0.01
Model Scaling	
3×-Batch-size	12.28 ± 0.03
5×-Batch-size	17.23 ± 0.06
3×-Samples	12.88 ± 0.01
5×-Samples	18.14 ± 0.11

Table 4.13 Computational time taken for our approach using 4k labelled images on the CIFAR-100 dataset using the 13-CNN architecture. We provide the time taken for a number of different settings used in the results section. All experiments were performed using one NVIDIA P100 GPU.

Computational Time

To give clarity on the how long our code takes to run we provide the computational run times of LaplaceNet on the CIFAR-100 dataset using the 13-CNN model for a variety of settings, see Table 4.13. Each experiment was run on one P100 NVIDIA GPU. From Table 4.13, we see that the time increased caused by increasing the batch size or increasing the number of samples is very similar. Component-wise, removing strong augmentation gives the largest decrease in computational time whilst removing the graphical propagation saved just over an hour on CIFAR-100. This represent a very small time trade off given the advantages present in using graphical pseudo-labels.

4.6 Conclusions and Further Work

In this Chapter we have presented two approaches CycleCluster [188] and LaplaceNet [189] which both seek to improve performance for deep semi-supervised image classification whilst decreasing complexity and adding theoretical understanding. Both approaches are built around graph-based pseudo-labelling but differ in the additional regularisation they use.

In CycleCluster we present a novel approach built around the direct implementation of the cluster assumption, rather than the dominant perturbation based approach. We show through extensive experimentation that cluster regularisation is a strong alternative to consistency regularisation due to both its performance and the fact that we remove

the need to design domain specific δ-perturbations which can often be costly and time-consuming for many domains.

In LaplaceNet we achieve state-of-the-art results whilst vastly stripping down model complexity. We propose and justify a multi-sample approach to data augmentation which better connects to the mathematics of vicinal risk minimisation. We experimentally show that an augmentation averaging not only increases generalisation but decreases the model's sensitivity to augmentation. Furthermore, we emphatically show the advantage that graph-based pseudo-labels have over their neural network counterparts, which additionally allows us to avoid the use of many technical tricks, such as confidence thresholding or temperature sharpening, typically used to improve pseudo-label accuracy.

Despite the success of this work there are still many challenges facing the field of deep semi-supervised in explaining its real world success. Given how dependent semi-supervised works have become on data augmentation, further theoretical work is needed to fully understand its benefit. The set of transformations themselves are often hand-crafted by experts in the fields. Furthermore, choosing between transformation sets is often only possible by comparing the accuracy of the model after training which takes a huge computational budget.

Additionally, whilst this work has shown that graph based pseudo-labels remove the need for confidence based weighting, other interesting pathways have opened up for uncertainty quantification of pseudo-labels. One such example being the work of Rizve et al [178] based around estimating MC-Dropout. Incorporating uncertainty quantification to filter out uncertain predictions could be key in improving performance in unbalanced datasets, as well as semi-supervised learning in general.

Chapter 5

Semi-Supervised Medical Image Classification with the Graph 1-Laplacian

In this chapter, we present a semi-supervised framework built around the $p = 1$ graphical Laplacian for the purpose of medical image classification. This research was done jointly with Angelica I. Aviles-Rivero, Nicolas Papadakis, Ruoteng Li, Qingnan Fan, Robby T Tan and Carola-Bibiane Schönlieb and resulted in the publications of both a transductive multi-purpose framework [12] and an inductive COVID detection model [15]. My role in this collaboration was in the collective effort on the conceptualisation of the semi-supervised algorithm and in the experimental testing of the models.

5.1 Introduction

The drastic developments in machine learning, as shown and discussed throughout this thesis, have led to the development of large and powerful deep learning models. These models have not only demonstrated state-of-the-art performance on curated benchmark datasets but in many real world applications such as autonomous driving. However, it is important that deep learning research is not only used to produce new products but additionally, and wherever possible, to positively impact the life of the average person.

The medical field generates mammoths amounts of data, more than it can handle. In the UK alone, the data generated yearly by the National Health Service was estimated by Ernst and Youngs to be worth approximately £9.6 billion pounds [230]. Furthermore, medical data can often only be interpreted by domain experts who are very time limited.

These two factors combine to make the medical field an appetising opportunity for the deployment of machine learning systems. This fact was quickly recognised by the medical imaging community and machine learning models were already being published in the late 70's such as the work by Fukushima [82].

With recent improvements in the training and optimisation of deep neural networks there has been an explosion in the number of applications of deep learning methods to the medical domain, as shown in the detailed survey by Litjens et al [136]. The use of deep learning methods have been shown in tasks including drug discovery [114], seizure prediction [128] and image segmentation [181]. In this chapter we focus on the medical imaging domain where several deep learning approaches have been designed to tackle the problem of image and object classification [119, 31] as well as organ and lesion detection [182, 234] and organ and substructure segmentation [238, 45]. We refer readers to the survey paper by Litjens et al [136] for a thorough overview of the field.

An issue which particularly effects the medical domain is the scarcity of labelled data and annotated images. On top of the standard time and financial costs of acquiring labelled data, medical data is a complex technical domain which requires years of studying to be understood. When curating a dataset the available annotators are much fewer in number, under strong time constraints and can be expensive to hire. With this in mind, the development of deep semi-supervised methods for medical image classification is of fundamental importance for lowering the barriers for deploying deep learning systems in the medical domain.

The $p = 1$ and $p \to 1$ graph Laplacian have recently been shown to produce superior performance than the commonly used $p = 2$ graph Laplacian in the works of [14] and [34] respectively. However, despite the efficient solving system developed in [34], many practitioners have continued to use the $p = 2$ graph Laplacian due to its computational speed. Recent advanced optimisation tools have increased the speed of implementing energies based on the $p = 1$ graph Laplacian to the point where they can now be used in large-scale models. Therefore, due to the paramount importance of accuracy in the medical domain, we use the $p = 1$ graph Laplacian for label diffusion.

In this chapter we propose both a transductive and an inductive framework for semi-supervised medical image classification utilising $p = 1$ graph Laplacian based pseudo-labelling. We provided an efficient solution for large-scale problems whilst keeping relevant mathematical guarantees such as convergence to a steady state. The transductive framework is purely graphical whilst the inductive framework is combined with a neural-network model which is cyclically trained using the output of the transductive framework. Our frameworks are very general, but in this chapter we focus on the diagnosis of chest

X-rays and we demonstrate through rigorous experimental that our semi-supervised methods outperform state-of-the-art supervised approaches whilst training on a fraction of the label data.

Contributions

In this Chapter we propose both a purely graphical transductive framework and a hybrid graphical neural network inductive framework for the semi-supervised classification of medical images. The main contributions from these frameworks are as follows.

1. For the first time in the medical domain, we implement a $p = 1$ graph Laplacian approach for label diffusion due to its superior performance over the standard $p = 2$ graph Laplacian [14]. We do this by implementing an accelerated primal-dual algorithm originally proposed by Chambolle et al [41]. We demonstrate that this approach is effective for both transductive node classification and inductive pseudo-labelling in the medical domain.

2. We apply our transductive model to the classification of chest X-rays in an implementation that we name GraphX$^{\text{Net}}$ [12]. Through detailed experimental results we demonstrate that our approach produces highly competitive results on the "ChestX-ray14" dataset. when given a fraction of the available labelled patient data.

3. For improved generalisation and inference speed, we combine our $p = 1$ graph Laplacian approach with a neural network feature extractor to create an inductive framework for medical image classification. We apply this inductive framework, which we name GraphXCovid [15], to the classification of COVID-19 using chest X-rays , which to the best of our knowledge, is the first SSL method implemented for COVID-19 detection. We demonstrate that our technique achieves better accuracy and higher higher sensitivity in COVID-19 diagnosis then comparable deep supervised techniques, whilst requiring far less labelled data.

The rest of the chapter is structured in the following way. In Section 5.2 we give the motivation and mathematical background to the $p = 1$ graphical Laplacian. In Section 5.3 we present the related work of deep learning approaches in the domain of medical imaging. In Section 5.4 we present the methodology and experimental results relating to GraphX$^{\text{Net}}$. In Section 5.5 we expand our framework into the inductive setting and present the methodology and experimental results relating to GraphXCovid. Finally, in Section 5.6 we conclude the chapter and discuss directions for further work.

5.2 Preliminaries

In this chapter we use a graphical approach to both transductive learning and pseudo-label generation. Our reasoning for using a graphical approach is two fold:

i An increased explaniblity. From a clinical perspective it is important to be able to infer why a certain image has been assigned a given class instead of accepting the black box nature of machine learning frameworks. A weighted graphical structure presents a nice intuitive representation in which images which are connected to their similar neighbours from which one can infer why labels were assigned without resorting to detailed parameter analysis.

ii Higher accuracy: From the work of Raghu et al [170] we know that medical domain is more complex than the natural imaging domain and that simply transferring knowledge or methods across domains leads to poor solutions. Given that naive neural network based predictions already struggle with simple toy model problems [8] and improvements based on network confirmation [91] have serious concerns, model based pseudo-labels make for a poor candidate method. Instead graphical works [189] and in particular the $p = 1$ Laplacian [14] have shown much better performance at a higher computer cost.

Building on the discussion of graph theory in chapter 2 we now define the $p = 1$ graph Laplacian and related mathematical concepts. From a joint distribution $\mathcal{Z} = (\mathcal{X}, \mathcal{Y})$ we have a dataset Z of size $n = n_l + n_u$ comprised of a labelled part of joint samples $Z_l = \{x_i, y_i\}_{i=1}^{n_l}$ and a unlabelled part $Z_u = \{x_i\}_{i=n_l+1}^{n}$ of single samples on $\mathcal{X}$. The labels come from a discrete set $y \in \{1, 2, .., C\}$ of size C. We represent the available data using an undirected weighted graph by $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{W})$ composed of n nodes $\mathcal{V} = \{v_1, ..., v_n\}$, which are connected by m edges $\mathcal{E} = \{e_{ij}\}$ and each edge $e_{ij} \in \mathcal{E}$ has an associated weight w_{ij} which is stored in the matrix $\mathcal{W} \in \mathbb{R}^{m \times n}$. For addressing this problem, we consider functions $u \in \mathbb{R}^n$ defined over the set $\mathcal{N}$ of n nodes $u : \mathcal{V} \to \mathbb{R}$.

In this Chapter we are focused on convex and absolutely p-homogeneous (i.e. $J(\alpha u) = |\alpha|^p J(u)$) non-local functionals of the form:

$$\mathcal{L}_p(u) = \sum_{e_{ij} \in \mathcal{E}} w_{ij} \left(\frac{u_i}{d_i^{1/p}} - \frac{u_j}{d_j^{1/p}} \right)^p, \tag{5.1}$$

where the summation is over the edges of the graph $\mathcal{G}$ and d_i is the degree of node v_i where $d_i = \sum_j w_{ij}$. This non-local functional can be seen to be the quadratic form of the degree normalised graph p-Laplacian. Instead of the commonly used $\mathcal{L}_2(u)$, we instead

focus on the the non-smooth, absolute one homogeneous energy defined by $\mathscr{L}_1(u)$. Note that $\mathscr{L}_1(u)$ can be rewritten as

$$\mathscr{L}_1(u) = ||WD^{-1}u||, \tag{5.2}$$

with an $n \times n$ diagonal matrix D with $D_{ii} = d_i$ containing the node degrees and an $m \times n$ matrix W that encodes the m edges in the graph. Each of these edges is represented on a different line of the sparse matrix W with the value w_{ij} (resp. $-w_{ij}$) on the column i (resp. j). To simplify notation we define the functional $J(u) = \mathscr{L}_1(u)$.

Subdifferential Let us first define ∂J as the set of possible subdifferentials of J so that $\partial J = \{\phi \text{ s.t. } \exists u \text{ with } \phi \in \partial J(u)\}$. Any absolutely one homogeneous functional J satisfies

$$J(u) = \sup_{\phi \in \partial J} \langle \phi, u \rangle, \tag{5.3}$$

so that $J(u) = \langle \phi, u \rangle$, $\forall \phi \in \partial J(u)$. For the particular function J defined in (5.1), we can observe that

$$\phi \in \partial J \Leftrightarrow \phi = D^{-1}W^\top z, \text{ with } ||z||_\infty \leq 1. \tag{5.4}$$

Considering the finite dimension setting, there exists $L_J < \infty$ such that $\|\phi\|_2 < L_J$, $\forall \phi \in \partial J$. We also have the following property.

Proposition 5.1. *For all $\phi \in \partial J$, with J defined in* (5.1)*, one has $\langle \phi, d \rangle = 0$ where $d = \{d_i\}_{i=1}^n$.*

Proof. Observing that $d = D\mathbb{1}_n$ and using (5.4) we have that there exists $z \in \mathbb{R}^m$ such that

$$\langle \phi, d \rangle = \langle D^{-1}W^\top z, D\mathbb{1}_n \rangle.$$

For all $z \in \mathbb{R}^m$, and by the construction of W, we see that that $W^\top z = \sum_{\{v_i, v_j\} \in \mathscr{E}} w_{ij} z_i \mathbf{e}_i - w_{ij} z_i \mathbf{e}_j$. Therefore

$$\langle W^\top z, \mathbb{1}_n \rangle = \sum_{\{v_i, v_j\} \in \mathscr{E}} w_{ij} z_i - w_{ij} z_i = 0. \tag{5.5}$$

□

Eigenfunction. Eigenfunctions of any functional J satisfy $\lambda u \in \partial J(u)$. For J being the nonlocal total variation, (i.e. when d_i is constant), eigenfunctions are known to be essential tools to provide a relevant clustering of the graph [218]. Subsequently, many methods [35, 28, 30] have been designed to estimate these eigenfunctions through the local minimisation of the Rayleigh quotient, which reads:

$$\min_{\|u\|_2=1} \frac{J(u)}{T(u)}, \tag{5.6}$$

with T being another absolutely one homogeneous function , which is typically a norm. Taking $T(u) = \|u\|_2$ one can recover eigenfunctions of J. For $T(u) = \|u\|_1$, one can also compute bi-valued functions u that are local minima of (5.6) and eigenfunctions of J [75]. Being bivalued, these estimations can easily be used to partition the the domain. These schemes also relate to the Cheeger cut of the graph induced by nodes u_i and edges w_{ij}. Balanced cuts can also be obtained by considering $T(u) = \|u - \text{median}(u)\|_1$ [29]. A last point to underline comes from Proposition 5.1 that states that eigenfunctions $\lambda u \in \partial J(u)$ should be orthogonal to d. It is thus important to design schemes that ensure this property.

5.3 Related Work

Whilst this related work section focuses on the application of semi-supervised learning for medical image classification, many prior works have have been done under the supervised learning paradigm. For brevity, we refer readers to survey works such as [144, 136] for a detailed examination of the supervised setting. For semi-supervised classification in the medical domain there have been several different families of proposed solutions and we refer the reader to the survey paper of Cheplygina et al [50] for a detailed overview of semi-supervised learning in the medical domain. In this section we review the methods of self-training and graphical methods in turn as our transductive and inductive classification frameworks are closely related to these approaches.

Self-Training via Pseudo-Labelling As previously discussed in Chapter 4, pseudo-labelling techniques work by estimating labels for unlabelled examples and then incorporating these newly pseudo-labelled pairs into a shared learning framework. A variation of this style of learning is "self-training" whereby a supervised learning algorithm is initially trained based on the labelled data only. This classifier is then applied to the unlabelled data to generate more labelled examples as input for the supervised learning algorithm. The major difference between the variations being that self-training uses a strictly supervised learning algorithm whilst pseudo-labelling can include other unsupervised or semi-supervised learning terms.

In medical imaging several different algorithms based around self-training have been proposed which differ in the criterion by which unlabelled images are added to the labelled data samples. Some approaches have required that domain experts validate guessed labels before unlabelled images are incorporated [200, 166] The obvious downside of

these approaches being that constantly seeking annotator feedback is an incredibly costly time overhead. Other approaches, in a similar manner to modern deep semi-supervised learning for natural imaging, seek to quantify how uncertain the supervised algorithm is about a label and use this uncertainty as a selection criterion. In the work of Wang et al [219] the authors propose selecting unlabelled examples whose uncertainty lies below some user inputted confidence threshold, a method that was later used in the natural image domain in works such as FixMatch [197]. In the work of van Rikxoort et al [214] the authors propose obtaining expert validation of uncertain images where, given an ensemble of classifiers, a produced label was uncertain in two cases: one classifier having low confidence whilst the others had high confidence and where different classifiers guessed different labels for the same image.

Graphical Methods In the medical domain, graphical semi-supervised methods are overwhelmingly used for the tasks of image segmentation and feature extraction. In image segmentation, graph cut approaches are used to propagate labels between similar pixels or superpixels such as in the works of [200, 146, 53]. In feature extraction approaches, the graph $p = 2$ Laplacian has been used to regularise the output of feature embedding methods by requiring that points close in the graph should be given similar representations in some lower-dimensional feature space [207, 225].

Whilst the use of semi-supervised graphical methods has been widely explored for image segmentation and feature extraction methods the same cannot be said for image classification. To the best of our knowledge there was only one other graphical semi-supervised medical image classification work, by Su et al [199], that was written in parallel with our own approach. Su et al's method sought to further regularise a Mean Teacher [205] model by using the label propagation approach of Zhou et al [249] combined with a Siamese loss which sought to pull images of the same class together and different classes apart in the feature space of the model.

Our Approach: In both our transductive and inductive framework we have marked differences compared to the current body of literature. Our approach is the first to use the $p = 1$ graph Laplacian for classification in the medical image domain and first pseudo-labelling approach to explore the use of hybrid graph neural-networks models for semi-supervised medical image classification. It is our belief that these methods will be able to perform comparably with the current state-of-the-art supervised approaches whilst using a fraction of the labelled data.

5.4 GraphX$^{\text{Net}}$

In this section we overview our transductive classification framework which we used for diagnosing Chest X-rays (CXRs), named GraphX$^{\text{Net}}$ [12]. However, we note that this framework is very general. GraphX$^{\text{Net}}$ constructs a weighted graphical representation of a set of images before using the $p = 1$ graph Laplacian to propagate information across the graph. We rigorously tested our approach on the ChestX-ray14 [229] dataset. Our approach learns to accurately classify CXRs with a performance comparable to state-of-the-art deep learning techniques, whilst using a far smaller amount of labelled data in the training phase. This work also represents the first time that graph representations have been used for X-ray classification.

The rest of this section is structured in the following way. Firstly, we cover the problem of diagnosing chest X-rays and related works in the field. Then we cover the methodology of GraphX$^{\text{Net}}$ focusing on the optimisation of our graphical approach. Finally, we present and discuss the experimental results that we use to validate our approach.

5.4.1 Chest X-ray Classification

The Chest X-ray (CXR) is the most commonly performed X-ray examination [80] and captures details of the lungs, heart, bones and blood vessels. CXRs play a critical role in diagnosing and monitoring conditions such as pneumonia [81], heart problems and certain types of cancer. However, it remains one of the most complex imaging studies to interpret [80] with the effectiveness and accuracy of the interpretation heavily relying on the expertise of the radiologists. Despite the expertise of the radiologists there is still a substantial clinical error on the final outcome [32]. Furthermore, the requirement of human expertise increases the financial cost and extends the time required for evaluation. This issue is highlighted in regions of the world where there is an inadequate supply of trained medical professionals, where it may be impossible to effectively use CXRs. Therefore, there is a clear need for automating the evaluation of CXRs to alleviate these issues.

CXR classification has been widely addressed by the community, yet it remains an open problem. Early works on CXR classification were based on handcrafted frameworks [208, 202]. However, these approaches required certain statistical measures to be present in an image, such as texture, geometry or intensity criterion, for it to be classified as abnormal. Such a constrained approach generalised poorly to unseen data. Due to the incredible results produced by deep learning in the field of computer vision, there has been a rush to apply deep learning architectures to the classification of CXRs [16, 229, 17],

which have shown promising results. The vast majority of these methods utilise deep convolutional neural networks with architectures such as ResNet [96] and DenseNet [104]. Several different training methods have been considered including: pre-trained networks, fine-tuned networks and networks trained from scratch on X-ray data [16].

Despite the improved performance of deep learning approaches compared to classical works, there are drawbacks from relying upon this supervised paradigm. Firstly, the limited size of medical imaging datasets often makes it infeasible to purely train deep learning architectures from medical imaging data. This limited dataset size is often due to delays in acquiring domain expert annotation. Secondly, due to the complexity of the task at hand, medical datasets often contain noisy and incorrect labels and supervised methods often overfit to this noise. Therefore, we propose a semi-supervised graphical framework for CXR classification which alleviates the need for large quantities of labelled data whilst improving generalisation due to the inclusion of label independent regularisation.

5.4.2 Methodology

GraphX$^{\text{Net}}$ tackles the problem of transductive classification for chest X-rays. Rather than train a neural network from scratch, we instead use the $p = 1$ graph Laplacian to propagate label information across a graphical representation of the dataset. In this case the graph is constructed from a feature representation generated from a model pre-trained on the ImageNet [64] dataset. Therefore, our approach is composed of two steps *graph construction* and *propagation* which we explain in detail. Firstly, we state the transductive problem statement as it differs from the prior inductive problem statement.

Transductive Problem Statement. From a joint distribution $\mathscr{X} \times \mathscr{Y}$ we have a dataset Z of size $n = n_l + n_u$ comprised of a labelled part of joint samples $Z_l = \{x_i, y_i\}_{i=1}^{n_l}$ and a unlabelled part $Z_u = \{x_i\}_{i=n_l+1}^{n}$ of single samples on $\mathscr{X}$. The labels come from a discrete set $y \in \{1, 2, .., C\}$ of size C . Our task is to accurately predict the labels for the set Z_u compared to the hidden ground truth.

Graphical Construction

From some pre-trained network, $\hat{t}_\theta(\cdot) : \mathscr{X} \to \mathbb{R}^{d_p}$, we extract a feature representation of the data input which we denote as $T = t(X_l \cup X_u) = \{\hat{t}(x_1), .., \hat{t}(x_n)\}$. From this feature representation we construct a weighted graphical representation $\mathscr{G} = (\mathscr{V}, \mathscr{E}, \mathscr{W})$ where each node represents a chest X-ray. Given some metric function $D : \mathbb{R}^{d_p} \times \mathbb{R}^{d_p} \to \mathbb{R}$, in our case we again use the normalised inner product, we first construct a fully connected edge set $\mathscr{E}$ where the weight of edge e_{ij} is given by $w_{ij} = D(\hat{t}(x_i), \hat{t}(x_j))$. We then sparsify the

edge set using symmetric K-nearest neighbours, so that $w_{ij} = D(\hat{t}(x_i), \hat{t}(x_j))$ if v_i is one of the K closest neighbours to v_j or vice versa with $w_{ij} = 0$ otherwise.

Label Propagation with the Graph 1-Laplacian

From the graphical representation, we are then free to propagate the initial label information across the graph using the $p = 1$ graph Laplacian. The classification result is then obtained by selecting the most probable class for each image. The general p graph Laplacian is given by

$$\mathscr{L}_p(u) = \sum_{i,j} w_{ij} \left\| \frac{u_i}{d_i^{1/p}} - \frac{u_j}{d_j^{1/p}} \right\|_p, \text{ with } p \geq 1 \text{ and } d_i = \sum_j w_{ij} > 0. \tag{5.7}$$

In the case $p = 1$ we have the concise formulation that $\mathscr{L}_1(u) = J(u) = |WD^{-1}u|$. Unlike the $p = 2$ which can be solved very quickly using a conjugate gradient approach, the $p = 1$ case requires a more complex and costly optimisation approach which we now detail.

For each class, $k = 1 \cdots C$, we consider a variable u^k that has values on all nodes of the graph. For all unlabelled nodes $i > n_l$, the class variables are then coupled with the constraints that for all nodes $i \geq n_l$:

$$\sum_{k=1}^{C} u_i^k = 1, \forall i > n_l. \tag{5.8}$$

This simple coupling leads to faster projection algorithms than simplex [29, 83] or non convex orthogonality constraints between [66]. For each class $k \in \{1, .., C\}$, we assume that there is a set of annotated nodes $\mathscr{I}_k \subset \{1 \cdots n_l\}$ such that $y_i = k \ \ \forall i \in \mathscr{I}_k$. Taking a small parameter $\epsilon > 0$, we therefore constrain that:

$$\begin{cases} u_i^k \geq \epsilon & \text{if } i \in \mathscr{I}_k \\ u_i^{k'} \leq -\epsilon & \text{if } i \in \mathscr{I}_k \text{ and } k' \neq k. \end{cases} \tag{5.9}$$

This information is then used in an iterative PDE process with a time parameter t, in which we seek to minimise the sum of normalised ratios $\sum_k \frac{J(u^k)}{|u^k|}$. Denoting $\mathbf{u} = [u^1, \cdots u^C]$ and a time step $\Delta t > 0$. Then formally, we seek to minimise:

$$\mathbf{u}^{(t+1)} = \arg\min_{\mathbf{u}} \frac{||\mathbf{u} - \mathbf{u}^{(t)}||^2}{2\Delta t} + \sum_{k=1}^{C} \left(J(u^{k,0}) - \frac{J(u^{k,(t)})}{|u^{k,(t)}|} \langle \text{sign}(u^{k,(t)}), u^{k,0} \rangle \right), \tag{5.10}$$

where $u^{k,0}$ is our initial solution for each class which contains the initial label information. Following [99, 76], a final shifting $u^{k,(t+1)} = u^{k,(t+1)} - \text{median}(u^{k,(t+1)})$ and a normalisation $\mathbf{u}^{(t+1)} = \mathbf{u}^{(t+1)}/||\mathbf{u}^{(t+1)}||$ are necessary at the end of each iteration to prevent from converging to trivial solutions.

When an unique u^k is considered, the scheme iteratively decreases the ratio $\frac{J(u^{k,(t)})}{|u^{k,(t)}|}$ since $\langle \text{sign}(u^{k,(t)}), u^{k,(t)} \rangle = |u^{k,(t)}|$, so that the solution $u^{k,(t+1)}$ of (5.10) necessarily satisfies:

$$J(u^{k,(t+1)}) \leq \frac{J(u^{k,(t)})}{|u^{k,(t)}|} \langle \text{sign}(u^{k,(t)}), u^{k,(t+1)} \rangle \leq \frac{J(u^{k,(t)})}{|u^{k,(t)}|} |u^{k,(t+1)}|. \tag{5.11}$$

As noticed in [76], the scheme makes $u^{k,(t)}$ converge to a bivalued function that naturally segments the graph. As C variables are coupled, the final labelling of a node i is chosen by selecting the $k \in \{1,..,C\}$ such that $u_i^k \geq u_i^{k'} \forall k' \neq k$. Therefore, $y_i = \arg\max_k u_i^k$.

Optimisation Scheme. For each time step t, the problem (5.10) is solved at successive time steps using the accelerated primal dual algorithm of [41]. Denoting as $\mathbf{v} = \mathbf{u}^{(t)}$ the current estimation and initialising $\mathbf{u}_0 = \tilde{\mathbf{u}}_0 = \mathbf{v}$, $z_0^k = WD{-}1u_0^k$ with positive parameters σ_0 and τ_0 satisfying $\sigma\tau||J||^2 < 1$, where $||J||$ is the induced norm of the operator, the algorithm to obtain $\mathbf{u}^{(t+1)}$ with an iterative sequence $\mathbf{u}_\ell$ indexed by ℓ reads:

$$\begin{cases} z_{\ell+1}^k & = z_\ell^k + \sigma_\ell W D^{-1} \tilde{u}_\ell^k \\ z_{\ell+1}^k & = \frac{z_{\ell+1}^k}{\max(1, |z_{\ell+1}^k|)} \\ u_{\ell+1}^k & = \frac{u_\ell^k + \tau_\ell \Delta t \left(\frac{\Delta_1(v^k)}{|v^k|} \text{sign}(v^k) + D^{-1} W z_{\ell+1}^k \right)}{1 + \tau_\ell \Delta t} \\ u_{\ell+1}^k & = \text{Proj}_C(u_{\ell+1}^k) \\ \gamma_\ell & = 1/\sqrt{1 + \tau_\ell / \Delta t}, \tau_{\ell+1} = \tau_\ell \gamma_\ell, \sigma_{\ell+1} = \sigma_\ell / \gamma_\ell \\ \tilde{u}_{\ell+1}^k & = u^k + \gamma_\ell (u^{k+1} - u^k), \end{cases}$$

where the projection onto the set of constraints C combining (5.8) and (5.9) reads pointwise:

$$\text{Proj}_C(u_i^k) = \begin{cases} \max(u_i^k, \epsilon) & \text{if } i \in \mathscr{I}_k \\ \min(u_i^k, -\epsilon) & \text{if } i \in \mathscr{I}_{k'} \text{ and } k' \neq k. \\ u_i^k - \frac{1}{L} \sum_{k'} u_i^{k'} & \text{if } i > l. \end{cases} \tag{5.12}$$

We use this process to iterate $\mathbf{u}^{(t+1)}$ over the time steps and then extract the final full labelling of the data by using $y_i = \arg\min_k u_i^k \ \forall \ i > n_l$

5.4.3 Results and Discussion

In this section, we detail the experiments that we conducted to validate the performance of GraphX$^{\text{Net}}$. In turn we describe the chosen dataset, the evaluation protocol, parameter selection and finally present numerical and visual results.

Dataset Description.

We evaluate our approach using the ChestX-ray14 [229] dataset, which is composed of 112,120 frontal chest view X-rays each having size of 1024×1024 pixels. The dataset is composed of 14 classes (pathologies).

Evaluation Methodology

Our evaluation protocol to validate our model is as follows. Firstly, we visually demonstrate the graphical representation of the ChestX-ray14 dataset alongside some example classifications. Secondly, and the main part of the evaluation, we compare our GraphX$^{\text{Net}}$ to the state-of-the-art approaches for chest X-ray classification. We compare our approach against two deep learning techniques: WANG17 [229] and YAO18 [240]. To evaluate the classifier quality of each method we performed a ROC analysis using the area under the curve (AUC) per pathology along with their average. Finally, beside the official split, we perform a comparison with random partitions on ChestX-ray8 using WANG17 [229] as baseline.

Implementation

For the graphical construction we use $K = 50$ nearest neighbours for each point. For our optimisation framework we use the following parameters: σ_0, τ_0 are randomly initialised such that $\sigma_0 \tau_0 ||J||^2 \leq 1$, the time step for the pde was set to $\delta t = 0.05$ and the primal-dual algorithm was ran for 100 steps for each time step. Furthermore, we set the value of epsilon to be $\epsilon = 1e^{-3}$, which was also the value used to threshold the convergence over the residuals.

Results and Discussion

Firstly, we visualise our proposed approach by demonstrating a graphical representation of the dataset alongside some example classifications which we provide in Fig. 5.1. The left side of the figure shows two graphs in which the first one illustrates the initial state of the graph created after computing the feature distances between X-rays in the dataset whilst

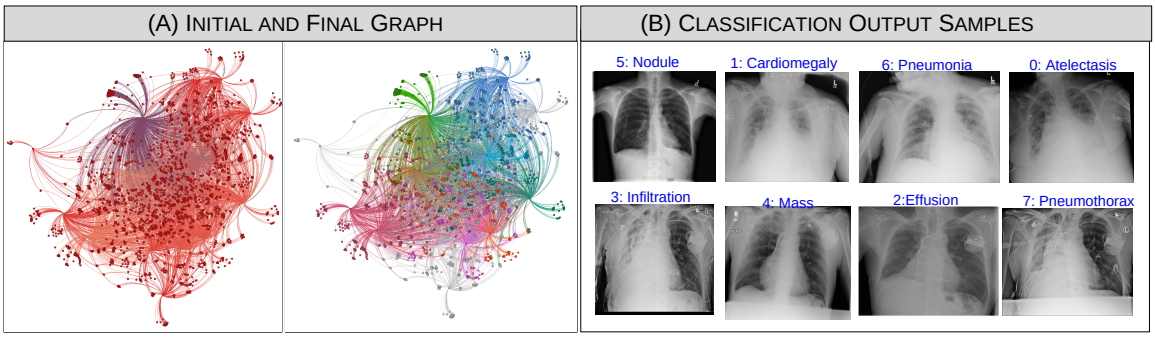


Fig. 5.1 Graphical Construction and Classification. Sub-figure (A) shows the graphical representation of the ChestX-ray14 dataset before and after the labelled data is propagated via the $p = 1$ graph Laplacian operator. Sub-figure (B) demonstrates examples of correct classifications produced by our framework.

Pathology	Wang17 [229]	Yao18[RF]	GraphX^{NET}
Cardiomegaly	0.81	0.856	0.8799
Emphysema	0.833	0.843	0.8407
Edema	0.805	0.806	0.802
Hernia	0.872	0.775	0.8722
Pneumothorax	0.799	0.805	0.837
Effusion	0.759	0.806	0.792
Mass	0.693	0.777	0.809
Fibrosis	0.786	0.743	0.8034
Atelectasis	0.7	0.733	0.7189
Consolidation	0.703	0.711	0.7336
Pleural Thicken	0.684	0.724	0.757
Thicken	0.669	0.724	0.7205
Nodule	0.658	0.684	0.7113
Pneumonia	0.661	0.673	0.7664
Average AUC	0.7451	0.7614	0.7888

Table 5.1 Comparison of the classification accuracy of GraphX^{NET} against state-of-the-art deep learning methods of Wang et al. [229] and Yao et al. [240]. Here we give the average AUC measure over all classes. Note that the supervised methods are trained with all labelled training data whilst GraphX^{NET} uses approximately 28% of the labelled data.

the second one shows the graph after computing (5.10). Each distinct colour on the graph represents a different diagnosis. The right hand side shows some example classifications generated by our approach that were correctly classified.

We then quantitatively evaluated the performance of our approach and compared it against the leading supervised deep-learning approaches of WANG17 [229] and YAO18 [240].

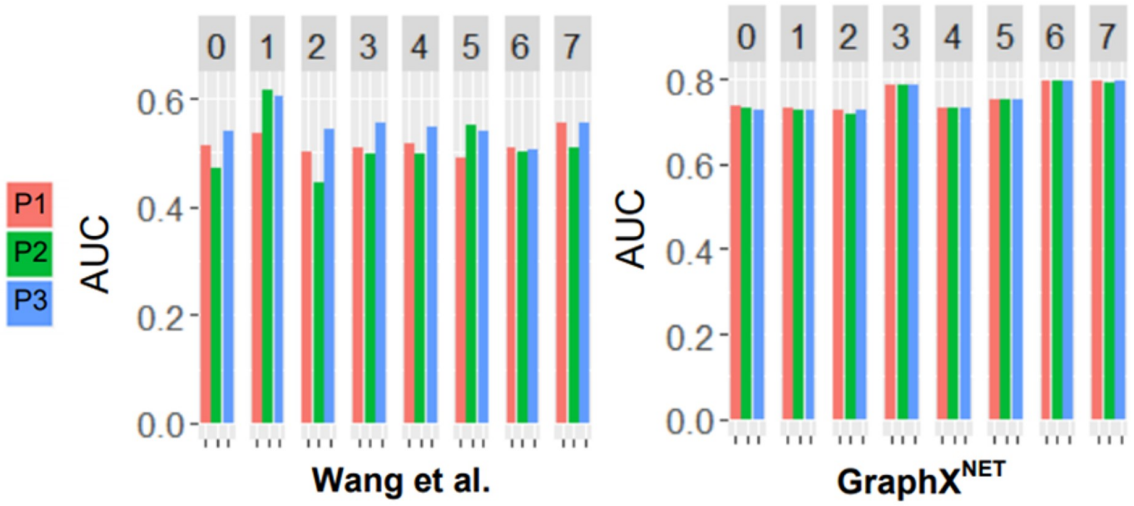
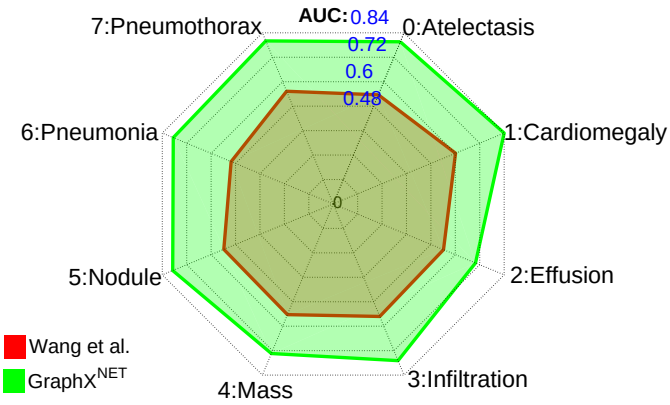


Fig. 5.2 Comparison of the performance of our GraphX$^{\text{NET}}$ and the approach of Wang et al. [229] on three randomly generated partitions of the ChestX-ray14 dataset over the seven largest classes. We clearly see that the supervised method of Wang is more sensitive to the paritioning compared to our semi-supervised approach which uses the global structure of the data as a regularisation.

To the best of our knowledge, at the time there was no other semi-supervised learning method for X-ray classification which could be used as a compared approach. To rigorously test the semi-supervised nature of our model we train the models of Yao and Wang using all available training data, which amounts to 70% of the total data, whilst only training our model using 28% of the available training data, which amounts to 20% of the total data. Table 5.1 shows the AUC results of all approaches where overall our approach outperformed the other methods across most pathologies. Even though YAO18 performs better in some classes, our approach uses a fraction of the labelled data to produce a better overall classification of the data.

Furthermore, due to the regularisation provided by the graph being label independent we are much more stable with respect to changes to the partition of the dataset. To verify this we generated three different random data partitions, split along patient lines, and tested the AUC of both GraphX$^{\text{NET}}$ and WANG17 [229]. We again trained Wang's and our model using the same amount of labels as previously described (70% and 20%). We present these results in Fig 5.2. We clearly see that our approach is far less sensitive to the partitioning of the dataset compared to the supervised approach of Wang et al.



WANG17 [229]					
%Labeled	70%				
AUC	0.548				
GraphX^{NET}					
%Labeled	2%	5%	10%	15%	20%
AUC	0.53	0.58	0.63	0.68	0.78

Table 5.2 Comparison of the classification accuracy of GraphX^{NET} against a state-of-the-art deep learning method by Wang et al. [229]. Here we give the average AUC measure over all eight classes using different amounts of labelled data. Additionally, we give a class by class comparison between the two methods using 70% and 20% of the labelled data for the Wang method and GraphX^{NET} respectively.

Finally as a ablation study we tested how the average AUC accuracy of our model changed using different percentages of the total data as labelled input. We present the result of this ablation study in Table 5.2. Our approach is able to produce a reasonable accuracy when only 2% of the data is labelled. Further supporting the view that semi-supervised learning will have a significant role to play in the medical imaging domain.

5.5 GraphXCovid

Despite the success of our transductive framework GraphX^{NET}, there are several weaknesses. Firstly, we relied on a pre-trained model to give us our initial feature representation. As demonstrated by the work of Raghu et al [170], this can be a significant bottleneck in performance. Secondly, and true for all transductive methods, if we want to classify new images then the transductive method has to be rerun again from scratch. This is a significant computational and time burden and unsuitable for the fast application of automated medical imaging system in practice.

Therefore, in this section we expand our transductive framework into an inductive framework based around the $p = 1$ graph Laplacian which learns its own feature representation and can infer labels on unseen data quickly. We focus the application of this framework on the semi-supervised detection of COVID-19 from X-ray images, which we term GraphXCovid. The aim of GraphXCovid was to show that the deep semi-supervised

learning paradigm could be applied to rapidly developing biological events. In such events, time constraints may not allow for the collection of large scale labelled datasets for supervised machine learning techniques to be trained and be deployed. However, given that semi-supervised methods alleviate the need for large quantities of labelled data, they are an ideal candidate for the rapid deployment of machine learning system to aid in the response.

This rest of this section is structured in the following way. Firstly, we discuss the problem and related work surrounding machine learnt COVID-19 diagnosis via chest X-rays. Then we detail the methodological changes present in the inductive framework with include combining a graphical $\mathscr{L}_p$ Laplacian approach with a neural network backbone as well as incorporating techniques to control class imbalances and confirmation bias. Finally, we demonstrate through a rigorous quantitative analysis that GraphXCovid outperforms the current state-of-the-art COVID-19 chest X-ray detection algorithms whilst using a fraction of the available labels to train on.

5.5.1 COVID-19 Detection via Chest X-rays

Since the outbreak of the novel coronavirus disease 2019 (COVID-19), which is caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2), there have been more than 210 million confirmed infected cases and more than 4 million deaths reported worldwide (as of September 2021). This threat has encouraged joint efforts to obtain accurate early detection of COVID-19 to try and limit the spread of the pandemic.

Whilst the polymerase chain reaction (PCR) COVID-19 test is the current gold standard for diagnosis, this type of test has demonstrated some limitations and burdens. In practical settings it is prone to false negatives [237, 231] and additionally several world regions struggle in obtaining fast accessibility to the test. Therefore, it has been suggested to use imaging techniques, including computerised tomography (CT) and chest X-rays (CXRs), in parallel with PCR testing to diversify the options available.

Computerised tomography (CT) has been a focus of attention in the literature for COVID-19 e.g. [250, 52]. However, the financial burden imposed in acquiring enough CT scan suites for it to be a scalable solution and the inefficiencies relating to room decontamination make CT challenging to be used on a routinely basis despite its high sensitivity [73]. Due to its wider availability and inexpensive cost, much hope has been placed in CXRs for both clinical and AI areas e.g. [108, 235, 54, 226].

As pointed out by the WHO chest imaging guidelines [161], imaging-based diagnosis using CXRs plays an important role for improving decision making in several cases [161]. Firstly, when PCR testing or results are not immediately available. Secondly, when initial

PCR testing is negative but there is high clinical suspicion of COVID-19. Despite the advantages of using CXRs, accurate interpretation remains a challenge [80]. This is because the accuracy of the interpretation relies on the radiologist's expertise level and there is still a substantial clinical error on the outcome [32]. Furthermore, access to trained radiologists is often not possible in developing countries. Therefore, there are grounds to see whether automated evaluations of CXRs can aid the radiologist in making their decision.

For the task of classifying COVID-19 using CXRs data, there has been a fast development of deep learning techniques e.g. [157, 7, 74, 100, 179, 3], in which supervised learning is the go-to paradigm. . The current leading supervised model for COVID, COVID-Net [226], has reported promising results with a sensitivity of 91% for COVID-19. However, the performance of these techniques strongly relies on a large and representative corpus of labelled data. In the medical domain and with a new disease, this might be a strong assumption in the design of a solution. Hence ,there are still plenty of room for improvements, namely on how to use the vast amount of available unlabelled data to prevent labelling errors and improve certainties.

Despite, the prior success of deep semi-supervised learning, and to the best of our knowledge, there has been no deep semi-supervised approach proposed for COVID-19 analysis. The potential performance of semi-supervised learning for medical imaging has nevertheless been shown in our prior transductive framework. Therefore, in this work we propose and implement an inductive semi-supervised model for COVID-19 detection. We readily compete and outperform state-of-the-art supervised techniques in identifying COVID-19 in CXRs, whilst using a small fraction of the available labels.

5.5.2 Methodology

In this section, we detail the methodology of our GraphXCovid approach. GraphXCovid's methodology has a lot in common with our previous work on semi-supervised natural image classification and the transductive framework from this Chapter. We use the label diffusion from the $p = 1$ graphical Laplacian to generate pseudo-labels for unlabelled images which are used to iteratively train a neural network backbone. Therefore, in this methodology section we quickly recap the graphical construction and diffusion steps before focusing on the combined graph neural network framework. In this setting we revert back to the inductive learning paradigm previously defined in Chapter 4.

Graph Construction

Prior to semi-supervised optimisation, the model is initially trained on the small number of available examples for a handful of epochs to allow us to construct the initial graph. In our work we simply use the standard supervised loss with a cross entropy loss function l_{ce}

$$L_S(X_L, Y_L; \theta) := \frac{1}{n_l} \sum_{i=1}^{n_l} l_s(f_\theta(x_i), y_i). \tag{5.13}$$

From some model state θ we construct a weighted graph representation $\mathscr{G} = (\mathscr{V}, \mathscr{E}, \mathscr{W})$. The graph is reconstructed at the end of every epoch of training using the updated parameters θ . The graphical construction for the inductive framework is almost identical to the transductive setting except for the fact that the feature information for each node f_i is given by the model's feature extraction layers $t_\theta(x_i)$ rather than some pre-learnt representation $\hat{t}_\theta(\cdot)$. We use the same metric function, the inner product and the same K-nearest neighbours edge selection.

From the constructed graph $\mathscr{G} = (\mathscr{V}, \mathscr{E}, \mathscr{W})$ we again use the accelerated primal-dual optimisation approach of Chambolle et al [41] to minimise the $p = 1$ graph Laplacian operator from Equation 5.7. Using this we are able to produce pseudo-labels $\hat{Y}$ for the unlabelled data points in Z_u by $\hat{y}_i = \arg\max_k u_i^k \; \forall i \geq n_l$.

Semi-Supervised Optimisation

These pseudo-labels are then used to train the model in a semi-supervised manner which we now detail. Similarly to past pseudo-label approaches in Chapter 4, we start off with a composite loss which has separate terms for both the labelled and unlabelled data.

$$L_W(Z, \hat{Y}; \theta) := \frac{1}{n_l} \sum_{i=1}^{n_l} l_s(f_\theta(x_i), y_i) + \frac{1}{n_u} \sum_{i=n_l+1}^{n} l_s(f_\theta(x_i), \hat{y}_i), \tag{5.14}$$

where l_s is the cross entropy loss. We use this loss to train the neural network f_θ on the labelled and pseudo-labelled data. However, despite the graph diffusion approach generating accurate pseudo-labels [12], it is not a perfect method. Some of the produced pseudo-labels will be incorrect and can potentially lead to confirmation bias, where the model becomes overconfident on its own incorrect predictions and propagates this error over time. This may be a particularly damaging effect when one of the classes, in this case COVID-19, is comparatively low in number as it would only take a few false positives to drastically change the decision boundary. In order to limit the effect of confirmation bias

we utilise two technical adjustments which regularise the class distribution of pseudo-labels and encourage the model to focus on high confident estimates respectively.

Firstly, we tackle the problem of imbalanced datasets which is a prevalent issue in the medical domain, often due to the relative probability of different diseases. The problem of classifying with a highly imbalanced dataset has been widely studied in the literature, e.g. [129, 93]. We apply a common strategy for imbalanced class population [93, 78] and add a weighting factor ζ inversely proportional to the effective number of samples per class, so that for class j

$$\zeta_j := \left(\sum_{i=1}^{n_l} \mathbb{1}_{y_i=j} + \sum_{i=n_l+1}^{n} \mathbb{1}_{\hat{y}_i=j} \right)^{-1}, \tag{5.15}$$

Secondly, we apply a weighting term to encourage the model to focus on its most confident predictions during training. As has been shown in several works e.g. [93, 192, 106, 188], by quantifying the confidence of a prediction one can reduce the effect that incorrect predictions have on the model as a whole. We quantify the confidence of a pseudo-label by using the Shannon entropy $H(\cdot)$ of the graphically produced pseudo-label rather than any network prediction. Therefore, we associate an uncertainty weighing factor, ω_i, to each $u_{\hat{y}_i}$ generated in the diffusion process, $\omega_i = 1 - (H(u_{\hat{y}_i})/\log(C))$. Using the class and entropy weights we then have an updated loss of the form

$$L_W(Z, \hat{Y}; \theta) := \frac{1}{n_l} \sum_{i=1}^{n_l} \zeta_{y_i} l_s(f_\theta(x_i), y_i) + \frac{1}{n_u} \sum_{i=n_l+1}^{n} \zeta_{\hat{y}_i} \omega_i l_s(f_\theta(x_i), \hat{y}_i). \tag{5.16}$$

Therefore the overall process for training the mode is as follows. We first initialise the model by optimising (5.13) for a set of epochs. We then perform graph-based diffusion on the $p = 1$ graph Laplacian, which produces the pseudo-labels. The original labels and the produced pseudo labels are then used to optimise the model parameters through the loss (5.16). This optimisation occurs for one pass through the unlabelled data. This iteration between pseudo-label generation and training continues until training finishes after a prior chosen number of optimisation steps. For clarity, we provide a full algorithm for training GraphXCovid in Algorithm 4.

5.5.3 Results and Discussion

In this section, we describe the experiments that we performed to validate GraphXCovid. This includes descriptions of the datasets used and evaluations protocol before the experimental results themselves are presented and discussed.

Algorithm 4 Training GraphXCovid

Input Dataset Z with labelled samples $Z_l = \{x_i, y_i\}_{i=1}^{n_l}$ with C total classes and unlabelled samples $Z_u = \{x_i\}_{i=n_l+1}^{n}$, Model f_θ of composite functions t_θ, g_θ
Parameters: Number of epochs E, Batch size b, labelled batch size b_l, unlabelled batch size b_u.
for i = 1,2,....,100 **do**
  for $j = 1, .., \lfloor \frac{n_l}{b} \rfloor$ **do** ▷ Initial Supervised Baseline
    Batch $B_L = \{x_i, y_i\}_{i=1}^{b} \subset Z_l$
    $\theta \leftarrow L_s = \frac{1}{b} \sum_{i=1}^{b} l_s(f_\theta(x_i), y_i)$
  end for
end for
for i = 1,..,E **do**
  $T = \{t_\theta(x_1), .., t_\theta(x_n)\}$ ▷ Extract Feature Embeddings
  Perform Graph Diffusion with $p = 1$ graph Laplacian
  $\hat{y}_i = \arg\max_k u_i^k \; \forall i \geq n_l$
  for $n_l \leq i \leq n$ **do**
    Calculate entropy weight ω_i
  end for
  for $1 \leq i \leq C$ **do**
    Calculate class weight ζ_i
  end for
  for $i = 1, .., \lfloor \frac{n-n_l}{b} \rfloor$ **do** ▷ Semi-Supervised Learning
    Batch $B_L = \{x_i, y_i\}_{i=1}^{b_l} \subset \{Z_l\}$, $B_U = \{x_i, \hat{y}_i\}_{i=1}^{b_u} \subset \{Z_u, \hat{Y}\}$
    $\theta \rightarrow \frac{1}{b_l} \sum_{i=1}^{b_l} \zeta_{y_i} l_s(f_\theta(x_i), y_i) + \frac{1}{b_u} \sum_{i=1}^{b_u} \zeta_{\hat{y}_i} \omega_i l_s(f_\theta(x_i), \hat{y}_i)$
  end for
end for

Dataset Description

Our approach was evaluated on the COVIDx Dataset which is is a multi-centre dataset first introduced by Wang and Wong [226]. The dataset is composed of a total of 15,254 CXR images. To the best of our knowledge, this dataset is the largest and most diverse for COVID-19 as the samples have been collected from different locations and acquired under a variety of different conditions and vendors. COVIDx itself merges five highly diverse datasets repositories: COVID-19 image data collection [54], Actualmed COVID-19 Chest X-ray Dataset Initiative [223], COVID-19 Chest X-ray Dataset Initiative [224], RSNA Pneumonia Challenge dataset [183, 229] and COVID-19 Radiography Database [51].

The COVIDx dataset contains CXRs with three diagnoses: healthy, pneumonia and COVID-19. The class breakdown, for the full and official partition CXR images, is illustrated in Fig. 5.3. As can be observed from these plots, the distribution of classes in the dataset is

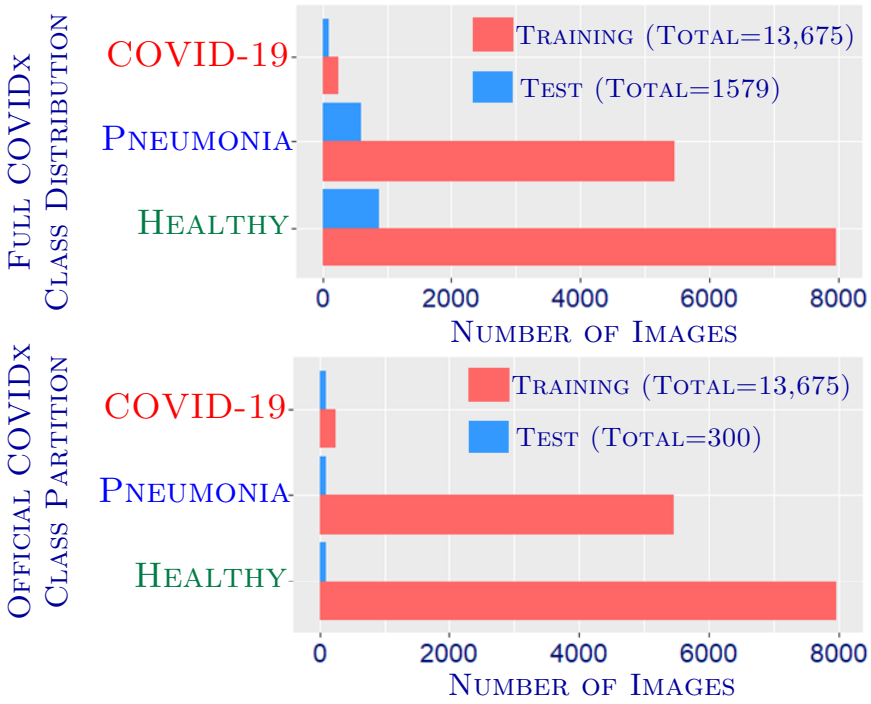


Fig. 5.3 Class distribution of COVIDx dataset. From top to bottom: samples/class distribution over the full COVIDx dataset and the official partition from COVIDx. We conducted experiments using both the full dataset and official partitions. This dataset imposes several challenges as it is highly imbalanced, in that there are significantly fewer COVID-19 samples than the Pneumonia and Healthy classes.

highly imbalances with the number of positive COVID-19 diagnoses being much smaller than the other two classes. The official dataset partition only considers 13,975 CXRs across 13,870 patients, with a training set composed of 13,675 CXRs and a test set of 300 CXRs. However, excluding this partition there exists 1579 CXR test images across the full dataset.

To further support the performance of our technique, we also use an external dataset to evaluate the generalisation of our model to out-of-distribution samples. To do this, we use the external dataset BIMCV-COVID19 [61] , which is composed of images from 11 hospitals from the Valencia Region, Spain. We randomly selected a subset of 200 patient-level samples covering all hospitals, where 75% of the samples were COVID-19 confirmed cases, as our aim was to shown generalisation to the target disease.

Evaluation Protocol

To validate GraphXCovid we proposed two major experiments to test how our model performs in the low-label regime and how it compares to the current supervised state-

of-the-art. Both experiments shared the same experimental protocol which we detail below.

1. In order to evaluate each model we used an evaluation criterion from both the qualitative and quantitative points of view. The qualitative performance was based on examination of the visual outputs of the classification maps whilst the quantitative performance was based on the per class computation of sensitivity, positive predictive value, F1-score and finally accuracy/error-rate. Furthermore, we report confidence intervals (95%) of all techniques.

2. We evaluated the performances of all models using three different partitions: i) the official partition in which the test set is 300 samples split evenly across the classes, ii) the full COVIDx dataset in which the main differences is that the test set is composed of 1,579 samples (see Fig. 5.3); and iii) an additional randomly generated partition.

3. We ran all experiments under the same conditions, and followed standard pre-processing protocol to normalise the images to have zero mean and unit variance and the images were resized to the resolution 480×480.

Furthermore, when comparing against other semi-supervised methods we follow standard protocol in semi-supervised learning. That is we randomly generate five different labelled subsets of the data and report the mean error and standard deviation over the generated splits.

Implementation Details

We now provide the implementation details for both our approach and comparison methods. For the COVID-Net [226] method we used the implementation and parameters provided in the original paper. In particular, we used the latest suggested model COVIDNet-CXR3-B. For the compared techniques we train the model using Stochastic Gradient Descent (SGD) for 250 epochs with weight decay $\omega = 5e-4$, momentum $p = 0.9$ and learning rate $l_r = 1e-2$ with the exception being $l_r = 1e-3$ for the semi-supervised method "Pseudo-Labelling" [131] and ResNet-18 to guarantee a fair comparison

For our own technique we used the same graphical construction and diffusion parameters as GraphX$^{\text{Net}}$ apart from the fact we used 150 steps of the primal-dual algorithm for each time step. We trained the model, a ResNet-18 architecture, using stochastic gradient descent with momentum with the following parameters: momentum $p = 0.9$, weight decay of $\omega = 2 \times 10^{-4}$, an initial learning rate of $l_r = 5e-2$. We also trained our model for

250 total epochs for a fair comparison. Additionally we used a batch size of $b = 100$ with $b_l = 50$ and $b_u = 50$.

Results and Discussion

For our first experiment we evaluated and compared the different methods on the official COVIDx partition. For comparison models we choose a range of supervised methods which included the leading fully supervised paper in the field COVID-Net [226] and a standard supervised learning algorithm on a range of neural network architectures: VGG-16 [196], ResNet-18 and ResNet-50 [94], InceptionV3 [203] and DenseNet-121 [104]. Given that, to the best of our knowledge, there exists no semi-supervised technique dedicated to COVID-19 identification, we also adapted one semi-supervised technique, Pseudo-Labelling [131] to the task of COVID detection. Pseudo-Labelling shares a pseudo-label philosophy close to ours but is not graphical and instead focuses on entropy minimisation. To highlight the power of semi-supervised methods, the supervised approaches were trained using all available labelled patient data, whilst the semi-supervised methods were only given 30% of the patients as labelled data with the rest being provided as unlabelled data.

We provide a detailed quantitative analysis of the performance of the different techniques in Table 5.3 which gives the per class metrics across the official data partition. Concerning positive predictive values, we observe that our GraphXCovid approach performs the best for healthy and pneumonia classes and it readily competes with COVID-Net in the COVID-19 class, despite using a fraction of the available labels. Furthermore, GraphXCovid outperforms the other semi-supervised method, Pseudo-Labelling, by quite some margin. From the other supervised methods we see that the InceptionV3 and DenseNet-121 architectures comes close to the performance of COVID-Net whilst the others are some way off. In terms of the sensitivity metric. GraphXCovid reports the highest values for pneumonia and COVID-19. For COVID-19, the true positive proportion is higher for our method (0.94) and COVID-Net (0.91) compared to the supervised baselines and pseduo-labelling method ($\leq$0.88). Finally we compare the F1-scores for all methods. Again we see the supervised baselines, aside from COVID-Net are not sufficiently competitive, whereas COVID-Net and our GraphXCovid technique are readily performing at a high level.

Offering a scenario closer to a real medical setting, we also used the full COVIDx partition. We compared our method with the supervised approach of COVID-Net, As reported in Table 5.4, GraphXCovid here performs better for all classes and all considered metrics, while only using 30% of the labelled set. Whilst for sensitivity results our method outperforms COVID-Net for both healthy and COVID-19 infected patients. We note that

Technique	Paradigm SL	Paradigm SSL	Class	PPV	Sensitivity	F1-Scores	Accuracy in 10^{-2}
VGG-16 [196]	✓		Healthy	0.83	0.95	0.88	81.3
			Pneumonia	0.76	0.89	0.82	
			COVID-19	0.88	0.60	0.71	
ResNet-18 [94]	✓		Healthy	0.84	0.91	0.87	86.7
			Pneumonia	0.85	0.90	0.87	
			COVID-19	0.90	0.79	0.84	
Pseudo-Labelling [131]		✓	Healthy	0.90	0.90	0.90	87.3
			Pneumonia	0.88	0.85	0.87	
			COVID-19	0.84	0.86	0.84	
ResNet-50 [94]	✓		Healthy	0.88	0.97	0.92	90.0
			Pneumonia	0.87	0.91	0.89	
			COVID-19	0.95	0.82	0.88	90.6†
InceptionV3 [203]	✓		Healthy	0.93	0.91	0.92	91.0
			Pneumonia	0.90	0.92	0.91	
			COVID-19	0.92	0.88	0.90	
DenseNet-121 [104]	✓		Healthy	0.92	0.94	0.93	91.7
			Pneumonia	0.90	0.93	0.91	
			COVID-19	0.93	0.88	0.90	
COVID-Net [226]	✓		Healthy	0.90	0.95	0.93	93.3
			Pneumonia	0.91	0.94	0.93	
			COVID-19	0.98	0.91	0.95	
GraphXCovid		✓	Healthy	0.93	0.92	0.98	**94.6**
			Pneumonia	0.96	0.96	0.92	
			COVID-19	0.95	0.94	0.95	

Table 5.3 Numerical comparison of our technique vs fully supervised approaches. The results report per class metrics, including sensitivity, positive predictive value and F1-scores along with the overall accuracy. Our technique readily competes with all supervised techniques whilst using far less labelled data. † denotes the score reported in [226].

	Healthy	Pneumonia	COVID-19
	Positive Predictive Value (PPV)		
COVID-Net	0.96	0.91	0.93
GraphXCovid	0.97	0.92	0.95
	Sensitivity		
COVID-Net	0.93	0.95	0.91
GraphXCovid	0.94	0.95	0.93
	F1-Scores		
COVID-Net	0.95	0.93	0.92
GraphXCovid	0.96	0.94	0.94

Table 5.4 Performance comparison of COVID-Net and our technique using the full dataset partition for COVIDx.

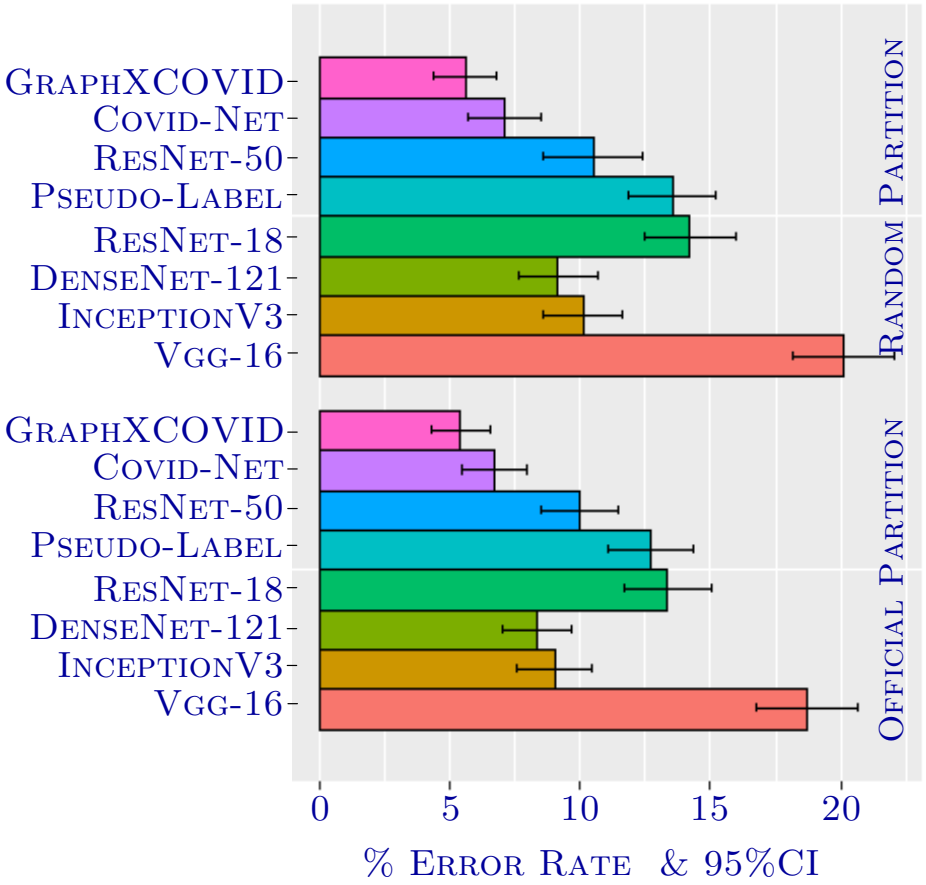


Fig. 5.4 Error rates on all models considered on the COVIDx dataset. Error rates are given on both the official partition and a randomly generated partition, split along patient data to ensure fairness. GraphXCovid and Pseudo-Label were trained with 30% of the labelled data available, the rest being provided as unlabelled examples, whilst all others methods were provided 100% of the labelled data. Despite this, our technique reported the lowest error rate (95%CI) which was consistent across both splits.

overall across the full COVIDx partition our approach improves its performance in respect to the supervised comparison.

To further support the previous results and to give a global performance view, we compute the error rate and the confidence intervals for both the official partition and a randomly generated partition. for each model. These results are reported in Fig. 5.4 For both experiments, VGG-16 reported the worst performance followed by ResNet-18. Our model performed the best among all the compared models, reporting an error of 5.4 ± 1.1 at the 95% confidence level. We also note that all supervised methods performed slightly worse on the random partition then on the official partition whilst for our approach the variation was negligible. This reconfirms that supervised methods are more dependent on

Fig. 5.5 Performance comparison of our technique and COVIDNet on the external BIMCV-COVID19 [61] dataset. The left hand side displays the error rate for the COVID-19 class whilst the right hand side shows the overall performance

the labelled data provided to them whilst semi-supervised learning can use the unlabelled data to reduce this dependence.

Throughout this section we note that our approach greatly outperforms the other SSL technique that was used as a comparison method, Pseudo-Labelling [131]. From Table 5.3, one can observe that GRAPHXCOVID offers a substantial improvement over Pseudo-Labelling for all metrics. Overall, we achieve better accuracy with an improvement of 8%, and reduce the error rate ($\pm$1.20CI) by more than half as displayed in Fig 5.4. We belief that this performance increase is driven by two factors. The work of [131] provides naive pseudo-labels directly from the network itself whilst our technique generates pseudo-labels in a more complex diffusion model based on the $p = 1$ graph Laplacian. Additionally, we incorporate pseudo-label certainty into the loss function by means of a certainty weight. Secondly, our technique includes methods for dealing with imbalanced datasets which the compared method does not.

As a final experiment, we test the generalisation capability of our approach and its closest competitor COVIDNet on *external datasets*. The methods are not trained on the external datasets in any way prior to evaluation, which is vitally important for real world evaluation of medical learning. For the external dataset we use the BIMCV-COVID19 [61] dataset and report the error rate at the 95% confidence level in Figure 5.5. Whilst both methods perform worse on the external dataset, an expected behaviour of deep learning models and particularly noticed in the medical domain e.g. [228], we see that the decrease in performance for COVIDNet is much greater than for our own model. *Why is our model more robust to external data?* Our technique has been carefully designed to mitigate, at some level, out of distribution samples by weighting instances with their confidence

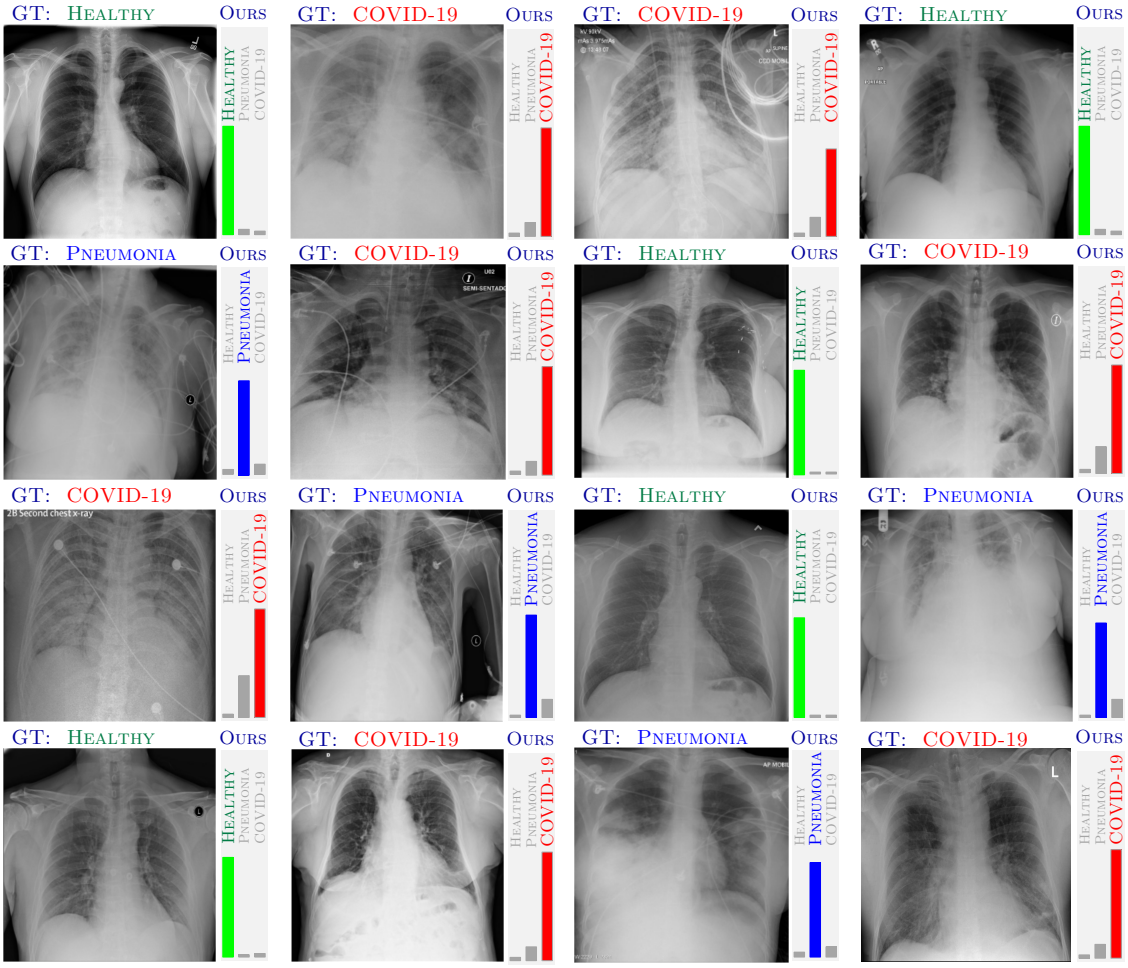


Fig. 5.6 Visualisation of correct predictions on the COVIDx dataset. The probability score, most likely class and the human diagnostic (GT) are compared for each image. We see from the confidence measures, that the model has clearly separated normal X-rays from infected X-rays as well as distinguished between pneumonia and COVID-19.

scores. Contrary, COVIDNet is a fully supervised method and as such is more prone to overfitting to the provided data.

Qualitative Analysis

Alongside our quantitative analysis we also provide detailed qualitative analysis into the visual results produced by our model. In Fig. 5.6, we present example scans where the probability scores of our technique corresponded to the correct human prediction (GT). Our method is clearly able to differentiate between healthy and disease scans as well as detect COVID-19 itself. However, our model is not perfect and incorrect classifications were also generated. We present a selection of these incorrect predictions in Fig. 5.7. Inac-

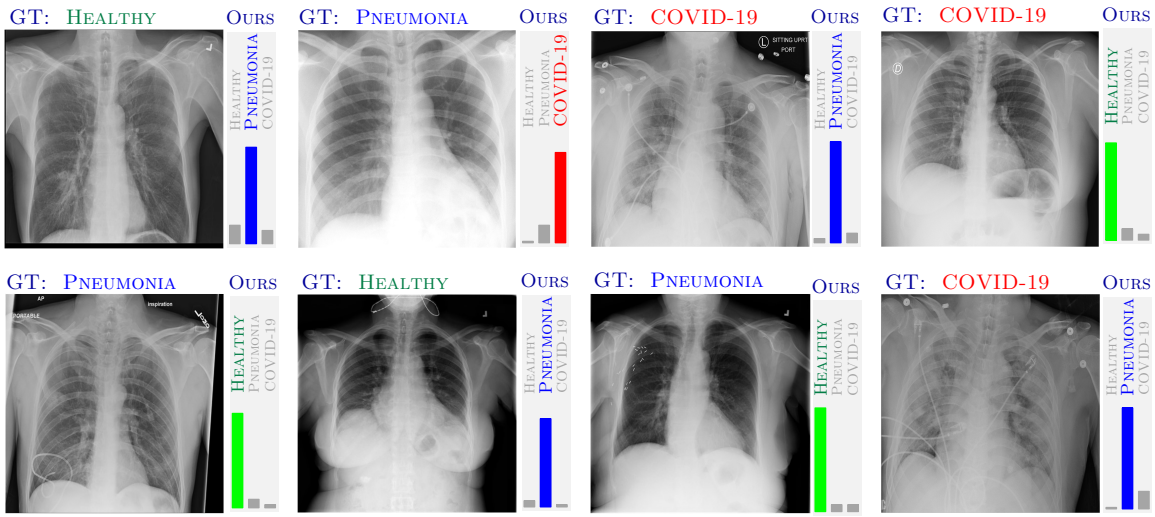


Fig. 5.7 Visualisation of incorrect predictions on the COVIDx dataset. The probability score, most likely class and the human diagnostic (GT) are compared for each image. Despite the success of the model, there is still confusion between health and unhealthy scans as well as pneumonia and COVID-19.

curacies can be caused by several different reasons: the chest X-ray machine generating a poor projection by introducing artifacts, blurry effects and noise in the chest X-ray images, different acquisition protocols and vendor machines being used across the dataset and finally errors in the learning process itself.

Given this potential for mis-classification, probability scores are not the most helpful tool to provide radiologists making clinical decisions. Rather, the highlighting of potential abnormal areas is a much more powerful tool to assisting clinicians in making informed decision. To incorporate this into our framework we use a Gradient-weighted Class Activation Mapping [191] solution to highlight abnormal and normal areas in the lungs. We provide several sample outputs of the attention maps in Fig 5.8. Corresponding to the mental model of radiologists, we project the attention maps on top of the chest X-rays to highlight abnormal regions of the lungs. This tool is designed to help the radiologist in making their decision in a simple user-friendly interface.

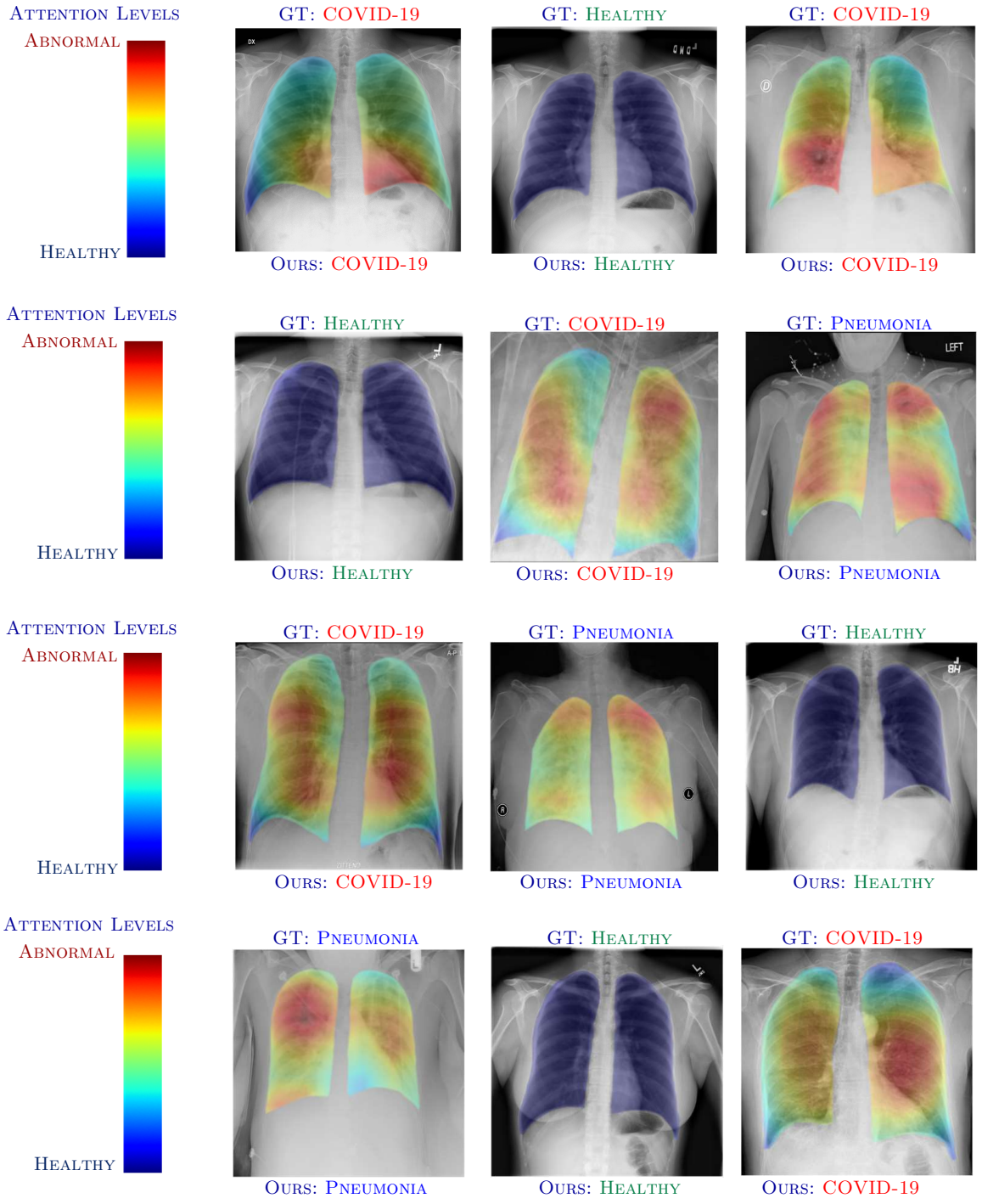


Fig. 5.8 Visualisation of the attention maps produced by our model overlaid on the corresponding chest X-ray image. We also provide the prediction output along with the ground truth (GT) which denotes the human consensus prediction. The attention maps focus in on abnormal regions in the lung, shown by red colours, which can be used to assist the radiologist in making decisions.

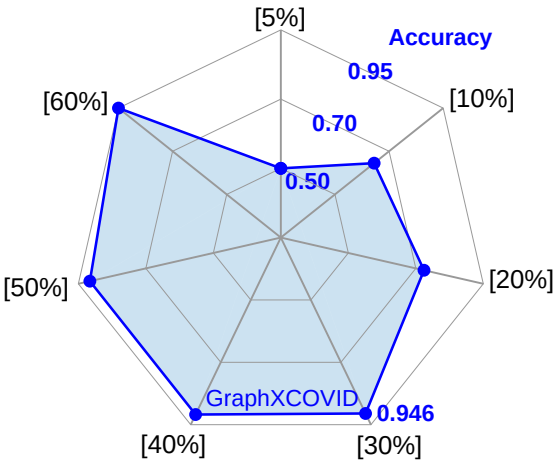


Fig. 5.9 Performance comparison of our technique under different percentage of labelled data. The performance of the model increases with more labelled data, but the increase in performance slows drastically past 30%.

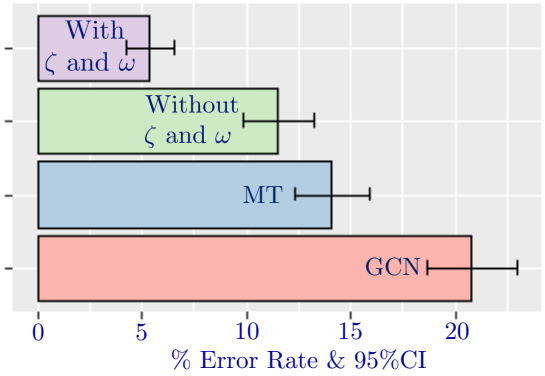


Fig. 5.10 Performance comparison of our technique (with and without including our weighting factors) against two deep SSL techniques GCN [124] and MT [205]. The results are reported, using 30% of labelled data, in terms of the error rate (95%)

Ablation Study

In addition to the main experiments discussed above we also perform an ablation study to test how our model performs when given different amounts of labelled patient data and the importance of different components within the model.

Firstly, we compared the accuracy of our approach on the official partition of COVIDx using different amounts of the available labelled data, ranging from 5% to 60%. Choosing such a range tests the semi-supervised aspects of our model, in that we ask: how much data do we need to have clinically assessed before we can generalise well? We present the results for this experiment in Fig 5.9 Unfortunately, we found that when the amount of labelled data used was very low, $\leq$ 10% of the available amount, the classification accuracy rapidly decreased. Demonstrating that a strong generalisation gap existed, even for SSL methods, when the labelled data was not representative of the full distribution. We also found limited performance improvements when the amount of labelled data used was $\geq$ 30%, demonstrating the ability of our SSL approach to generalise well from small amounts of labelled data.

To further test the generalisation of our approach we compared our approach against two more popular deep semi-supervised works: "Mean Teacher" (MT) [205] and "Graphical Convolutional Networks" (GCN) [124]. We considered the same label count of 30% and used the parameters suggested in these works. The results are reported in Fig 5.10.

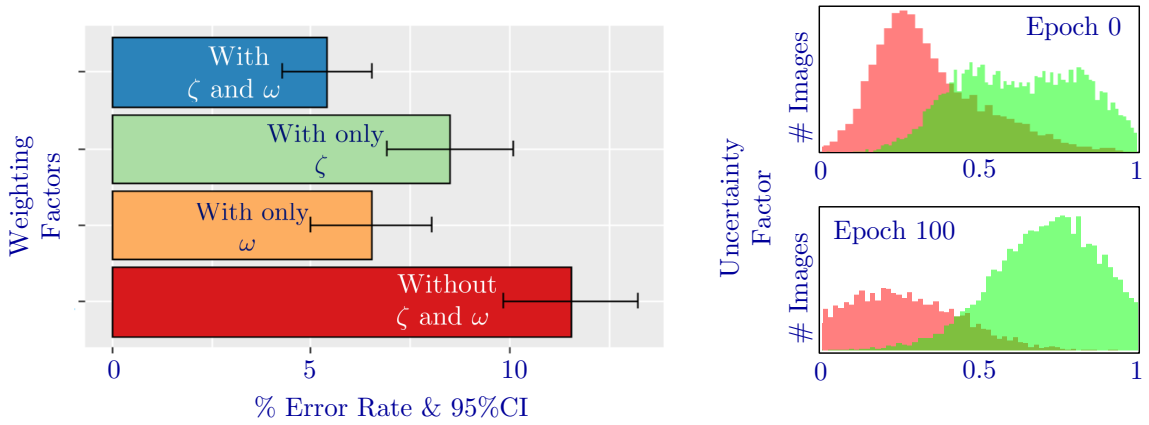


Fig. 5.11 Ablation study on the influence of the uncertainty and class balancing factors, ω and ζ, in the performance of our model. The plot displays the error rate with 95% confidence interval when training with 30% of the available labels. On the right hand side we show how the value of the uncertainty factor ω changes over time for both incorrectly (red) and correctly (green) guessed pseudo-labels.

The GCN approach ultimately failed to be competitive with the other considered methods. We believe this is due to the GCN's lack of specialisation in extracting image features. MT competes readily with the basic version of our model, i.e. without uncertainty and balance parameters but our full model is by some margin the best method considered. Supporting our view that GraphXCovid achieves generalisation for the task of COVID detection.

We also create an ablation study regarding the influence of the weighting factors, ω and ζ, in our model. Using 30% of labelled data we compared the error rate of our model when considering one, both and none of the two factors. The results are reported at the left side of Fig 5.11. We see that both factors improve the performance of the model with entropy weighting ω providing slightly more benefit than class balancing as it promotes correctly labelled pseudo-labels whilst reducing the effect of incorrect guesses. This effect is illustrated on the right side of Fig 5.11. In this illustration, we display ω for all the unlabelled samples for the first and hundredth epoch of learning. The red and green colours denote incorrect and correct pseudo-labels, with respect to the ground truth. From this figure, we can observe that the certainty in the pseudo-labels is improved as the epochs evolve.

5.6 Conclusions and Further Work

In this Chapter we explored the topic of semi-supervised learning for medical image diagnosis, with an application to chest X-rays. Our motivation was that the medical imaging domain is a very complex setting which uniquely suffers from problems relating to the acquiring labelled data. Therefore, the development and deployment of semi-supervised learning frameworks is key to the progress of machine learnt medical diagnosis.

Due to the complexity of the setting and the fact that accuracy is of vital importance, we approached this problem using the $p = 1$ graph Laplacian which has been shown to be a powerful and accurate tool for label diffusion and node classification. We implemented an accelerated primal-dual algorithm for optimising the $p = 1$ graphical Laplacian in a general C class optimisation problem. We then included this approach into both a transductive framework and a hybrid graphical neural network inductive framework.

The transductive framework worked by taking a pre-trained model and using its feature representation of the dataset to construct a weighted undirected graphical representation. From this we performed label diffusion from the initially label points to the whole graph. Due to the limits of transductive frameworks, such as slow inference speed, we expanded this work to an inductive framework. This inductive framework worked in a cyclical process where the pseudo-labels generated by label diffusion of the $p = 1$ graphical Laplacian were used to train a neural network in a semi-supervised manner. The updated model was then used to construct a new graphical representation which produces more accurate pseudo-labels which were again used to train the network in an iterative manner.

We used the transductive framework to tackle the general problem of chest X-ray diagnosis and the inductive framework to tackle the issue of COVID-19 diagnosis. We performed a detailed quantitative and qualitative experimental evaluations of our frameworks including detailed ablation studies. Overall, for both problems, our approaches were able to outperform state-of-the-art supervised approaches whilst using a fraction of the amount of labelled data. Not only does this speak to the potential deployment of X-ray diagnosis for COVID-19 detection, it highlights the overall success that semi-supervised learning could have on reducing the barriers for deployment of machine learning frameworks in the medical domain.

However, this research is only a tiny part of the bigger picture of automating medical decisions. Further work is needed to improve the explainability of machine learning systems for them to be accepted at a patient level and similar work is needed to be done on uncertainty quantification to understand when the output of machine learning systems may not be useful in the clinical setting.

Chapter 6

Conclusion and Outlook

This thesis tackles a common issue in machine learning: labelled data is hard to acquire. We highlight the potential for deep semi-supervised learning to overcome this hurdle for image classification by reducing the requirement for labelled data. Through the research demonstrated in this thesis, it has been shown that in a variety of different domains, semi-supervised learning can perform comparably or outperform supervised models whilst requiring a fraction of the labelled data. This research led to the publication of several works including [190, 188, 189, 12, 15]. In particular, we demonstrated that working directly or indirectly with graphical representations of data, in the case of transductive learning or pseudo-label generation, is a powerful tool for producing high performing classification models. We conclude this work by recapping the major contributions of each of the three research chapters before offering thoughts on avenues of further work for deep semi-supervised classification.

Graph-Based Hyperspectral Image Classification

In Chapter 3 we proposed a classical graphical propagation approach for classifying hyperspectral images which resulted in the publication of [190]. This approach was built around designing and implementing a novel superpixel segmentation algorithm for hyperspectral data which preserved edge structures and grouped pixels similar in colour space. By over-segmenting images in this way we constructed a region-based graph, rather than a pixel-based graph, which has a far smaller number of nodes. We then used the $p = 2$ graph-Laplacian to propagate information across this superpixel graph. We demonstrated through experiments on benchmark datasets that our approach was able to outperform the current state-of-the-art with as little as 3 labels per class. Highlighting that clever

adaption to the image domain can improve the outcome of graphical semi-supervised methods.

Pseudo-Labelling approaches for Natural Image Classification

In Chapter 4 we explored the domain of natural image classification. Recent works such as Fixmatch [197] and UDA [193] have shown great success using costly complex approaches whilst other works such as the work of [8] have proposed technical tricks to improve pseudo-label methods. In this Chapter we aimed to use the advantages of graphical semi-supervised learning to remove these arbitrary tricks and unnecessary complexity whilst improving performance. To this end we proposed two approaches. Firstly, *CycleCluster* [188] proposed using a direct implementation of cluster regularisation rather than the domain specific perturbation regularisation. We demonstrate that our approach is competitive with perturbation approaches whilst being resistant to the number of clusters selected. In our second approach, *LaplaceNet* [189], we theoretically explored augmentation techniques and justified and experimentally validated a multi-sample approach. Combining this augmentation strategy with a hybrid energy neural-network model, we were able to produce state-of-the-art results on several benchmarks datasets whilst removing the need for several technical tricks, such as temperature scaling and confidence thresholding, which were thought to be essential for good performance. Additionally, we performed several ablations including showing the advantage of graphical pseudo-labels over network based alternative.

Medical Image Classification with the $p = 1$ graph Laplacian

In the final research Chapter, we investigated the topical task of medical image classification. Medical imaging particularly suffers from a lack of labelled data due to the complexity in training medical professionals and so developing semi-supervised algorithms should massively speed up the deployment of machine learnt medical systems. In this Chapter, we worked with the computationally more complex $p = 1$ graph Laplacian energy for the task of node classification due to recent work showing its superiority to other graphical methods. We implemented an accelerated primal-dual optimisation algorithm to minimise this energy. Using this scheme, we created both a transductive framework, which we applied to general chest x-ray diagnosis [12] and, via inclusion of a neural network backbone, an inductive framework which we use to classify COVID-19 in chest x-rays [15]. To the best of our knowledge our works were the first to encourage deep semi-supervised techniques for the medical domain and COVID-19 in particular. We demonstrated on

benchmark datasets that our approach was able to outperform state-of-the-art supervised methods whilst requiring a fraction of the labels.

6.1 Further Work

Despite our success in applying graphical semi-supervised learning to a varied range of domains, there are still many open problems for improving the success of deep graphical semi-supervised methods in the imaging domain. In this section we highlight a couple of these issues that were present in both our research and the field.

1. **Graph construction:** In all approaches in this thesis, we have taken a structure that exists in some feature space and formed a graphical representation of this structure, typically by using some metric and K-nearest neighbours, despite the fact that the graph didn't exist naturally. Although this approach appears to generally work well, there is currently a lack of understanding on how to construct effective graphical representation for imaging data. As the graph is constructed, it would appear from experimental results that there is little benefit to using deep graphical models to propagate label information compared to classical graphical energies such as the p-graph Laplacian. Therefore, the construction could be the major bottleneck in performance.

2. **Strong data augmentation:** The success of semi-supervised learning techniques in natural imaging has become dependent on *strong data-augmentations*. However, it is unclear how we generalise these augmentations to drastically different imaging domains such as hyperspectral or medical data and such generalisations pose many interesting questions. Such as: Is there always a set of transformations that we could use to significantly increase performance? Do we need to rely upon expert opinion to build up libraries of transformation, as was done for natural imaging, or can the transformations themselves be learnt? Does there exists a set of learnt transformations which are better than strong-augmentations for the natural image domain and what do these transformations represent? Answering these questions may be key in improving the deployment of semi-supervised learning to more complex domains. From the theoretical perspective there has been some early developments, such as the work of [47, 60], but this represents a vastly unexplored area for future study.

References

[1] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. (2012a). Slic superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, pages 2274–2282.

[2] Achanta, R., Shaji, A., Smith, K., Lucchi, A., Fua, P., and Süsstrunk, S. (2012b). Slic superpixels compared to state-of-the-art superpixel methods. *IEEE transactions on pattern analysis and machine intelligence*.

[3] Afshar, P., Heidarian, S., Naderkhani, F., Oikonomou, A., Plataniotis, K. N., and Mohammadi, A. (2020). Covid-caps: A capsule network-based framework for identification of covid-19 cases from x-ray images. *arXiv preprint arXiv:2004.02696*.

[4] Aiazzi, B., Alparone, L., and Baronti, S. (2012). Spectral distortion in lossy compression of hyperspectral data.

[5] Albawi, S., Mohammed, T. A., and Al-Zawi, S. (2017). Understanding of a convolutional neural network. In *2017 International Conference on Engineering and Technology (ICET)*, pages 1–6. Ieee.

[6] Amghibech, S. (2003). Eigenvalues of the discrete p-laplacian for graphs. *Ars Combinatoria*, 67:283–302.

[7] Apostolopoulos, I. D., Aznaouridis, S. I., and Tzani, M. A. (2020). Extracting possibly representative covid-19 biomarkers from x-ray images with deep learning approach and image data related to pulmonary diseases. *Journal of Medical and Biological Engineering*, page 1.

[8] Arazo, E., Ortego, D., Albert, P., O'Connor, N. E., and McGuinness, K. (2019). Pseudo-labeling and confirmation bias in deep semi-supervised learning. *arXiv preprint arXiv:1908.02983*.

[9] Argyriou, A., Herbster, M., and Pontil, M. (2005). Combining graph laplacians for semi-supervised learning. In *NIPS*, volume 5, pages 67–74. Citeseer.

[10] Arsigny, V., Fillard, P., Pennec, X., and Ayache, N. (2006). Geometric means in a novel vector space structure on symmetric positive-definite matrice. *SIAM J. Matrix Anal. Appl.*, 29(1):328–347.

[11] Athiwaratkun, B., Finzi, M., Izmailov, P., and Wilson, A. G. (2019). There are many consistent explanations of unlabeled data: Why you should average. *International Conference on Learning Representations (ICLR)*.

[12] Aviles-Rivero, A. I. and et al. (2019). Graphx-net - chest x-ray classification under extreme minimal supervision. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 504–512.

[13] Aviles-Rivero, A. I., Papadakis, N., Li, R., Alsaleh, S. M., Tan, R. T., and Schonlieb, C.-B. (2019). Beyond supervised classification: Extreme minimal supervision with the graph 1-laplacian. *arXiv:1906.08635*.

[14] Aviles-Rivero, A. I., Papadakis, N., Li, R., Alsaleh, S. M., Tan, R. T., and Schonlieb, C.-B. (2020). When labelled data hurts: Deep semi-supervised classification with the graph 1-laplacian.

[15] Aviles-Rivero, A. I., Sellars, P., Schönlieb, C.-B., and Papadakis, N. (2022). Graphxcovid: Explainable deep graph diffusion pseudo-labelling for identifying covid-19 on chest x-rays.

[16] Baltruschat, I., Nickisch, H., Grass, M., Knopp, T., and Saalbach, A. (2019). Comparison of deep learning approaches for multi-label chest x-ray classification. *Scientific reports*, pages 1–10.

[17] Bar, Y., Diamant, I., Wolf, L., Lieberman, S., Konen, E., and Greenspan, H. (2015). Chest pathology detection using deep learning with non-medical training. *IEEE International Symposium on Biomedical Imaging*, pages 294–297.

[18] Belkin, M., Matveeva, I., and Niyogi, P. (2004). Regularization and semi-supervised learning on large graphs. In *International Conference on Computational Learning Theory*, pages 624–638. Springer.

[19] Belkin, M., Niyogi, P., and Sindhwani, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of machine learning research*, 7.

[20] Benardos, P. and Vosniakos, G.-C. (2007). Optimizing feedforward artificial neural network architecture. *Engineering applications of artificial intelligence*, 20(3):365–382.

[21] Bengio, Y., Courville, A., and Vincent, P. (2013). Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828.

[22] Berthelot, D., Carlini, N., Cubuk, E., Kurakin, A., Zhang, H., and Raffel, C. (2020). Remixmatch: Semi-supervised learning with distribution matching and augmentation anchoring. In *Eighth International Conference on Learning Representations*.

[23] Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., and Raffel, C. A. (2019a). Mixmatch: A holistic approach to semi-supervised learning. In *Advances in Neural Information Processing Systems*.

[24] Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., and Raffel, C. A. (2019b). Mixmatch: A holistic approach to semisupervised learning. In *In Advances in Neural Information Processing Systems*.

[25] Bertozzi, A. and Flenner, A. (2012). Diffuse interface models on graphs for classification of high dimensional data. *Multiscale Modeling and Simulation*, 10(3):1090–1118.

[26] Bottou, L. (2012). Stochastic gradient descent tricks. In *Neural networks: Tricks of the trade*, pages 421–436. Springer.

[27] Bowles, C., Chen, L., Guerrero, R., Bentley, P., Gunn, R., Hammers, A., Dickie, D. A., Hernández, M. V., Wardlaw, J., and Rueckert, D. (2018). Gan augmentation: Augmenting training data using generative adversarial networks.

[28] Bresson, X., Laurent, T., Uminsky, D., and Von Brecht, J. (2012). Convergence and energy landscape for Cheeger Cut clustering. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1385–1393.

[29] Bresson, X., Laurent, T., Uminsky, D., and Von Brecht, J. (2013a). Multiclass total variation clustering. In *Advances in Neural Information Processing Systems*.

[30] Bresson, X., Laurent, T., Uminsky, D., and Von Brecht, J. H. (2013b). An adaptive total variation algorithm for computing the balanced cut of a graph. *arXiv preprint arXiv:1302.2717*.

[31] Brosch, T. and R., T. (2013). U-net: convolutional networks for biomedical image segmentation. *Proceedings of the Medical Image Computing and Computer-Assisted Intervention*, 8150:633–640.

[32] Bruno, M. A., Walker, E. A., and Abujudeh, H. H. (2015). Understanding and confronting our mistakes: the epidemiology of error in radiology and strategies for error reduction. *Radiographics*, pages 1668–1676.

[33] Bruzzone, L., Chi, M., and Marconcini, M. (2005). Transductive svms for semisupervised classification of hyperspectral data. *International Geoscience and Remote Sensing Symposium*.

[34] Bühler, T. and Hein, M. (2009a). Spectral clustering based on the graph p-laplacian. *International Conference on Machine Learning*.

[35] Bühler, T. and Hein, M. (2009b). Spectral clustering based on the graph p-laplacian. *International Conference on Machine Learning (ICML)*.

[36] Calder, J., Cook, B., Thorpe, M., and Slepcev, D. (2020). Poisson learning: Graph based semi-supervised learning at very low label rates. In *International Conference on Machine Learning*, pages 1306–1316. PMLR.

[37] Campos, G. O., Zimek, A., Sander, J., Campello, R. J., Micenková, B., Schubert, E., Assent, I., and Houle, M. E. (2016). On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. *Data mining and knowledge discovery*, 30(4):891–927.

[38] Camps-Valls, G., Marsheva, B., and Zhou, D. (2007a). Semi-supervised graphbased hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.*, 45(10):3044–3054.

[39] Camps-Valls, G., Marsheva, T., and Zhou, D. (2007b). Semi-supervised graphbased hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.*, 45(10):3044–3054.

[40] Caron, M., Bojanowski, P., Joulin, A., and Douze, M. (2018). Deep clustering for unsupervised learning of visual features. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 132–149.

[41] Chambolle, A. and Pock, T. (2011). A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of mathematical imaging and vision*.

[42] Chang, C. C. and Lin, C. J. (2011). Libsvm: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.*, 2(3):27:1–27:27.

[43] Chapelle, O., Weston, J., Bottou, L., and Vapnik, V. (2000). Vicinal risk minimization.

[44] Chapelle, O., Zien, A., and Schölkopf, B. (2006). *Semisupervised learning*. MIT Press.

[45] Chen, J., Yang, L., Zhang, Y., Alber, M., and Chen, D. (2016). Combining fully convolutional and recurrent neural networks for 3d biomedical image segmentation. *Proceedings of the Advances in Neural Information Processing Systems*, pages 3036–3044.

[46] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., and Yuille, A. L. (2017). Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 40(4):834–848.

[47] Chen, S., Dobriban, E., and Lee, J. H. (2020). A group-theoretic framework for data augmentation. *Journal of Machine Learning Research*, 21(245):1–71.

[48] Chen, Y., Nasrabadi, N. M., and Tran, T. D. (2011). Hyperspectral image classification using dictionary-based sparse representation. *IEEE Transactions on Geoscience and Remote Sensing*, 49(10):3973–3985.

[49] Chen, Y., Wang, G., and Dong, S. (2003). Learning with progressive transductive support vector machine. *Pattern Recognition Letters*, 24(12):1845–1855.

[50] Cheplygina, V., de Bruijne, M., and Pluim, J. P. (2019). Not-so-supervised: A survey of semi-supervised, multi-instance, and transfer learning in medical image analysis. *Medical Image Analysis*, 54:280–296.

[51] Chowdhury, M. and et al. (2020). Covid-19 radiography database. *[Online] Available: https://www.kaggle.com/tawsifurrahman/covid19-radiography-database*.

[52] Chung, M. and et al. (2020). Ct imaging features of 2019 novel coronavirus (2019-ncov). *Radiology*, 295(1):202–207.

[53] Ciurte, A., Bresson, X., Cuisenaire, O., Houhou, N., Nedevschi, S., Thiran, J., and Cuadra, M. (2014). Semi-supervised segmentation of ultrasound images based on patch representation and continuous min cut. *PLoS ONE.*, 9.

[54] Cohen, J. P., Morrison, P., and Dao, L. (2020). Covid-19 image data collection. *arXiv preprint arXiv:2003.11597*.

[55] Cohen, T. and Welling, M. (2016). Group equivariant convolutional networks. In *International conference on machine learning*, pages 2990–2999. PMLR.

[56] Cormen, T., Leiserson, C., and Rivest, R. (1990). Introduction to algorithms.

[57] Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., and Le, Q. V. (2018). Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*.

[58] Cubuk, E. D., Zoph, B., Shlens, J., and Le, Q. V. (2020). Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703.

[59] Cui, B., Xie, X., Xiudan, M., Ren, G., and Ma, Y. (2018). Superpixel-based extended random walker for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.*, 56(6):3233–3243.

[60] Dao, T., Gu, A., Ratner, A., Smith, V., De Sa, C., and Ré, C. (2019). A kernel theory of modern data augmentation. In *International Conference on Machine Learning*, pages 1528–1537. PMLR.

[61] De La Iglesia Vayá, M., Saborit, J. M., Montell, J. A., Pertusa, A., Bustos, A., Cazorla, M., Galant, J., Barber, X., Orozco-Beltrán, D., García-García, F., et al. (2020). Bimcv covid-19+: a large annotated dataset of rx and ct images from covid-19 patients. *arXiv preprint arXiv:2006.01174*.

[62] Delalleau, O., Bengio, Y., and Le Roux, N. (2005). Efficient non-parametric function induction in semi-supervised learning. In *AISTATS*, volume 27.

[63] Deng, J., Dong, W., Socher, R., Li, L., and Fei-Fei, L. (2009a). Imagenet: a large-scale hierarchical image database. CVPR.

[64] Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009b). ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.

[65] DeVries, T. and Taylor, G. W. (2017). Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*.

[66] Dodero, L., Gozzi, A., Liska, A., Murino, V., and Sona, D. (2014). Group-wise functional community detection through joint laplacian diagonalization. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 708–715.

[67] Dópido, I., Li, J., Marpu, P., Plaza, A., Dias, J., and Benediktsson, J. (2013). Semisupervised self-learning for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 51(7):4032–4044.

[68] Ellis, R. J. and Scott, P. W. (2004). Evaluation of hyperspectral remote sensing as a means of environmental monitoring in the st. austell china clay (kaolin) region, cornwall, uk. *Remote sensing of environment*, 93(1-2):118–130.

[69] Fang, L., He, N., Li, S., Plaza, A. J., and Plaza, J. (2018a). A new spatial–spectral feature extraction method for hyperspectral images using local covariance matrix representation. *IEEE Transactions on Geoscience and Remote Sensing*, 56(6):3534–3546.

[70] Fang, L., He, N., Li, S., and Plaza, J. (2018b). A new spatial-spectral feature extraction method for hyperspectral images using local covariance matrix representation. *IEEE Trans. Geosci. Remote Sens.*, 56(6):3534–3546.

[71] Fang, L., Li, S., Duan, W., Ren, J., and Benediktsson, J. A. (2015a). Classification of hyperspectral images by exploiting spectral–spatial information of superpixel via multiple kernels. *IEEE Trans. Geosci. Remote Sens.*, 53(12):6663–6674.

[72] Fang, L., Li, S., Kang, X., and Benediktsson, J. A. (2015b). Spectral-spatial classification of hyperspectral images with a superpixel-based discriminative sparse model. *IEEE Transactions on Geoscience and Remote Sensing*, 53(8):4186–4201.

[73] Fang, Y. and et al. (2020). Sensitivity of chest ct for covid-19: comparison to rt-pcr. *Radiology*, page 200432.

[74] Farooq, M. and Hafeez, A. (2020). Covid-resnet: A deep learning framework for screening of covid19 from radiographs. *arXiv preprint arXiv:2003.14395*.

[75] Feld, T., Aujol, J., Gilboa, G., and Papadakis, N. (2019a). Rayleigh quotient minimization for absolutely one-homogeneous functionals. *Inverse Problems*.

[76] Feld, T. M., Aujol, J.-F., Gilboa, G., and Papadakis, N. (2019b). Rayleigh quotient minimization for absolutely one-homogeneous functionals. *Inverse Problems*.

[77] Ferguson, M., Ak, R., Lee, Y.-T. T., and Law, K. H. (2017). Automatic localization of casting defects with convolutional neural networks. In *2017 IEEE international conference on big data (big data)*, pages 1726–1735. IEEE.

[78] Fernandez, A., García, S., Galar, M., Prati, R. C., Krawczyk, B., and Herrera, F. (2018). *Learning from Imbalanced Data Sets*. Springer.

[79] Filippone, M., Camastra, F., Masulli, F., and Rovetta, S. (2008). A survey of kernel and spectral methods for clustering. *Pattern recognition*, 41(1):176–190.

[80] Folio, L. R. (2012). *Chest imaging: an algorithmic approach to learning*. Springer Science & Business Media.

[81] Franquet, T. (2001). Imaging of pneumonia: trends and algorithms. pages 196–208.

[82] Fukushima, K. (1980). Neocognitron: a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cybern*, 36:193–20.

[83] Gao, Y., Adeli-M, E., Kim, M., Giannakopoulos, P., Haller, S., and Shen, D. (2015). Medical image retrieval using multi-graph learning for mci diagnostic assistance. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 86–93.

[84] Gao, Y., Ji, R. M., Cui, P., Dai, Q., and Hua, G. (2011). Hyperspectral image classification through bilayer graph-based learning. *IEEE Transactions on Image Processing*, 23(7):2769–2778.

[85] Girshick, R. (2015). Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448.

[86] Glorot, X., Bordes, A., and Bengio, Y. (2011). Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323. JMLR Workshop and Conference Proceedings.

[87] Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep learning*. MIT press.

[88] Grandvalet, Y. and Bengio, Y. (2005). Semi-supervised learning by entropy minimization. In *Advances in neural information processing systems*, pages 529–536.

[89] Grira, N., Crucianu, M., and Boujemaa, N. (2004). Unsupervised and semi-supervised clustering: a brief survey. *A review of machine learning techniques for processing multimedia content*, 1:9–16.

[90] Gu, Y., Wang, C., You, D., Zhang, Y., Wang, S., and Zhang, Y. (2012). Learn multiple-kernel svms for domain adaptation in hyperspectral data. *IEEE Transactions on Geoscience and Remote Sensing*, 50(7):2852–2865.

[91] Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On calibration of modern neural networks. *CoRR*, abs/1706.04599.

[92] Hawkins, D. M. (2004). The problem of overfitting. *Journal of chemical information and computer sciences*, 44(1):1–12.

[93] He, H. and Ma, Y. (2013). *Imbalanced learning: foundations, algorithms, and applications*. John Wiley & Sons.

[94] He, K., Zhang, X., Ren, S., and Sun, J. (2016a). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

[95] He, K., Zhang, X., Ren, S., and Sun, J. (2016b). Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[96] He, K., Zhang, X., Ren, S., and Sun, J. (2016c). Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[97] He, K., Zhang, X., Ren, S., and Sun, J. (2016d). Identity mappings in deep residual networks. In *In European Conference on Computer Vision*.

[98] Hein, M. and Bühler, T. (2010). An inverse power method for nonlinear eigenproblems with applications in 1-spectral clustering and sparse pca. In *Advances in Neural Information Processing Systems (NIPS)*, pages 847–855.

[99] Hein, M., Setzer, S., Jost, L., and Rangapuram, S. S. (2013). The total variation on hypergraphs-learning on hypergraphs revisited. In *Advances in Neural Information Processing Systems*, pages 2427–2435.

[100] Hemdan, E. E.-D., Shouman, M. A., and Karar, M. E. (2020). Covidx-net: A framework of deep learning classifiers to diagnose covid-19 in x-ray images. *arXiv preprint arXiv:2003.11055*.

[101] Hoeffding, W. (1963). Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30.

[102] Hu, J., Shen, L., and Sun, G. (2018). Squeeze-and-excitation networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7132–7141.

[103] Hu, Z., Yang, Z., Hu, X., and R, N. (2021). Simple: Similar pseudo label exploitation for semi-supervised classification. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[104] Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *IEEE conference on Computer Vision and Pattern Recognition (CVPR)*.

[105] Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. PMLR.

[106] Iscen, A., Tolias, G., Avrithis, Y., and Chum, O. (2019). Label propagation for deep semi-supervised learning. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[107] Jackson, J. and Schulman, J. (2019). Semi-supervised learning by label gradient alignment. *arXiv preprint arXiv:1902.02336*.

[108] Jacobi, A., Chung, M., Bernheim, A., and Eber, C. (2020). Portable chest x-ray in coronavirus disease-19 (covid-19): A pictorial review. *Clinical Imaging*.

[109] Jebara, T., Wang, J., and Chang, S.-F. (2009). Graph construction and b-matching for semi-supervised learning. In *Proceedings of the 26th annual international conference on machine learning*, pages 441–448.

[110] Jia, S., Deng, B., Zhu, J., Jia, X., and Li, Q. (2017). Superpixel-based multitask learning framework for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 55(5):2572–2588.

[111] Jia, S., Deng, B., Zhu, J., Jia, X., and Li, Q. (2018). Local binary pattern-based hyperspectral image classification with superpixel guidance. *IEEE Transactions on Geoscience and Remote Sensing*, 56(2):749–759.

[112] Jia, S., Deng, X., Zhu, J., Xu, M., and Zhou, J Jia, X. (2019a). Collaborative representation-based multiscale superpixel fusion for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.*

[113] Jia, S., Lin, Z., Deng, B., and Zhu, J. (2019b). Cascade superpixel regularized gabor feature fusion for hyperspectral image classification. *IEEE Transactions on Neural Networks and Learning Systems*.

[114] Jimènez, J., Škalič, M., Martínez-Rosell, G., and De Fabritiis, G. (2018). Kdeep: protein-ligand absolute binding affinity prediction via 3d-convolutional neural networks. *J Chem Inf Model*, 58:287–296.

[115] Joachims, T. (2003). Transductive learning via spectral graph partitioning. In *International Conference on Machine Learning (ICML)*, pages 290–297.

[116] Jolliffe, I. T. and Cadima, J. (2016). Principal component analysis: a review and recent developments. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 374(2065):20150202.

[117] Kang, X., Li, S., and Benediktsson, J. A. (2014a). Feature extraction of hyperspectral images with image fusion and recursive filtering. *IEEE Trans. Geosci. Remote Sens.*, 52(6):3742–3752.

[118] Kang, X., Li, S., and Benediktsson, J. A. (2014b). Spectral-spatial hyperspectral image classification with edge-preserving filtering. *IEEE Trans. Geosci. Remote Sens.*, 52(5):2666–2677.

[119] Kawahara, J., Brown, C., Miller, S., Booth, B., Chau, V., Grunau, R., Zwicker, J., and Hamarneh, G. (2016). Brainnetcnn: convolutional neural networks for brain networks; towards predicting neurodevelopment. *Neuroimage*.

[120] Ke, Z., Wang, D., Yan, Q., Ren, J., and Lau, R. W. (2019). Dual student: Breaking the limits of the teacher in semi-supervised learning. In *Proceedings of the IEEE International Conference on Computer Vision*.

[121] Keller, J. M., Gray, M. R., and Givens, J. A. (1985). A fuzzy k-nearest neighbor algorithm. *IEEE transactions on systems, man, and cybernetics*, (4):580–585.

[122] Kim, K. I., Steinke, F., and Hein, M. (2009). Semi-supervised regression using hessian energy with an application to semi-supervised dimensionality reduction. In *Advances in Neural Information Processing Systems (NIPS)*, pages 979–987.

[123] Kingma, D. P., Mohamed, S., Rezende, D. J., and Welling, M. (2014). Semi-supervised learning with deep generative models. In *Advances in neural information processing systems*, pages 3581–3589.

[124] Kipf, T. N. and Welling, M. (2016). Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

[125] Kriegel, H.-P., Kröger, P., Sander, J., and Zimek, A. (2011). Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(3):231–240.

[126] Krizhevsky, A. and Hinton, G. (2009). Learning multiple layers of features from tiny images.

[127] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105.

[128] Kuhlmann, L., Lehnertz, K., Richardson, M., Schelter, B., and Zaveri, H. (2018). Seizure prediction – ready for a new era. *Nat Rev Neurol*.

[129] Kukar, M., Kononenko, I., et al. (1998). Cost-sensitive learning with neural networks. In *Proceedings of the 13th European Conference on Artificial Intelligence*, volume 98, pages 445–449.

[130] Laine, S. and Aila, T. (2017). Temporal ensembling for semi-supervised learning. *International Conference on Learning Representations (ICLR)*.

[131] Lee, D.-H. (2013). Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on Challenges in Representation Learning, ICML*.

[132] Li, S., Liu, B., Chen, D., Chu, Q., Yuan, L., and Yu, N. (2020). Density-aware graph for deep semi-supervised visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13400–13409.

[133] Li, W., Chen, C., Su, H., and Du, Q. (2015). Local binary patterns and extreme learning machine for hyperspectral imagery classification. *IEEE Trans. Geosci. Remote Sens.*, 53(7):3681–3693.

[134] Li, Y., Hao, Z., and Lei, H. (2016). Survey of convolutional neural network. *Journal of Computer Applications*, 36(9):2508–2515.

[135] Likas, A., Vlassis, N., and Verbeek, J. J. (2003). The global k-means clustering algorithm. *Pattern recognition*, 36(2):451–461.

[136] Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A., van Ginneken, B., and Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. *Medical Image Analysis*, 42:60–88.

[137] Liu, M. Y., Tuzel, O., Ramalingam, S., and R, C. (2011). Entropy rate superpixel segmentation. *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2097–2104.

[138] Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., and Alsaadi, F. E. (2017a). A survey of deep neural network architectures and their applications. *Neurocomputing*, 234:11–26.

[139] Liu, Y., Gao, G., and Gu, Y. (2017b). Tensor matched subspace detector for hyperspectral target detection. *IEEE Transactions on Geoscience and Remote Sensing (TGRS)*, 55(4):1967–1974.

[140] Liu, Y. J., Yu, C., Yu, M., and He, Y. (2016). Manifold slic: A fast method to compute content-sensitive superpixels. *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pages 651–659.

[141] Lloyd, S. (1982). Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137.

[142] Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440.

[143] Lu, D. and Weng, Q. (2007). A survey of image classification methods and techniques for improving classification performance. *International journal of Remote sensing*, 28(5):823–870.

[144] Lundervold, A. S. and Lundervold, A. (2019). An overview of deep learning in medical imaging focusing on mri. *Zeitschrift für Medizinische Physik*, 29(2):102–127.

[145] Luo, Y., Zhu, J., Li, M., Ren, Y., and Zhang, B. (2018). Smooth neighbors on teacher graphs for semi-supervised learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8896–8905.

[146] Mahapatra, D., Vos, F., and Buhmann, J. (2016). Active learning based segmentation of crohns disease from abdominal mri. *Comput. Methods Programs Biomed.*, 128:75–85.

[147] Makantasis, K., Karantzalos, K., Doulamis, A., and Doulamis, N. (2015). Deep supervised learning for hyperspectral data classification through convolutional neural networks. *IEEE Int. Geosci. Remote Sens. Symp. Italy*, pages 4959–4962.

[148] Manfreda, S., McCabe, M., Miller, P., et al. (2018). On the use of unmanned aerial systems for environmental monitoring. *Remote Sensing*, 10(4):641.

[149] Melgani, F. and Bruzzone, L. (2004). Classification of hyperspectral remote sensing images with support vector machines. *IEEE Transactions on Geoscience and Remote Sensing*, 42(8):1778–1790.

[150] Mercier, G. and Lennon, M. (2003). Support vector machines for hyperspectral image classification with spectral-based kernels. *Proceedings of the International IEEE Geoscience and Remote Sensing Symposium*, 1:288–290.

[151] Miyato, T., Dai, A. M., and Goodfellow, I. (2017). Adversarial training methods for semi-supervised text classification. *International Conference on Learning Representations (ICLR)*.

[152] Miyato, T., Maeda, S.-i., Koyama, M., and Ishii, S. (2018). Virtual adversarial training: a regularization method for supervised and semi-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 41(8):1979–1993.

[153] Mohri, M., Rostamizadeh, A., and Talwalkar, A. (2018). *Foundations of machine learning*. MIT press.

[154] Moon, T. K. (1996). The expectation-maximization algorithm. *IEEE Signal processing magazine*, 13(6):47–60.

[155] Müller, B., Reinhardt, J., and Strickland, M. T. (1995). *Neural networks: an introduction*. Springer Science & Business Media.

[156] Nadler, B., Srebro, N., and Zhou, X. (2009). Semi-supervised learning with the graph laplacian: The limit of infinite unlabelled data. *Advances in neural information processing systems*, 22:1330–1338.

[157] Narin, A., Kaya, C., and Pamuk, Z. (2020). Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks. *arXiv preprint arXiv:2003.10849*.

[158] Nath, S. S., Mishra, G., Kar, J., Chakraborty, S., and Dey, N. (2014). A survey of image classification methods and techniques. In *2014 International conference on control, instrumentation, communication and computational technologies (ICCICCT)*, pages 554–557. IEEE.

[159] Oliver, A., Odena, A., Raffel, C., Cubuk, E., and Goodfellow, I. (2018). Realistic evaluation of deep semi-supervised learning algorithms. In *Advances in Neural Information Processing Systems (NeurIPS)*.

[160] Omran, M. G., Engelbrecht, A. P., and Salman, A. (2005). Differential evolution methods for unsupervised image classification. In *2005 IEEE Congress on Evolutionary Computation*, volume 2, pages 966–973. IEEE.

[161] Organization, W. H. et al. (2020). Use of chest imaging in covid-19: a rapid advice guide, 11 june 2020. Technical report, World Health Organization.

[162] Ouali, Y., Hudelot, C., and Tami, M. (2020). An overview of deep semi-supervised learning. *arXiv preprint arXiv:2006.05278*.

[163] Pal, M. (2005). Random forest classifier for remote sensing classification. *International journal of remote sensing*, 26(1):217–222.

[164] Pan, Z., Healey, G., Prasad, M., and Tromberg, B. (2003). Face recognition in hyperspectral images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 25(12):1552–1560.

[165] Paoletti, M., Haut, J., Plaza, J., and Plaza, A. (2019). Deep learning classifiers for hyperspectral imaging: A review. *ISPRS Journal of Photogrammetry and Remote Sensing*, 158:279–317.

[166] Parag, T., Plaza, S., and Scheffer, L. (2014). Small sample learning of superpixel classifiers for em segmentation. *Medical Image Computing and Computer-Assisted Intervention*, pages 389–397.

[167] Peterson, L. E. (2009). K-nearest neighbor. *Scholarpedia*, 4(2):1883.

[168] Pina, R. and Puetter, R. (1993). Bayesian image reconstruction: The pixon and optimal image modeling. *Publications of the Astronomical Society of the Pacific*, 105(688):630.

[169] Qin, Z., Yu, F., Liu, C., and Chen, X. (2018). How convolutional neural network see the world-a survey of convolutional neural network visualization methods. *arXiv preprint arXiv:1804.11191*.

[170] Raghu, M., Zhang, C., Kleinberg, J., and Bengio, S. (2019). Transfusion: Understanding transfer learning for medical imaging. In *In Advances in neural information processing systems.*, page 3347–3357.

[171] Rasmus, A., Berglund, M., Honkala, M., Valpola, H., and Raiko, T. (2015). Semi-supervised learning with ladder networks. In *Advances in neural information processing systems (NIPS)*, pages 3546–3554.

[172] Rasti, B., Hong, D., Hang, R., Ghamisi, P., Kang, X., Chanussot, J., and Benediktsson, J. A. (2020). Feature extraction for hyperspectral imagery: The evolution from shallow to deep: Overview and toolbox. *IEEE Geoscience and Remote Sensing Magazine*, 8(4):60–88.

[173] Ratle, F., Camps-Valls, G., and Weston, J. (2013). Semisupervised self-learning for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.*, 48(5):2271–2282.

[174] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*.

[175] Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99.

[176] Ren, X. and Malik, J. (2003a). Learning a classification model for segmentation. *International Conference on Computer Vision*, pages 10–17.

[177] Ren, X. and Malik, J. (2003b). Learning a classification model for segmentation. In *null*, page 10. IEEE.

[178] Rizve, M. N., Duarte, K., Rawat, Y. S., and Shah, M. (2021). In defense of pseudo-labeling: An uncertainty-aware pseudo-label selection framework for semi-supervised learning.

[179] Roberts, M., Driggs, D., Thorpe, M., Gilbey, J., Yeung, M., Ursprung, S., Aviles-Rivero, A. I., Etmann, C., McCague, C., Beer, L., et al. (2021). Common pitfalls and recommendations for using machine learning to detect and prognosticate for covid-19 using chest radiographs and ct scans. *Nature Machine Intelligence*, 3(3):199–217.

[180] Ronneberger, O., Fischer, P., and Brox, T. (2015a). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241.

[181] Ronneberger, O., Fischer, P., and Brox, T. (2015b). U-net: convolutional networks for biomedical image segmentation. *Proceedings of the Medical Image Computing and Computer-Assisted Intervention*, 9351:234–241.

[182] Ronneberger, O., Fischer, P., and Brox, T. (2016). Automated detection of pulmonary nodules in pet/ct images: ensemble false-positive reduction using a convolutional neural network technique. *Med. Phys.*, 43:2821–2827.

[183] RSNA (2019). The radiological society of north america. *[Online]: https://www.kaggle.com/c/rsna-pneumonia-detection-challenge/data*.

[184] Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

[185] Rumelhart, D. E., Durbin, R., Golden, R., and Chauvin, Y. (1995). Backpropagation: The basic theory. *Backpropagation: Theory, architectures and applications*, pages 1–34.

[186] Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., and Chen, X. (2016). Improved techniques for training gans. In *Advances in neural information processing systems (NIPS)*, pages 2234–2242.

[187] Sellars, P., Aviles-Rivero, A., Papadakis, N., Coomes, D., Faul, A., and Schönlieb, C.-B. (2019). Semi-supervised learning with graphs: Covariance based superpixels for hyperspectral image classification. *IEEE International Symposium on Geoscience and Remote Sensing (IGARSS)*.

[188] Sellars, P., Aviles-Rivero, A., and Schönlieb, C. B. (2020a). Cyclecluster: Modernising clustering regularisation for deep semi-supervised classification. *arXiv preprint arXiv:2001.05317*.

[189] Sellars, P., Aviles-Rivero, A., and Schönlieb, C. B. (2021). Laplacenet: A hybrid energy-neural model for deep semi-supervised classification. *arXiv preprint arXiv:2106.04527*.

[190] Sellars, P., Aviles-Rivero, A. I., and Schönlieb, C.-B. (2020b). Superpixel contracted graph-based learning for hyperspectral image classification. *IEEE Transactions on Geoscience and Remote Sensing*, 58(6):4180–4193.

[191] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision*, pages 618–626.

[192] Shi, W., Gong, Y., Ding, C., MaXiaoyu Tao, Z., and Zheng, N. (2018). Transductive semi-supervised deep learning using min-max features. In *European Conference on Computer Vision (ECCV)*, pages 299–315.

[193] Shih, G. and Wu, e. a. (2019). Augmenting the national institutes of health chest radiograph dataset with expert annotations of possible pneumonia. *Radiology: Artificial Intelligence*.

[194] Shorten, C., Khoshgoftaar, T., Sander, P., and Xu, X. (2019). A survey on image data augmentation for deep learning. In *Journal of Big Data 6, Article number: 60*.

[195] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[196] Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*.

[197] Sohn, K., Berthelot, D., Li, C.-L., Zhang, Z., Carlini, N., Cubuk, E. D., Kurakin, A., Zhang, H., and Raffel, C. (2020). Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*.

[198] Stutz, D., Hermans, A., and Leibe, B. (2018). Superpixels: an evaluation of the state-of-the-art". *Computer Vision and Image Understanding*, 166:1–27.

[199] Su, H., Shi, X., Cai, J., and Yang, L. (2019). Local and global consistency regularized mean teacher for semi-supervised nuclei classification. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 559–567. Springer.

[200] Su, H., Yin, Z., Kanade, T., and Zhu, J. (2016). Interactive cell segmentation based on active and semi-supervised learning. *IEEE Trans. Med. Imaging*, 35(3):762–777.

[201] Sun, Z., Wang, C., Wang, H., and Li, J. (2013). Learn multiple-kernel svms for domain adaptation in hyperspectral data. *IEEE Geosci. Remote Sens. Lett.*, 10(5):1224–1228.

[202] Sutton, R. N. and Hall, E. L. (1972). Texture measures for automatic classification of pulmonary disease. *IEEE Transactions on Computers*, pages 667–676.

[203] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *IEEE conference on computer vision and pattern recognition*, pages 2818–2826.

[204] Tan, M. and Le, Q. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114. PMLR.

[205] Tarvainen, A. and Valpola, H. (2017). Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems (NIPS)*, pages 1195–1204.

[206] Thai, L. H., Hai, T. S., and Thuy, N. T. (2012). Image classification using support vector machine and artificial neural network. *International Journal of Information Technology and Computer Science*, 4(5):32–38.

[207] Tiwari, P., Kurhanewicz, J., Rosen, M., and Madabhushi, A. (2010). Semi supervised multi kernel (sesmik) graph embedding: identifying aggressive prostate cancer via magnetic resonance imaging and spectroscopy. *Medical Image Computing and Computer-Assisted Intervention.*, pages 666–673.

[208] Toriwaki, J.-I., Suenaga, Y., Negoro, T., and Fukumura, T. (1973). Pattern recognition of chest x-ray images. *Computer Graphics and Image Processing*, pages 252–271.

[209] Tuzel, O., Porikli, F., and Meer, P. (2006). Region covariance: A fast descriptor for detection and classification. *Proc. Eur. Conf. Comput. Vis.*, pages 589–600.

[210] Uzkent, B., Hoffman, M. J., and Vodacek, A. (2016). Real-time vehicle tracking in aerial video using hyperspectral features. *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 36–44.

[211] Uzkent, B., Rangnekar, A., and Hoffman, M. J. (2017). Aerial vehicle tracking by adaptive fusion of hyperspectral likelihood maps. *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 233–242.

[212] Van Der Maaten, L., Postma, E., Van den Herik, J., et al. (2009). Dimensionality reduction: a comparative. *J Mach Learn Res*, 10(66-71):13.

[213] Van Engelen, J. E. and Hoos, H. H. (2020). A survey on semi-supervised learning. *Machine Learning*, 109(2):373–440.

[214] van Rikxoort, E., Galperin-Aizenberg, M., Goldin, J., Kockelkorn, T., van Ginneken, B., and Brown, M. (2010). Multi-classifier semi-supervised classification of tuberculosis patterns on chest ct scans. *Pulmonary Image Analysis*, pages 41–48.

[215] Vapnik, V. (2013). *The nature of statistical learning theory*. Springer science & business media.

[216] Verma, V., Lamb, A., Kannala, J., Bengio, Y., and Lopez-Paz, D. (2019). Interpolation consistency training for semi-supervised learning. *International Joint Conference on Artificial Intelligence (IJCAI)*.

[217] Vinyals, O., Blundell, C., Lillicrap, T., and Wierstra, D. e. a. (2016). Matching networks for one shot learning. NIPS.

[218] von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416.

[219] Wang, B., Liu, W., Prastawa, M., Irimia, A., Vespa, P., van Horn, J., Fletcher, P., and Gerig, G. (2014). 4d active cut: An interactive tool for pathological anatomy modeling. *International Symposium on Biomedical Imaging*, pages 529–532.

[220] Wang, F., Jiang, M., Qian, C., Yang, S., Li, C., Zhang, H., Wang, X., and Tang, X. (2017a). Residual attention network for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164.

[221] Wang, J., Jebara, T., and Chang, S.-F. (2008). Graph transduction via alternating minimization. In *International conference on Machine learning (ICML)*, pages 1144–1151. ACM.

[222] Wang, J. and Xia, Y. (2012). Fast graph construction using auction algorithm. *arXiv preprint arXiv:1210.4917*.

[223] Wang, L. and et al. (2020a). Actualmed covid-19 chest x-ray dataset initiative. *[Online] Available: https://github.com/agchung/Actualmed-COVID-chestxray-dataset*.

[224] Wang, L. and et al. (2020b). Figure 1 covid-19 chest x-ray dataset initiative. *[Online]: https://github.com/agchung/Figure1-COVID-chestxray-dataset*.

[225] Wang, L., Li, S., Chen, Y., Lin, J., Liu, C., Zeng, X., and Li, S. (2017b). Direct aneurysm volume estimation by multi-view semi-supervised manifold learning. *International Symposium on Biomedical Imaging, IEEE*, pages 1222–1225.

[226] Wang, L. and Wong, A. (2020). Covid-net: A tailored deep convolutional neural network design for detection of covid19 cases from chest radiography images. *[Online] Available: https://arxiv.org/abs/2003.09871*.

[227] Wang, T., Zhu, Z., and Blasch, E. (2010). Bio-inspired adaptive hyperspectral imaging for real-time target tracking. *IEEE Sensors Journal*, 10(3):647–654.

[228] Wang, X., Liang, G., Zhang, Y., Blanton, H., Bessinger, Z., and Jacobs, N. (2020). Inconsistent performance of deep learning models on mammogram classification. *Journal of the American College of Radiology*, 17(6):796–803.

[229] Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., and Summers, R. M. (2017c). Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2097–2106.

[230] Wayman, C. and Huneriach, N. (2019). "realising the value of health care data: a framework for the future".

[231] Wikramaratna, P., Paton, R. S., Ghafari, M., and Lourenco, J. (2020). Estimating false-negative detection rate of sars-cov-2 by rt-pcr. *medRxiv*.

[232] Williams, C. K. I. and Seeger, M. (2001). Using the nyström method to speed up kernel machines. *Proc. Neural Information Processing Systems*.

[233] Wold, S., Esbensen, K., and Geladi, P. (1987). Principal component analysis. *Chemometrics and intelligent laboratory systems*, 2(1-3):37–52.

[234] Wolterink, J., Leiner, T., de Vos, B., van Hamersvelt, R., Viergever, M., and Isgum, I. (2016). Automatic coronary artery calcium scoring in cardiac ct angiography using paired convolutional neural networks. *Med. Image Anal.*, 34:123–136.

[235] Wong, H. Y. F. and et al. (2020). Frequency and distribution of chest radiographic findings in covid-19 positive patients. *Radiology*, page 201160.

[236] Xie, Q., Dai, Z., Hovy, E., Luong, M.-T., and Le, Q. V. (2020a). Unsupervised data augmentation for consistency training. *Neural Information Processing Systems*.

[237] Xie, X., Zhong, Z., Zhao, W., Zheng, C., Wang, F., and Liu, J. (2020b). Chest ct for typical 2019-ncov pneumonia: relationship to negative rt-pcr testing. *Radiology*, page 200343.

[238] Xie, Y., Zhang, Z., Sapkota, M., and Yang, L. (2016). Spatial clockwork recurrent neural network for muscle perimysium segmentation. *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, 9901:185–193.

[239] Yang, F. and Jiang, T. (2003). Pixon-based image segmentation with markov random fields. *IEEE Transactions on Image Processing*, 12(12):1552–1559.

[240] Yao, L., Prosky, J., Poblenz, E., Covington, B., and Lyman, K. (2018). Weakly supervised medical diagnosis and localization from multiple resolutions. *arXiv preprint arXiv:1803.07703*.

[241] Zagoruyko, S. and Komodakis, N. (2016). Wide residual networks. In *Proceedings of the British Machine Vision Conference*.

[242] Zehtabian, A. and Ghassemian, H. (2014). A pixon-based hyperspectral image segmentation method used for remote sensing data classification. In *7'th International Symposium on Telecommunications (IST'2014)*, pages 436–440. IEEE.

[243] Zhan, Y., Hu, D., Wang, Y., and Yu, X. (2018). Semisupervised hyperspectral image classification based on generative adversarial networks. *IEEE Trans. Geosci. Remote Sens. Letters*, 15(2):212–216.

[244] Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2021). Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM*, 64(3):107–115.

[245] Zhang, H., Cisse, M., Dauphin, Y. N., and Lopez-Paz, D. (2017a). mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

[246] Zhang, Y., Du, B., Zhang, L., and Liu, T. (2017b). Joint sparse representation and multitask learning for hyperspectral target detection. *IEEE Transactions on Geoscience and Remote Sensing*, 55(2):894–906.

[247] Zhao, W. and Du, S. (2016). Spectral–spatial feature extraction for hyperspectral image classification: A dimension reduction and deep learning approach. *IEEE Trans. Geosci. Remote Sens.*, 54(8):4544–4554.

[248] Zhou, D., Bousquet, O., Lal, T., Weston, J., and Schölkopf, B. (2004a). Learning with local and global consistency. *NIPS*, pages 595–602.

[249] Zhou, D., Bousquet, O., Lal, T. N., Weston, J., and Schölkopf, B. (2004b). Learning with local and global consistency. In *Advances in Neural Information Processing Systems (NIPS)*, pages 321–328.

[250] Zhou, S., Wang, Y., Zhu, T., and Xia, L. (2020). Ct features of coronavirus disease 2019 (covid-19) pneumonia in 62 patients in wuhan, china. *American Journal of Roentgenology*, pages 1–8.

[251] Zhu, L., Chen, Y., Ghamisi, P., and Benediktsson, J. A. (2018). Generative adversarial networks for hyperspectral image classification. *IEEE Trans. Geosci. Remote Sens.*, 56(9):5046–5063.

[252] Zhu, W., Chayes, V., Tiard, A., Sanchez, S., Dahlberg, D., Bertozzi, A. L., Osher, S., Zosso, D., and Kuang, D. (2017a). Unsupervised classification in hyperspectral imagery with nonlocal total variation and primal-dual hybrid gradient algorithm. *IEEE Transactions on Geoscience and Remote Sensing*, 55(5):2786–2798.

[253] Zhu, X., Ghahramani, Z., and Lafferty, J. D. (2003a). Semi-supervised learning using gaussian fields and harmonic functions. In *International conference on Machine learning (ICML'03)*, pages 912–919.

[254] Zhu, X., Ghahramani, Z., and Lafferty, J. D. (2003b). Semi-supervised learning using gaussian fields and harmonic functions. In *P International conference on Machine learning (ICML)*, pages 912–919.

[255] Zhu, X. and Lafferty, J. (2005). Harmonic mixtures: combining mixture models and graph-based methods for inductive and scalable semi-supervised learning. In *International conference on Machine Learning (ICML)*, pages 1052–1059.

[256] Zhu, Z. et al. (2017b). Unsupervised classification in hyperspectral imagery with non-local total variation and primal-dual hybrid gradient algorithm. *IEEE Trans. Geosci. Remote Sens.*, 55(5):2786–2798.