

**AI-ENHANCED MAMMOGRAPHY SCREENING: ELEVATING EARLY
BREAST CANCER DETECTION**

BY

Tharini Vijesh Kumar

A REPORT
SUBMITTED TO
Universiti Tunku Abdul Rahman
in partial fulfillment of the requirements

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONOURS)
Faculty of Information and Communication Technology
(Kampar Campus)

JUNE 2025

COPYRIGHT STATEMENT

© 2025 Tharini Vijesh Kumar. All rights reserved.

This Final Year Project report is submitted in partial fulfillment of the requirements for the degree of Bachelor of Computer Science (Honours) Information and Communication Technology at Universiti Tunku Abdul Rahman (UTAR). This Final Year Project report represents the work of the author, except where due acknowledgment has been made in the text. No part of this Final Year Project report may be reproduced, stored, or transmitted in any form or by any means, whether electronic, mechanical, photocopying, recording, or otherwise, without the prior written permission of the author or UTAR, in accordance with UTAR's Intellectual Property Policy.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to all these individuals who have mentored and supported me throughout this journey. First and foremost, I extend my deepest appreciation to my supervisor, Mrs. Nur Lyana Binti Albasahah, for equipping me with invaluable guidance and patience since day one. Her deep insights into AI and medical imaging have been instrumental in shaping this project into what it has become today.

To my aunt and grandma, who stood by me through my highs and lows, being there, believing in me even when I had no reason to believe in myself. Thank you for being the anchor of my life and raising me with kindness and warmth that will be with me for the rest of my life.

This achievement is as much yours as it is mine.

ABSTRACT

Breast cancer detection from mammography images is challenging due to limited annotated datasets and, along with the high similarity between classes, leads to deep learning models overfitting. This study explored Prototypical Contrastive Learning (PCL), a self-supervised learning approach, to address these challenges by learning meaningful representations with minimal reliance on labels. The original PCL framework was adapted for mammography by enabling single-GPU training, integrating multiple backbones (ResNet-34, ResNet-50, MobileNet, and DenseNet-121), and handling greyscale images. Both the original and cropped datasets were used to assess the effect of removing irrelevant background, while more experiments were run with varying batch sizes, learning rates, and temperature parameters. Results showed consistent improvements with cropping and hyperparameter tuning. DenseNet-121 achieved the best performance, with 92.3% accuracy, 92.5% precision, and 96.2% specificity, while MobileNet provided competitive results with 66.7% accuracy and 69.6% precision on the cropped dataset. The findings shed a light onto PCL framework's potential as a viable complement to supervised methods in low-data medical imaging tasks.

Area of Study: Medical Image Analysis, Artificial Intelligence in Healthcare

Keywords: Breast Cancer Detection, Mammography, Self-Supervised Learning,

Prototypical Contrastive Learning, Backbones

TABLE OF CONTENTS

COPYRIGHT STATEMENT.....	I
ACKNOWLEDGEMENTS.....	I I
ABSTRACT.....	I I I
TABLE OF CONTENTS.....	IV
LIST OF FIGURES.....	VI
LIST OF TABLES.....	VIII
LIST OF ABBREVIATIONS.....	IX
CHAPTER 1 Introduction	
1.1 Project Background.....	1
1.2 Problem Statement and Motivation.....	2
1.3 Objectives.....	3
1.4 Project Scope and Direction.....	3
1.5 Contributions.....	4
1.6 Report Organization.....	4
CHAPTER 2 Literature Review	
2.1 Previous Works on Deep Learning.....	6
2.2 Limitations of Previous Studies.....	20
2.3 Proposed Solutions.....	24
CHAPTER 3 System Model	
3.1 System Design Diagram.....	30
3.2 Implementation.....	33
3.3 System Operation.....	43
3.4 Implementation Issues.....	49
3.5 Concluding Remark.....	50
CHAPTER 4 Results	
4.1 Introduction.....	51
4.2 Results for ResNet-50.....	51
4.3 Results for ResNet-34.....	55
4.4 Results for MobileNet.....	59
4.5 Results for DenseNet-121.....	63
4.6 Summary of Results.....	67
Chapter 5 Discussion	
5.1 Introduction.....	69
5.2 Backbone-Specific Discussions.....	69
5.3 Effect of Dataset Variation.....	72
5.4 Effect of Hyperparameter Tuning.....	73
5.5 Strengths and Limitations of the Study.....	74
5.6 Concluding Remarks.....	75

CHAPTER 6 Conclusion and Future Work

6.1 Conclusion.....	76
6.2 Future Work.....	77
REFERENCES.....	78
APPENDIX.....	84
A.1 Poster.....	84

LIST OF FIGURES

Figure Number	Title	Page
<i>Figure 2.1</i>	The "MetaMed" approach's general framework, as presented by Singh et al.	7
<i>Figure 2.2</i>	Pipeline of the Proposed Method by Jiang et al.	8
<i>Figure 2.3</i>	The general structure of Z. Dai et al.'s suggested method, PFEMed	8
<i>Figure 2.4</i>	N-way K-shot paradigm representation by E. Pachetti et al.	8
<i>Figure 2.5</i>	The flowchart of R. Gong et al.'s task-driven, self-supervised bi-channel networks	9
<i>Figure 2.6</i>	An overview of the few-shot pathology image classification process by H. Quan et al.	9
<i>Figure 2.7</i>	The framework of the few-shot classification method used by Chen et al.	10
<i>Figure 2.8</i>	The Proposed Approach of Eva Pachetti et al.	11
<i>Figure 2.9</i>	X. Yue et al.'s method	11
<i>Figure 2.10</i>	The Overall Architecture of Chen et al.'s Proposed Approach	12
<i>Figure 2.11</i>	Medina et al.'s Self-Supervised Prototypical Transfer Learning	12
<i>Figure 2.12</i>	Overview of Koch et al.'s Proposed Method	13
<i>Figure 2.13</i>	Overview of MABEL and its integration with clinical PACS	14
<i>Figure 2.14</i>	Few-shot Classification of Ultrasound Breast Cancer Images Using Meta-Learning Algorithms by Işık and Paçal	15
<i>Figure 2.15</i>	Prototypical Neural Network Architecture by Apeksha et al.	15
<i>Figure 2.16</i>	Research Flow for Sarvinda et al.'s Proposed Method	17
<i>Figure 2.17</i>	Wang et al.'s Flowchart of Preprocessing	18
<i>Figure 2.18</i>	Umesh Kumar's U-Net Flow Chart	18
<i>Figure 2.19</i>	Architecture of Dentret as proposed by Musa Adamu et al.	19
<i>Figure 2.20</i>	The DenseNet Architecture by R. K. Pattanaik	20
<i>Figure 3.1</i>	System Architecture Diagram	30
<i>Figure 3.2</i>	Example of Mammography Images Before and After Cropping	36

<i>Figure 3.3</i>	Screenshot of Google Colab Folder showing the subdirectories of the dataset used	43
<i>Figure 3.4</i>	Screenshot of removing the excess background of a benign image for the cropped dataset using ImageJ	44
<i>Figure 3.5</i>	Screenshot of Colab Environment showing the GPU (T4) allocation	45
<i>Figure 3.6</i>	Screenshot of Epoch 60 of ResNet-34	45
<i>Figure 3.7</i>	Screenshot of checkpoints saved in Google Drive	46
<i>Figure 3.8</i>	Screenshot of Confusion Matrix for MobileNet	47
<i>Figure 3.9</i>	Screenshot of post-training embedding visualization	47
<i>Figure 4.1</i>	t-SNE visualisation of the best-performing ResNet-50 configuration (cropped dataset, batch size = 32, LR = 0.0005, T = 0.1)	53
<i>Figure 4.2</i>	UMAP visualisation of the best-performing ResNet-50 configuration (cropped dataset, batch size = 32, LR = 0.0005, T = 0.1)	54
<i>Figure 4.3</i>	t-SNE visualisation of the best-performing ResNet-34 configuration (cropped dataset, batch size = 32, LR = 0.0005, T = 0.1)	57
<i>Figure 4.4</i>	UMAP visualisation of the best-performing ResNet-34 configuration (cropped dataset, batch size = 32, LR = 0.0005, T = 0.1)	58
<i>Figure 4.5</i>	t-SNE visualisation of the best-performing MobileNet configuration (cropped dataset, batch size = 32, LR = 0.0005, T = 0.1)	61
<i>Figure 4.6</i>	UMAP visualisation of the best-performing MobileNet configuration (cropped dataset, batch size = 32, LR = 0.0005, T = 0.1)	62
<i>Figure 4.7</i>	t-SNE visualisation of the best-performing DenseNet-121 configuration (cropped dataset, batch size = 32, LR = 0.0005, T = 0.1)	65
<i>Figure 4.8</i>	UMAP visualisation of the best-performing DenseNet-121 configuration (cropped dataset, batch size = 32, LR = 0.0005, T = 0.1)	66

LIST OF TABLES

<i>Table Number</i>	<i>Title</i>	<i>Page</i>
<i>Table 2.1</i>	Proposed Solutions of Previous Works	24
<i>Table 3.1</i>	Specifications of Laptop	33
<i>Table 3.2</i>	Example of Experimental Runs (ResNet-50 backbone)	41
<i>Table 4.1</i>	ResNet-50 Default Run (Original Dataset)	51
<i>Table 4.2</i>	ResNet-50 Dataset Variation (Cropped Dataset)	52
<i>Table 4.3</i>	ResNet-50 Batch Size Variation	52
<i>Table 4.4</i>	ResNet-50 Learning Rate Variation	52
<i>Table 4.5</i>	ResNet-50 Temperature Variation	53
<i>Table 4.6</i>	ResNet-34 Default Run (Original Dataset)	55
<i>Table 4.7</i>	ResNet-34 Dataset Variation (Cropped Dataset)	55
<i>Table 4.8</i>	ResNet-34 Batch Size Variation	56
<i>Table 4.9</i>	ResNet-34 Learning Rate Variation	56
<i>Table 4.10</i>	ResNet-34 Temperature Variation	56
<i>Table 4.11</i>	MobileNet Default Run (Original Dataset)	59
<i>Table 4.12</i>	MobileNet Dataset Variation (Cropped Dataset)	59
<i>Table 4.13</i>	MobileNet Batch Size Variation	60
<i>Table 4.14</i>	MobileNet Learning Rate Variation	60
<i>Table 4.15</i>	MobileNet Temperature Variation	60
<i>Table 4.16</i>	DenseNet-121 Default Run (Original Dataset)	63
<i>Table 4.17</i>	DenseNet-121 Dataset Variation (Cropped Dataset)	63
<i>Table 4.18</i>	DenseNet-121 Batch Size Variation	63
<i>Table 4.19</i>	DenseNet-121 Learning Rate Variation	64
<i>Table 4.20</i>	DenseNet-121 Temperature Variation	64
<i>Table 4.21</i>	Consolidated Comparison Table	67

LIST OF ABBREVIATIONS

<i>AI</i>	Artificial Intelligence
<i>CADe</i>	Computer-Aided Detection
<i>CADx</i>	Computer Aided Diagnosis
<i>DCPN</i>	Dual-Channel Prototype Network
<i>DDP</i>	Distributed Data Parallel
<i>DL</i>	Deep Learning
<i>Faiss</i>	Facebook AI Similarity Search
<i>FSL</i>	Few-Shot Learning
<i>FUDA</i>	Few-Shot Unsupervised Domain Adaptation
<i>GAN</i>	Generative Adversarial Networks
<i>GDSL</i>	Gaussian Disturbance Soft Label
<i>IP-IRM</i>	Iterative Partition-based Invariant Risk Minimization
<i>PCS</i>	Prototypical Cross-domain Self-Supervised Learning
<i>PCL</i>	Prototypical Contrastive Learning
<i>ProtoNet</i>	Prototypical Netwrks
<i>PVT</i>	Pyramid Vision Transformer
<i>SSL</i>	Self-supervised learning
<i>TSBN</i>	Task-driven, self-supervised bi-channel networks
<i>VAE</i>	Variational Autoencoder

CHAPTER 1

Introduction

In this chapter, we present the project background, define the problem statement and motivation, and outline the objectives along with the scope and direction of the study. The chapter also underlines the key contributions of this research and concludes with the overall report organization.

1.1 Project Background

Breast cancer has always been and still unfortunately is a major global health concern; thus, increasing survival rates requires early detection. Over 410,000 women (out of the over a million who do get diagnosed) succumb to breast cancer each year [21]. It is widely agreed that this is due to the infrastructure and resources not being easily available or attainable by low- and middle-income countries, and these countries thus lack routine mammography screening. Both of these aspects exacerbate the numbers of late-stage diagnoses [21] [22]. In resource-constrained settings where traditional screening methods might not be sufficient, the project aims to study a deep learning-based approach for breast cancer detection using mammography images. DL models have shown great promise in medical image analysis; however, their performance typically depends on the availability of large, labeled datasets, which are often scarce in the medical domain. This scarcity is then further compounded by the fact that mammography images, especially across benign, malignant, and normal cases, often exhibit very subtle visual differences, making classification a task that requires one to do a lot of research and training.

To address these challenges, this research explores the use of Prototypical Contrastive Learning (PCL), a self-supervised learning (SSL) method that has achieved success on datasets with clear inter-class differences but remains underexplored in domains with high intra-class similarity, such as breast cancer imaging.

1.2 Problem Statement and Motivation

Although considerable progress has been made in the development of cancer screening technologies, breast cancer remains one of the most severe causes of death among females all over the world, including in developing countries. Among the main reasons for high mortality is the difficulty in detecting the disease, particularly in patients with dense breast tissue [24]. Tumours are masked in mammograms by dense tissue, making them more difficult to observe during regular screening.

PCL has shown promising results in representation learning for low-data scenarios, but its application remains relatively new and heavily unexplored, especially within the field of medical imaging. Existing PCL research has largely focused on datasets with highly distinct classes, whereas breast cancer mammography images present high intra-class similarity, making feature discrimination considerably more challenging. Furthermore, studies applying PCL to backbones other than ResNet are scarce, leaving a gap in understanding how alternative architectures such as MobileNet might perform in this context. This lack of exploration highlights the need to investigate PCL’s potential for fine-grained, resource-constrained medical image classification using diverse backbone networks.

The core motivation behind this study stems from the ability of human intelligence to learn quickly from limited samples and that of which applies the prior knowledge to solve new, complex problems. In medical diagnosis, especially in the detection of conditions such as breast cancer, the limited availability of data presents itself as a significant challenge. Additionally, data collection in such scenarios is costly and time-consuming, making it difficult to train under conventional deep-learning techniques, which rely on innumerable data points to produce an accurate classification. While DL and machine learning models have shown promise in medical image analysis, they need help with generalisation when trained on limited data, often leading to biased models and unreliable results.

Researchers have explored other methods, such as transfer learning and fine-tuning techniques, to address the aforementioned challenges. Once again, these methodologies

depend on the assumption that the pre-trained models inherit their datasets from a distribution similar to that of the target medical data. This condition is rarely achieved. Active research areas, such as synthetic data generation through augmentation or the use of GANs, have been explored to address this issue. However, GANS come with its own set of complexities, particularly extreme parameter optimisation. Although these methodologies are certainly helpful, they do not however fully address the challenges posed by the limited data in medical imaging.

This gap presents an opportunity to explore alternative SSL techniques that remain underexamined in this field. In particular, PCL offers a promising yet largely unexplored direction—especially for fine-grained medical imaging tasks—where its potential to learn discriminative representations from limited, high intra-class similarity data has not been fully investigated. By pursuing this line of research, we can contribute to expanding the current understanding of SSL’s role in advancing medical image classification under resource-constrained conditions.

1.3 Objectives

The primary objective of this research is to study the unexplored areas of PCL for breast cancer image classification, which is known for its data scarcity. Specifically, the study aims to implement a PCL to address the scarcity of annotated medical data, evaluate the performance of multiple backbone architectures—ResNet-34, ResNet-50, MobileNet and DenseNet-121—within this framework to extend exploration, assess PCL’s ability to handle the high intra-class similarity characteristic of breast cancer mammography images compared to datasets with more distinct classes, and analyze the broader potential of PCL as a complementary approach to traditional supervised learning in fine-grained medical imaging tasks.

1.4 Project Scope and Direction

The direction of this research is exploratory in nature. This research aims to bridge the unexplored gap in the existing literature on PCL, particularly in a high-intra class similarity scenario, which is what mammograms are. By testing multiple backbone

architectures, different parameter configurations, and preprocessing strategies, this study seeks to gain further insights on PCL and how it can be adapted to such resource-constrained applications. Although this research is focused on breast cancer detection, the findings produced are intended to contribute to broadening and widening the understanding of how SSL techniques, particularly PCL, can be extended to other low-data areas of study.

1.5 Contributions

This research contributes to the limited body of work available on the PCL framework being used for medical imaging, specifically breast cancer detection using mammography images. It explores the effectiveness of PCL in this resource-constrained environment, such as the mammograms where intra-class similarity is very low. This scenario (low intra class similarity) is rarely addressed in prior studies, and the studies that do exist only use ResNet as their backbone. Thus, this research shines a light into the unclear waters to study PCL with multiple backbone architectures, which includes ResNet-34, ResNet-50, MobileNet, and DenseNet-121, extending beyond the commonly used ResNet-50 model. The study also investigates the impact of backbone parameter configurations on the results of classification and presents best practices for fine-grained, low-data medical image classification. It also delves into the impact of background subtraction through cropping images and presents practical recommendations concerning preprocessing methods in mammography analysis. Overall, this work informs the knowledge on how SSL, and PCL in particular, can be applied to fine- and resource-poor environments.

1.6 Report Organization

This report is organised into 6 chapters: Chapter 1 Introduction, Chapter 2 Literature Review, Chapter 3 System Design, Chapter 4 Results, Chapter 5 Discussion, and Chapter 6 Conclusion and Future Works. The first chapter is the introduction of this project, which includes problem statement, project background and motivation, project scope, project objectives, project contribution, highlights of project achievements, and report organisation. The second chapter is the literature review carried out on several existing

studies on medical imaging, contrastive learning, transfer learning, and backbones, then evaluating the strengths and weaknesses of each of the studies. The third chapter is discussing the model of the experiments and how it was implemented in detail. The fourth chapter is regarding the results obtained from the experiments of each of the backbones. Furthermore, the fifth chapter is a deep dive into the discussion of the results, reasoning how the results were obtained and why, and comparing the results with the existing studies. Lastly, the final chapter includes the conclusion and the future work to be done.

CHAPTER 2

Literature Review

2.1 Previous Works on Deep Learning

2.1.1 Application of Few-Shot Learning for Medical Image Classification

In medical image classification, FSL (FSL) is becoming increasingly important, especially in cases where data is limited, like in rare disease categories. Singh et al. 's [13] "MetaMed" technique, which uses the Reptile algorithm for gradient-based meta-learning, fine-tunes models across tasks with few examples, achieving over 70% accuracy on datasets like Pap Smear, and ISIC 2018. Jiang et al. [17] expanded on FSL by proposing a multi-learner model that brings together meta-learning, transfer learning, and metric learning. Their method showed promising results on datasets like BLOOD and CHEST. In a similar direction, Z. Dai et al. [14] introduced PFEMed, which improved classification accuracy by using a dual-encoder structure alongside a prior-guided Variational Autoencoder (VAE), with good performance, especially on the Pap smear dataset. These advancements show how FSL can overcome the usual limitations of traditional methods and push forward the field of medical image classification. In addition, E. Pachetti and S. Colantino's work [3] showed that meta-learning methods like ProtoNets can deliver reliable performance even when only small amounts of medical imaging data are available.

CHAPTER 2 Literature Review

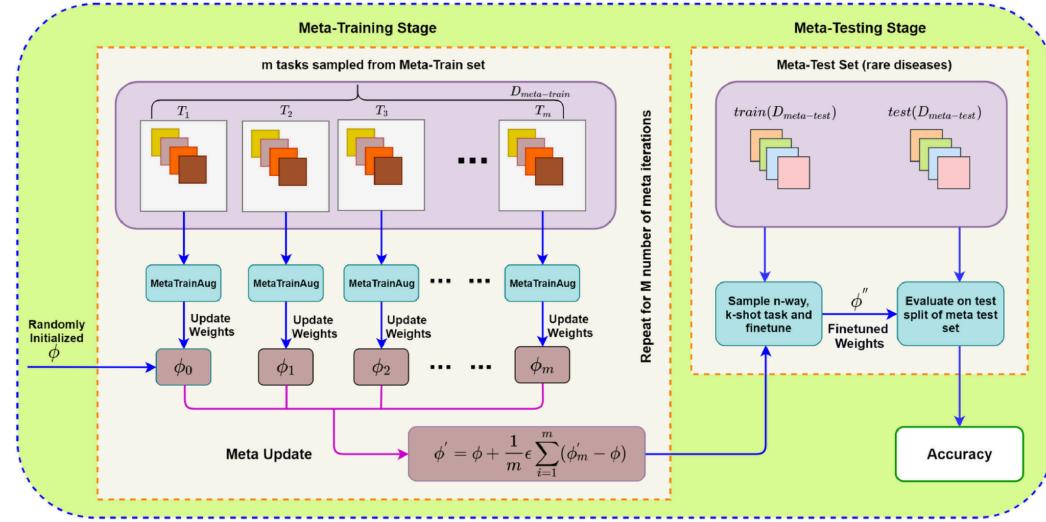


Figure 2.1: The "MetaMed" approach's general framework, as presented by Singh et al.

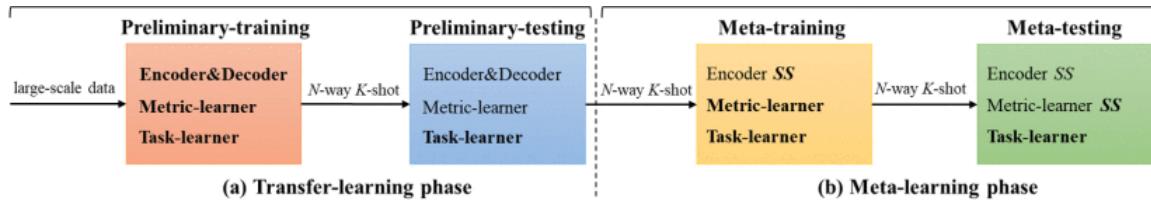


Figure 2.2: Pipeline of the Proposed Method by Jiang et al.

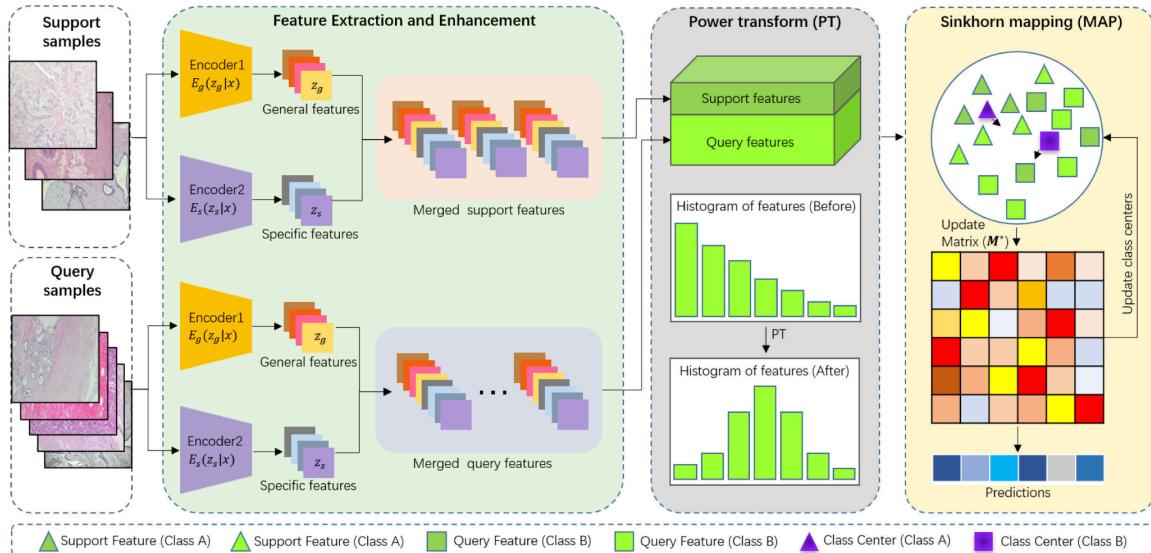


Figure 2.3: The general structure of Z. Dai et al.'s suggested method, PFEMed

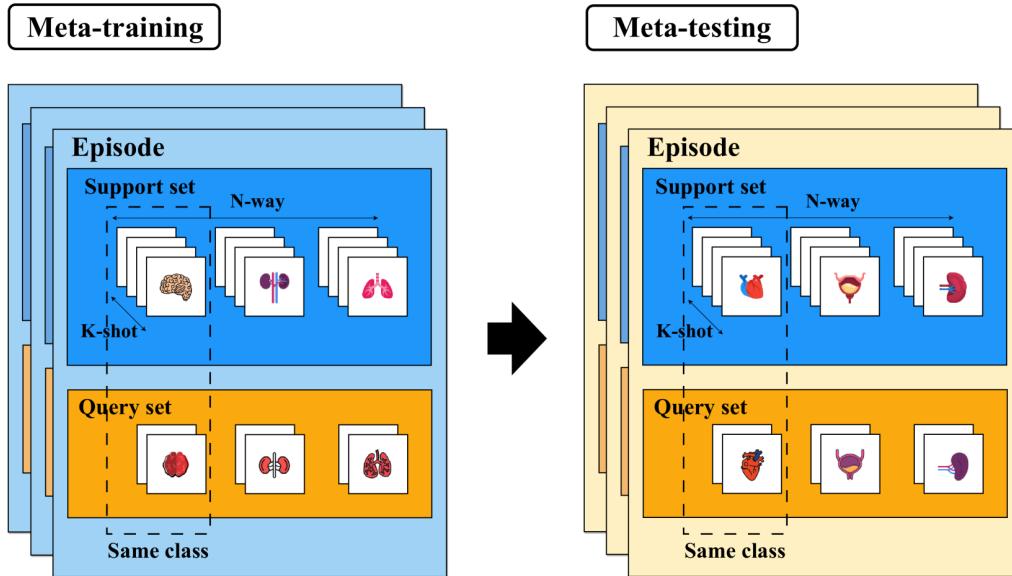


Figure 2.4: *N-way K-shot paradigm representation by E. Pachetti et al.*

2.1.2 SSL as a Pretraining Step

Advances in DL in recent times have brought to light the significance of SSL as a practical pre-training step, especially in situations where labelled data is scarce. As exemplified in various studies, SSL techniques enable models to learn robust feature representations by solving pretext tasks using unlabeled data. Using greyscale image mapping (GSIM) to embed class label information directly into the image restoration process, R. Gong et al. [7] introduced a task-driven self-supervised bi-channel network (TSBN) framework that improves the discriminative power of learned features for breast cancer diagnosis. Comparably, H. Quan et al.'s Dual-Channel Prototype Network (DCPN) [10] combines SSL with multi-scale feature extraction to achieve notable accuracy gains in few-shot pathology image classification tasks. Chen et al. [5] employ an unsupervised learning-based two-step approach to detect tumour metastases. They rank cells in unlabeled patches using unsupervised cell ranking on hematoxylin, and eosin (H&E)-stained whole-slide images (WSIs). To increase accuracy, and manage a limited number of labelled samples, they then fine-tune a classification network with minimal labelled data.

CHAPTER 2 Literature Review

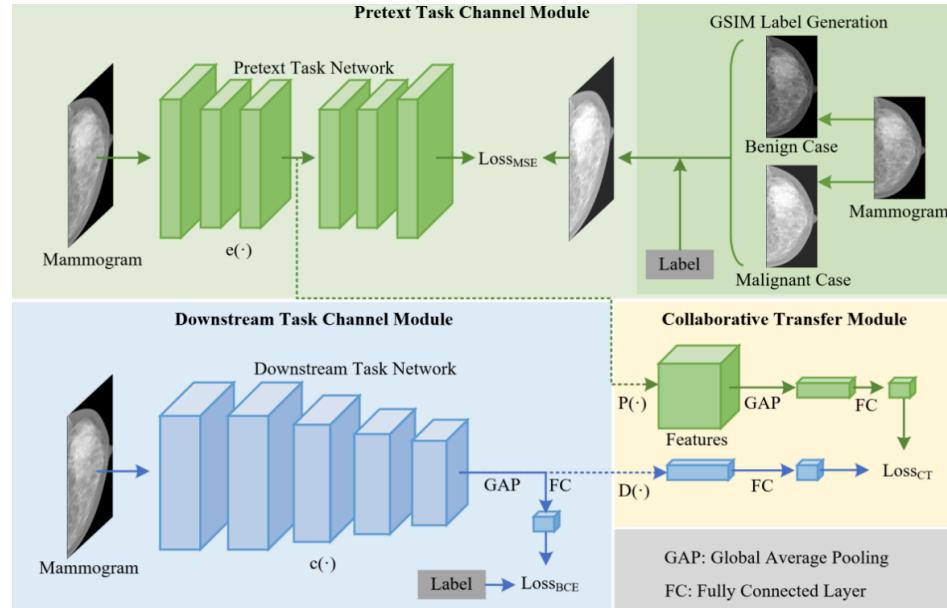


Figure 2.5: The flowchart of R. Gong et al.'s task-driven, self-supervised bi-channel networks

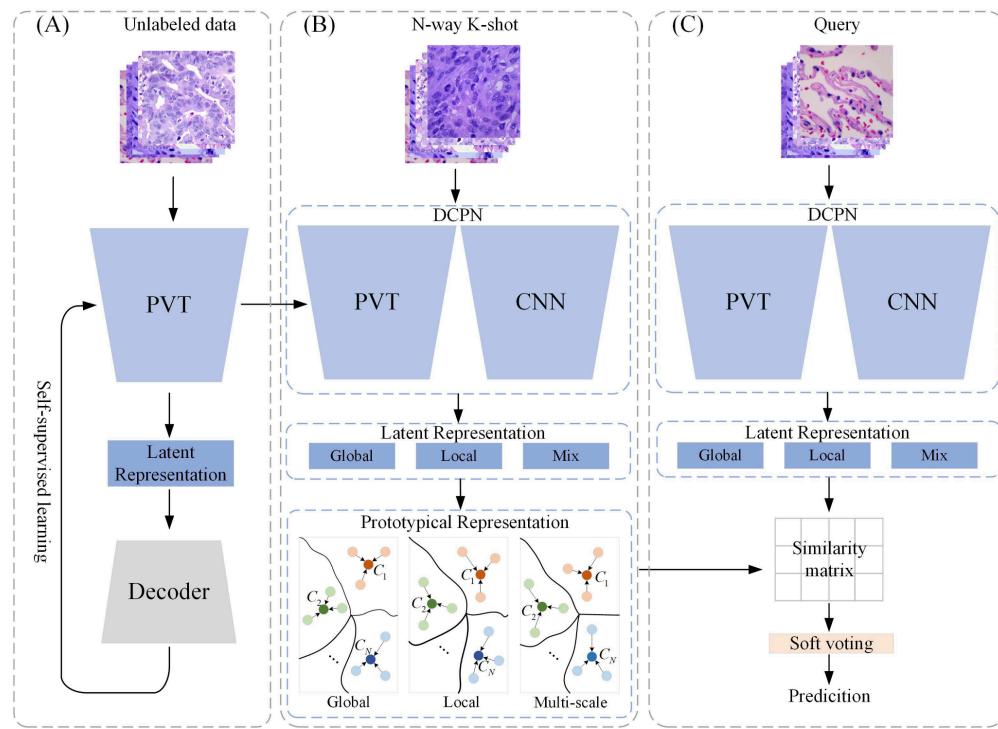


Figure 2.6: An overview of the few-shot pathology image classification process by H. Quan et al. using the DCPN method

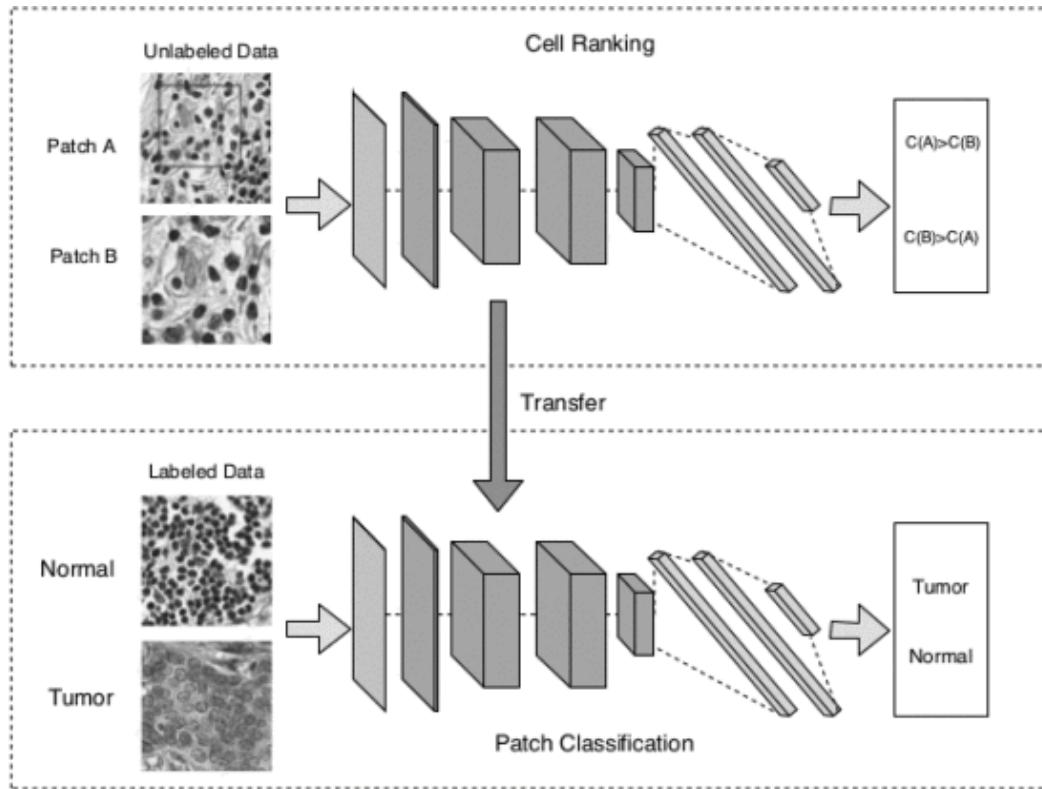


Figure 2.7: The framework of the few-shot classification method used by Chen et al.

Furthermore, disentangled SSL, and meta-learning were combined by Eva Pachetti et al. [12] to improve model performance in medical image classification in situations with limited data. Their method, which included using the IP-IRM algorithm for pre-training, and then meta-fine-tuning, showed resilience to different distribution shifts. Using PCL within the Prototypical Cross-domain SSL (PCS) framework, Xiangyu Y. et al. [15] improved few-shot unsupervised domain adaptation. They successfully aligned the semantic structures beyond domain boundaries. Meanwhile, by combining contrastive SSL with episodic training, Chen et al. [18] achieved state-of-the-art results in few-shot classification across cross-domain scenarios, further validating the efficacy of SSL as a pre-training step. ProtoTransfer, first presented by Medina et al., clusters unlabeled prototypical samples, and uses SSL to produce metric solid embeddings. This method works very well for few-shot classification tasks because it greatly reduces the dependence on annotated data, and outperforms many unsupervised meta-learning techniques [19]. Using Siamese networks, Koch et al. [20] similarly showed the efficacy

CHAPTER 2 Literature Review

of SSL-inspired methods for one-shot learning, which well generalises to new classes with little data, underscoring SSL's potential for creating compelling, adaptable models. Together, these studies highlight how SSL can dramatically improve FSL models' performance, especially when labelled data is hard to come by.

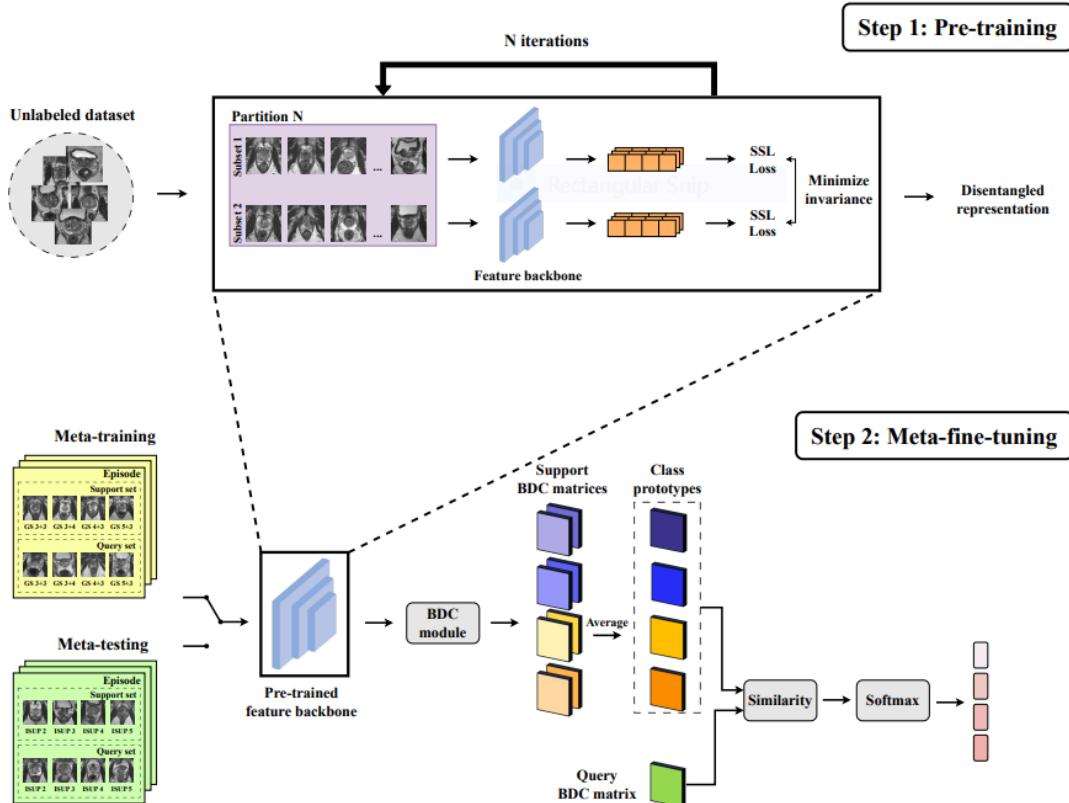


Figure 2.8: The Proposed Approach of Eva Pachetti et al.

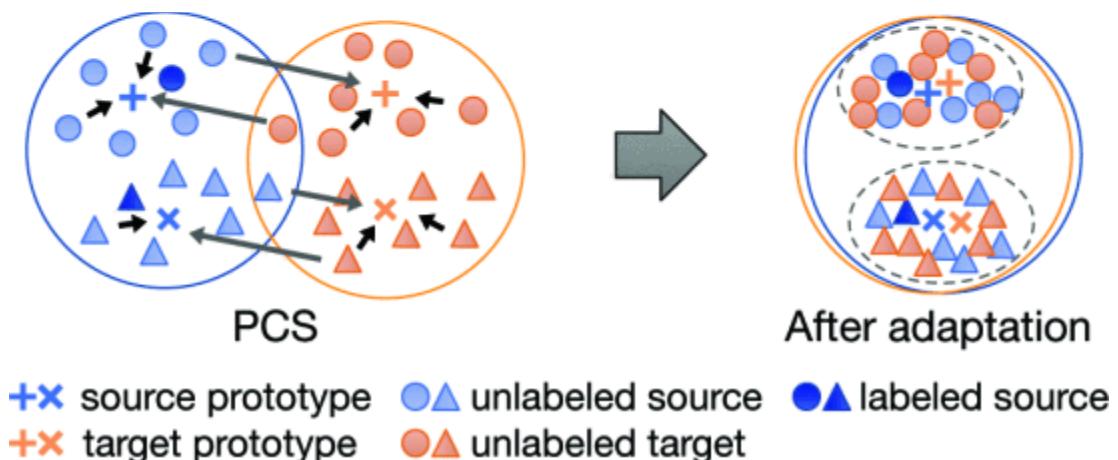


Figure 2.9: The method proposed by X. Yue et al.

CHAPTER 2 Literature Review

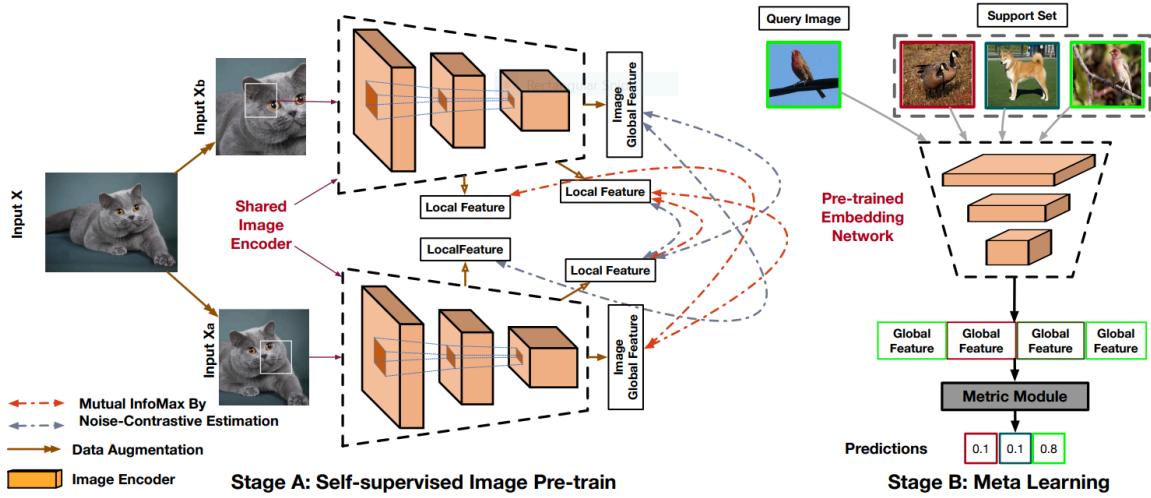


Figure 2.10: The Overall Architecture of Chen et al.’s Proposed Approach

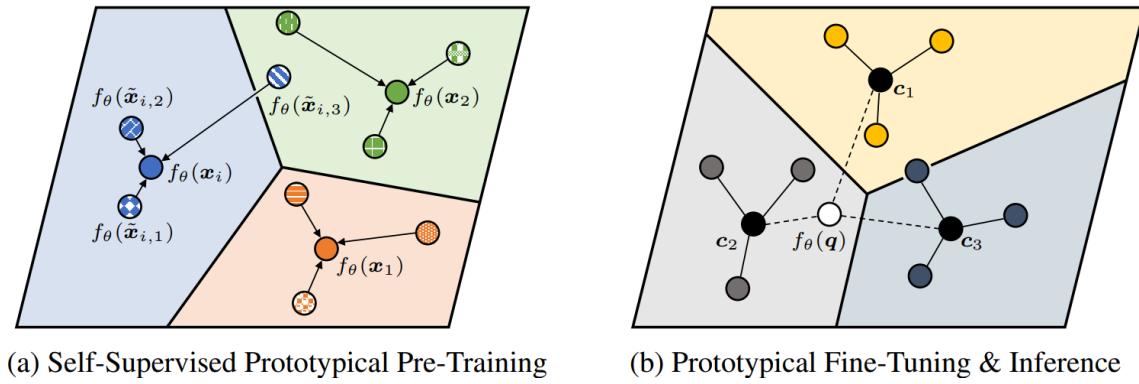


Figure 2.11: Medina et al.’s method clusters images with augmentations near class prototypes, then fine-tunes a linear layer for classification

CHAPTER 2 Literature Review

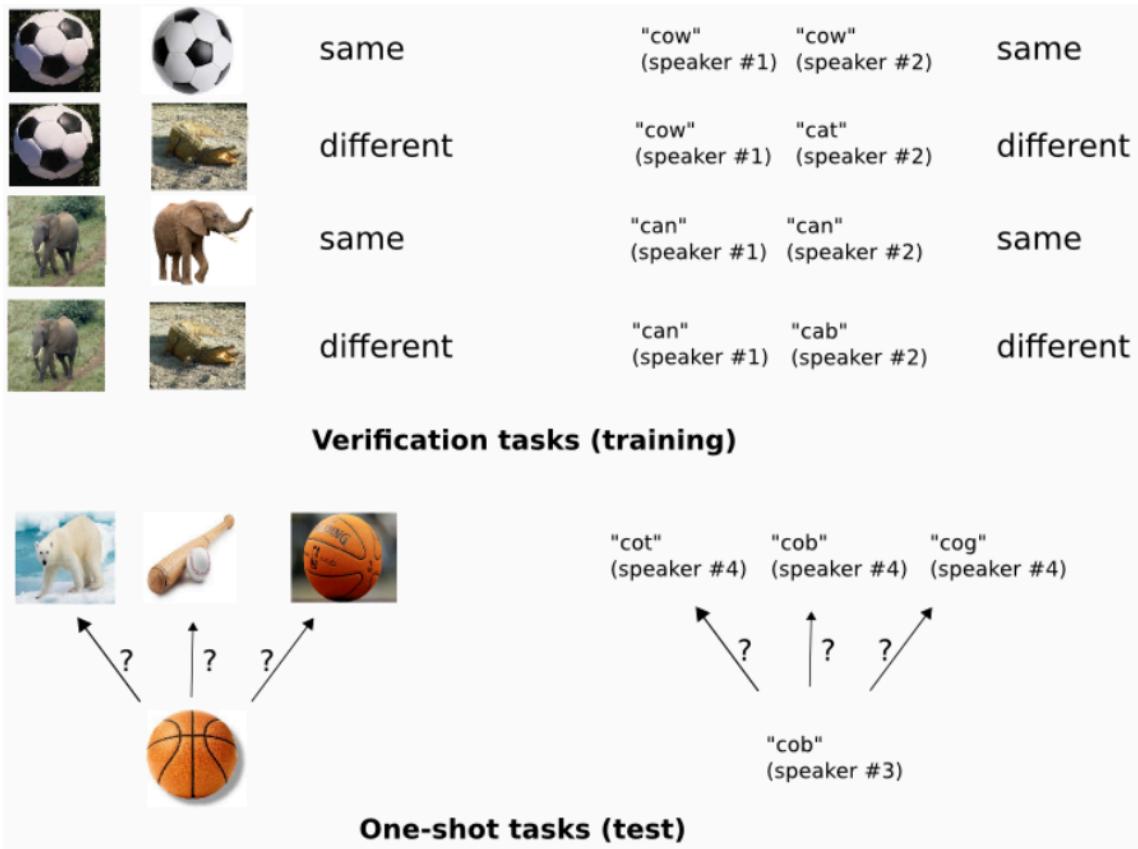


Figure 2.12 Overview of Koch et al.'s Proposed Method

2.1.3 AI Integration in Diagnostic Systems

Z. Yang et al. [1] explored the U-Net architecture for breast cancer detection, demonstrating how accurate the biomedical image segmentation can be in helping to improve diagnostic systems. The symmetric encoder-decoder structure of U-Net that they used was found to be very effective at segmenting complex mammographic lesions, which helped improve the accuracy and effectiveness of their CADx and CADe systems. In another study done by N. Berberian et al. [2], they researched and developed a portable thermal imaging tool that was powered by AI. Their approach offered a more accessible and affordable option for breast cancer imaging compared to traditional methods by detecting temperature changes in breast tissue. The AI-based thermal imaging system achieved a sensitivity of 89% and a specificity of 83%, performing as well as or even better than mammography, especially in areas where standard screening tools are harder

CHAPTER 2 Literature Review

to access. Overall, these studies show the growing potential of using AI to improve both accessibility and accuracy in breast cancer detection.

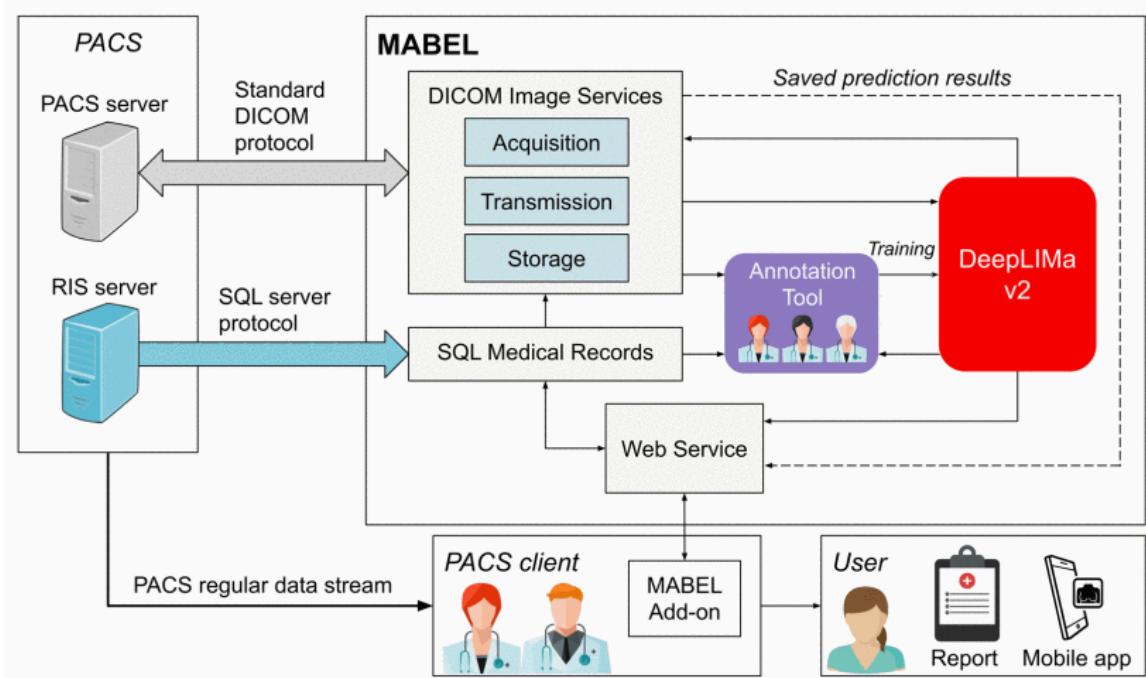


Figure 2.13: Overview of MABEL and its integration with clinical PACS as proposed by Z. Yang et al.

2.1.4 Prototypical Networks, and Transfer Learning in Few-Shot Learning

Prototypical networks (ProtoNet) have become a popular choice in FSL tasks because they can work well with limited annotated data. For example, Işık and Paçal [4] tested ProtoNet on ultrasound breast cancer images, using a ResNet50 backbone, and saw accuracy improvements. In a 10-shot learning setup, they achieved an accuracy of 88.2% to 88.9%, outperforming the baseline accuracy of 83.1%. This shows ProtoNet's ability to handle cases with class distributions that are unbalanced and small datasets in medical imaging. Similarly, Apeksha et al. [8] applied ProtoNet to breast ultrasound classification, reaching test accuracies of 85.75%, 91.28%, and 91.67% with 5, 10, and 15 labeled examples, respectively. Their results also demonstrate how ProtoNet can effectively differentiate between benign, malignant, and normal cases, with the added benefit of using novel data augmentation techniques. In their study [29], Junnan Li propose PCL, an unsupervised learning framework that combines clustering and

CHAPTER 2 Literature Review

contrastive learning to train deep neural networks on millions of unlabeled images. Their approach achieved accuracy rates between 73.6% and 86.2%, demonstrating PCL's potential to bridge the gap between representation learning paradigms and advance towards fully self-supervised AI systems.

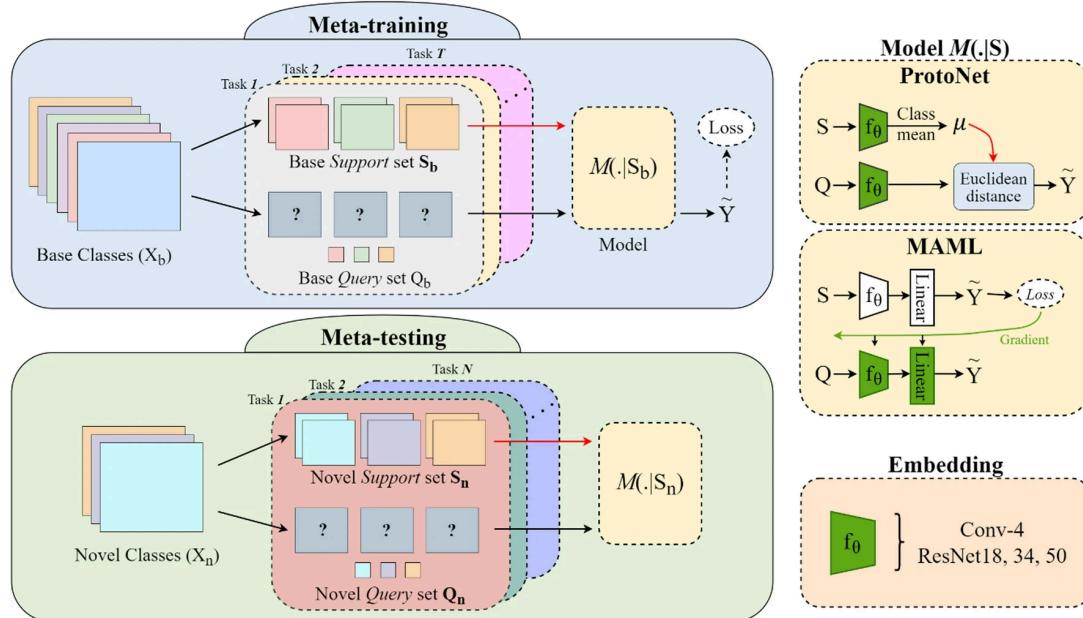


Figure 2.14: Few-shot Classification of Ultrasound Breast Cancer Images Using Meta-Learning Algorithms by Işık, and Paçal

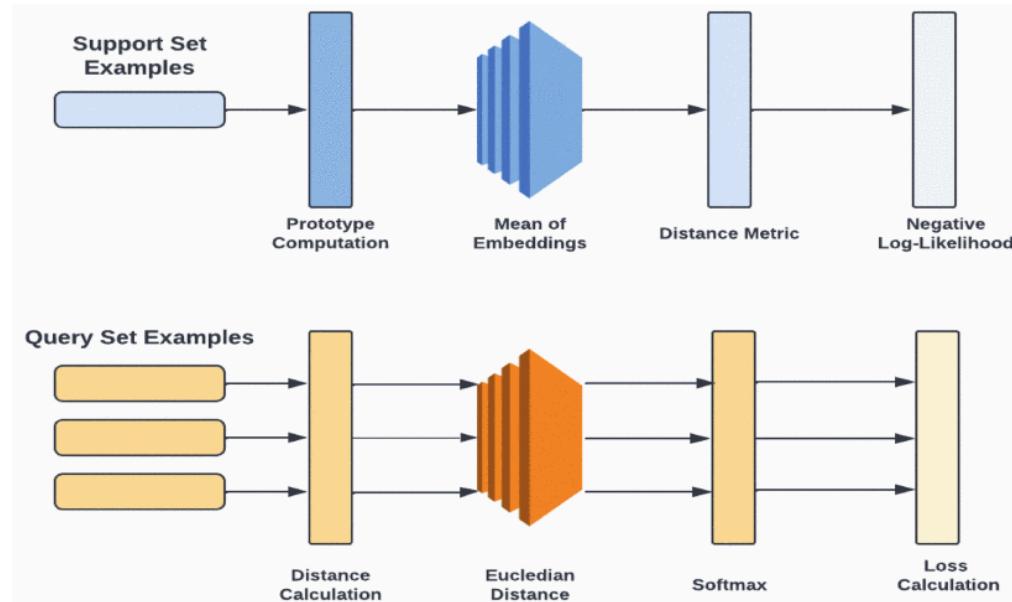


Figure 2.15: Prototypical Neural Network Architecture by Apeksha et al.

CHAPTER 2 Literature Review

To improve ProtoNet's performance with limited labeled data, recent work has combined it with self-supervised and transfer learning techniques. For instance, Medina et al. [19] proposed ProtoTransfer, which uses SSL to create metric embeddings by clustering unlabeled prototypical samples and their augmentations. This approach lessens the need for labelled data and outperforms the current unsupervised meta-learning methods, achieving results close to those of supervised techniques. Similarly, the model built by Xiangyu Y. et al. [15] introduced Prototypical Cross-Domain SSL (PCS), which boosts FUDA by aligning semantic structures between domains using PCL. This results in better accuracy across various domain pairs, outperforming traditional domain-classifier approaches. These developments show how combining SSL with prototypical methods can help manage domain shifts and limited data, making them valuable for medical image classification.

2.1.5 Backbones in Medical Imaging

ResNet, short for Residual Network, is a type of deep convolutional neural network designed to solve the vanishing gradient issue, allowing deeper networks to be trained by using shortcut connections. Sarwinda et al. [26] tested ResNet-18 and ResNet-50 for classifying colorectal cancer in colon gland histopathology images. They discovered that ResNet-50 performed better than ResNet-18 across different test sets, showing it was more reliable and consistent in biomedical image analysis. Besides that, Behar and Shrivastava [27] used a ResNet50-based CNN model to distinguish between malignant and benign breast cancer tumors in histopathology images. They improved the model's performance by using transfer learning and fine-tuning. The model delivered strong results, with high accuracy, precision, and F1 scores, outperforming previous studies.

CHAPTER 2 Literature Review

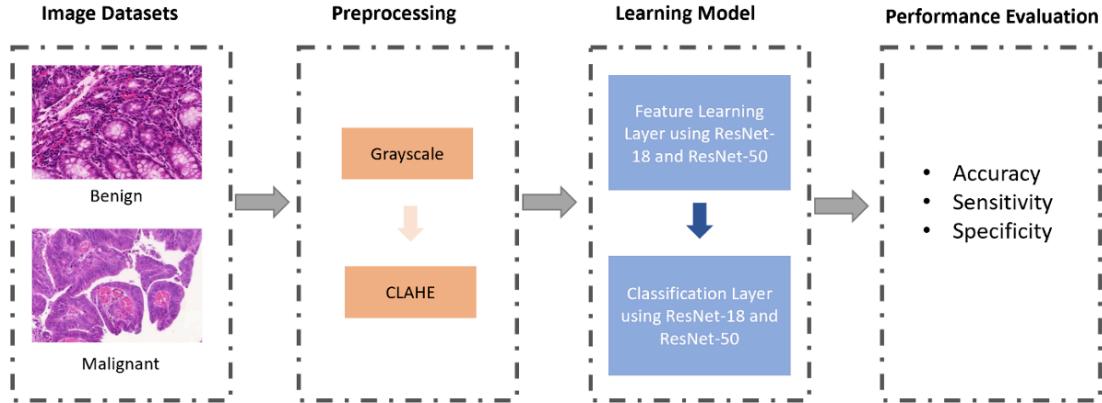


Figure 2.16: Research Flow for Sarvinda et al.'s Proposed Method

Nihar Ranjan Panda et al. employed the MobileNet architecture with transfer learning for breast cancer detection using histopathological images, refining hyperparameters and integrating interpretability features to enhance clinical relevance. Their approach achieved a classification accuracy of 92.92%, outperforming several existing models in differentiating between benign and malignant tumors. Here's a concise two-sentence literature review entry:

Besides, Umesh Kumar et. al. [31] proposed an optimized hybrid MobileNet-V3 model for skin cancer detection, integrating Standard U-Net, improved attention mechanisms, dilated convolutions, and Bayesian hyperparameter optimization to enhance feature extraction and classification. Evaluated on the HAM-10000 dataset, the model achieved 98.86% accuracy, 97.84% precision, 96.35% sensitivity, and 97.32% specificity, outperforming several established architectures including MobileNet, VGG-16, ResNet-152v2, and VGG-19.

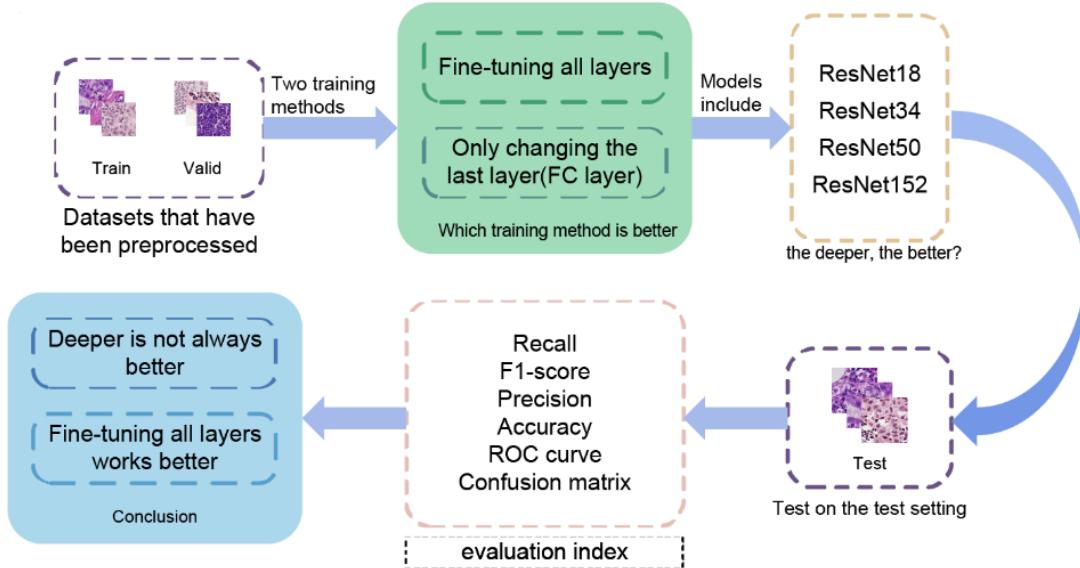


Figure 2.17: Wang et al.'s Flow chart of Preprocessing

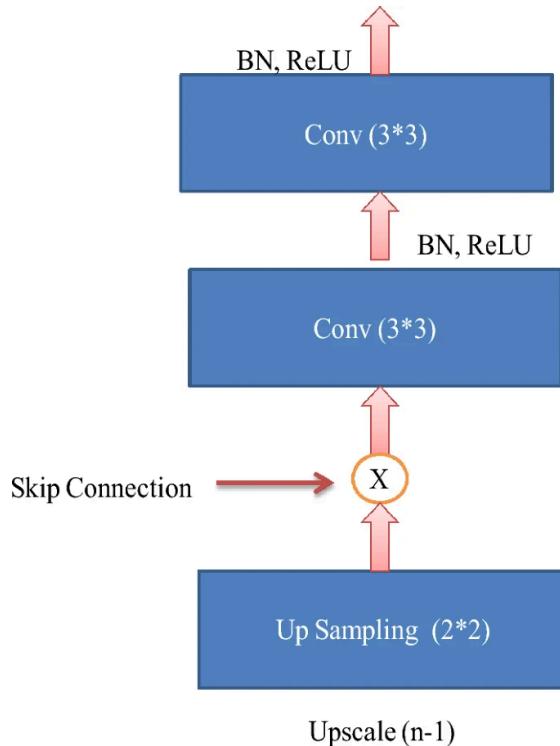


Figure 2.18: Umesh Kumar's U-Net Flow Chart

The study by Musa Adamu Wakili et al. introduces DenTnet, a DenseNet-based transfer learning model for breast cancer histopathological image classification. Their method achieved superior results, reaching 99.28% accuracy on the BreakHis dataset (80:20

CHAPTER 2 Literature Review

split), while also addressing issues of overfitting and high computational cost seen in prior approaches [32]. The work by R. K. Pattanaik et al. proposes a hybrid DenseNet121-ELM model, where DenseNet121 is combined with an Extreme Learning Machine optimized using AdaGrad for breast cancer classification on mammogram images. Using the DDSM dataset, their approach achieved up to 99.47% training accuracy and 99.14% testing accuracy (batch size 128), with very high sensitivity (99.94%) and specificity (99.37%), showing strong potential for clinical application.

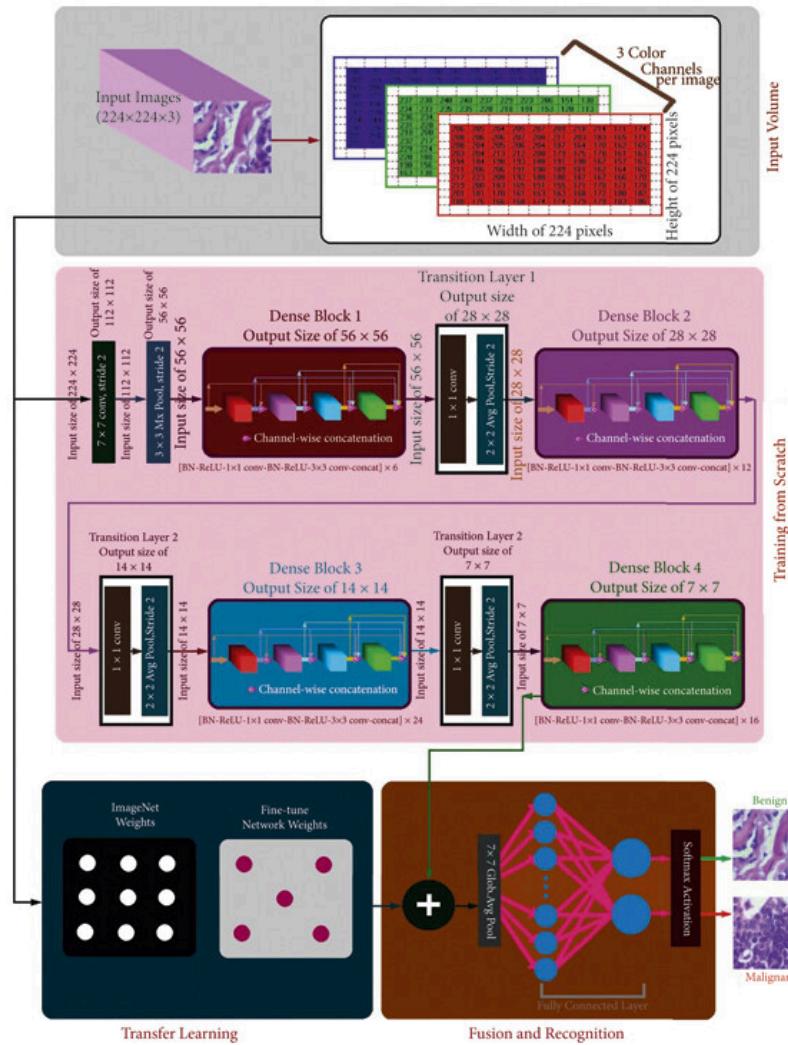


Figure 2.19: Architecture of Dentret as proposed by Musa Adamu et al.

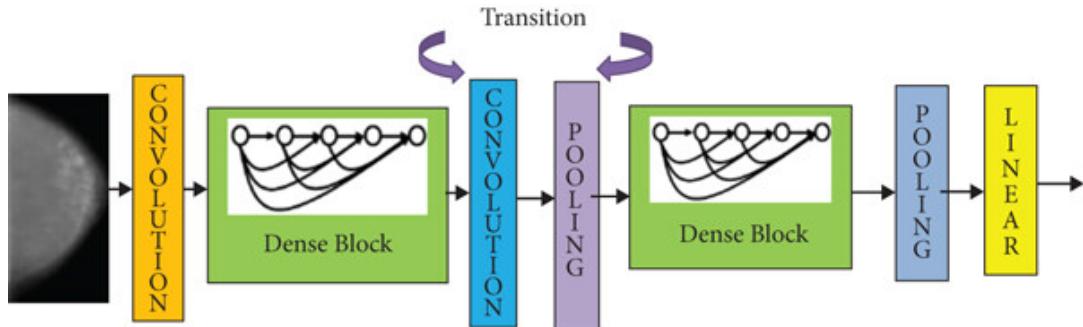


Figure 2.20: The DenseNet Architecture by R. K. Pattanaik

2.2 Limitations of Previous Studies

2.2.1 Limitations of Few-Shot Learning for Medical Image Classification

Singh et al.'s MetaMed approach [13] uses advanced augmentation techniques like MixUp, CutOut, and CutMix to improve generalisation. However, these strategies add complexity to the model training process and may require careful adjustment for different medical datasets. Furthermore, meta-learning methods can be computationally demanding, which limits their use in resource-limited environments. There is also the risk of overfitting to the meta-training tasks, which can make the model less adaptable to new, unseen tasks. Z. Dai et al.'s PFEMed method [14] faces challenges when dealing with datasets that contain many classes. The VAE module's prior branch needs at least one example from each class, which leads to huge batch sizes. Even with fewer classes, the datasets have to be broken into smaller chunks, which adds another layer of complexity. Jiang et al.'s method [17] shows that while larger image sizes can improve performance, they also introduce unwanted noise, which could limit the method's effectiveness in real-world scenarios where image quality can vary.

2.2.2 Limitations of using SSL as a Pretraining Step

While studies that used SSL as a pretraining step yielded innovative results, they also had drawbacks. A common challenge is data dependency, where models' performance is heavily reliant on the quality and quantity of training data, particularly in medical datasets where data may be limited or unbalanced [7] [5] [19]. Another shared issue is

CHAPTER 2 Literature Review

overfitting in large networks when dealing with limited labelled data, resulting in reduced performance, and generalisation problems, especially across different domains [18] [19] [15]. Additionally, domain gap, and sensitivity to target domains pose difficulties for transfer learning in few-shot settings, where the performance declines significantly when there are large disparities between the training, and testing sets or diverse target domains [18] [19] [10].

Moreover, network complexity, and computational demands are frequently highlighted, where the complexity of models like bi-channel networks increases resource requirements, and may necessitate careful tuning [7] [19]. Some studies also point out the ineffectiveness of pretext tasks in generalising to other tasks or domains [7] [19]. At the same time, noise in results due to inadequate annotations remains a limitation in medical image segmentation tasks [5]. In some cases, specialised methods like IP-IRM may lead to overly specific representations, negatively impacting performance in very low-data scenarios [12]. Lastly, the reliance on the pre-training quality, and limited data variety for distortion testing further hinder generalisation across broader applications [19] [20].

2.2.3 Limitations of AI Integration in Diagnostic Systems

The paper by Z. Yang et al. [1] on MABEL faces limitations, including a primary focus on single-view performance, with multi-view analysis yet to be fully integrated or assessed, and insufficient consideration of additional time factors affecting overall diagnostic efficiency, such as image acquisition and doctor review. Similarly, Berberian et al. [2] encounter limitations due to a small sample size, and a single-center study, which restricts the ability to calculate statistically significant differences for tumour subtypes, and limits the generalizability of their findings. The lack of access to a cancer registry prevents tracking interval cancers and, restricts comparisons to cancers detected by mammography and biopsy-proven or specimen-proven cases. Also, the small sample sizes limited how well different subgroups could be evaluated. Both studies however points out the need for bigger, diverse datasets and better methods to tackle these issues.

2.2.4 Limitations of ProtoNet, and Transfer Learning in Few-Shot Learning

Studies on FSL techniques, like transfer learning and prototypical networks, have identified several common limitations. Both Işık, and Paçal [4], and Apeksha et al. [8] emphasize that more research must be done on data augmentation techniques to mitigate overfitting and improve learning from sparse datasets. Moreover, little research has been done on how distance metrics can enhance model accuracy. The prototypical cross-domain SSL framework, which primarily relies on PCL, might struggle with capturing more complex category structures. Even after aligning semantic structures, it can still have difficulty with aligning low-level features, as pointed out by Xiangyu et al. [15]. Medina et al. [19] emphasized the significance of a high-quality self-supervised pre-training in this field. They noted that poor pre-training can hurt classification results. They also highlighted the computational challenges of SSL and fine-tuning, which often require great amounts of resources.

2.2.5 Limitations of Backbones in Medical Imaging

Although the results from these studies are promising, there are still a few limitations to consider. For one, the quality and variability of mammography images can greatly impact the model's performance. These images often vary in resolution and imaging conditions, as pointed out by Sarwinda et al. [26]. Moreover, the lack of diverse training data particularly is the case for mammography often leads overfitting or decreased generalisability, as highlighted by Behar and Shrivastava [27]. Another issue which was pointed out was class imbalance, where malignant cases are far less frequent than benign ones, which can affect the model's ability to accurately classify images, a challenge discussed in multiple studies on medical imaging [28]. As for MobileNet, the integration of two complex architectures (MobileNet-V3 with U-Net Architecture) increases computational demands during both training and inference, potentially limiting its practicality in resource-constrained environments. Meanwhile, the drawback of the Dentret model (hybrid of DenseNet and Transfer Learning) as detailed by Musa Adamu et. al. is that the model relies solely on the deep network based features that were extracted from the raw images, which may limit its robustness[32]. R. K. Pattanaik et al's

CHAPTER 2 Literature Review

DenseNet-121 model on the other hand required a substantial amount of time to simulate, which they highlight as the drawback of the model.

2.3 Proposed Solutions

Table 2.1: Proposed Solutions of Previous Works

No.	Authors	Year	Techniques	Outcome	Future Works, further exploration into:
1	Singh et al.	2021	Reptile Algorithm Gradient-Based Meta-Learning	Accuracy > 70%	Gradient-based meta-learning for multiple modalities of medical images.
2	Jiang et al.	2022	Meta-Learning - Autoencoder - Metric Learner - Task Learner. GDSL Scheme	Average Accuracy: 95% of confidence interval	Multi-learner-based model
3	Z. Dai et al.	2022	Pfemed Variational Autoencoder	Outperforms by 2.63%	FSL
4	R. Gong et al.	2021	CADe Task-Driven Self-Supervised Bi-Channel Network (TSBN) Framework Residual Dense Network (RDN) U-Net	Accuracy: TSBN-U: 85.53% TSBN-R: 85.78% Sensitivity: TSBN-U: 84.00% TSBN-R: 83.00% Specificity: TSBN-U: 86.07% TSBN-R: 86.75% F1: TSBN-U: 75.06% TSBN-R: 75.18%	Ways to design different types of pretext tasks in the TSBN framework. Transfer performance between two different networks.

CHAPTER 2 Literature Review

5	H. Quan et al.	2024	DCPN SSL Multi-Scale Feature Extraction	Accuracy: 1 shot: 70.86% 5-shot: 82.57% 10-shot: 85.2%	Large-scale self-supervised pretraining, and domain adaptation strategies
6	Chen et al.	2019	Patch-Level Unsupervised Cell Ranking Approach	Accuracy: Few-shot: Resnet50: 0.8601 VGG19: 0.8720 Inception V3: 0.8961 Fully Supervised: VGG16: 0.9790 Inception V1: 0.9840	Incorporate unlabeled data. The ‘learning to rank’ idea.
7	Eva Pachetti, S. Tsaftaris, and S. Colantonio	2024	Disentangled SSL Meta-Learning	Final Ablation Experiment: Highest AUROC: 0.533, 0.620, 0.585, and 0.617 Fully-Supervised Fine-Tuning AUROC: ResNet-18: 0.923 ResNet-50: 0.946 VGG-16: 0.853 DenseNet-121: 0.896	Meta-learning algorithm Create more detailed prototypes for enhanced capabilities
8	Xiangyu Y. et al.	2021	Prototypical Cross-Domain SSL Few-Shot Unsupervised Domain Adaptation	Accuracy: 76.1% with 94% less labelled source images	FUDA, and PCS Framework
9	D. Chen et al.	2021	Contrastive SSL With Episodic	Accuracy: 84.29%	To obtain higher accuracy in the ISIC

CHAPTER 2 Literature Review

			Training Embedding Network With SSL)		dataset. Combining fine-tuning by a meta-learning process Cross-domain FSL setup Develop an end-to-end method. Few-shot detection
10	Medina et al.	2020	Prototransfer Self-Supervised Prototypical Transfer Learning	outperforms state-of-the-art unsupervised meta-learning methods by 4% to 8% on mini-ImageNet few-shot classification tasks	ProtoTrasfer for few-shot classification.
11	Koch et al.	2015	Siamese Networks	Accuracy 92%	More detailed algorithm that uses information on stroke paths that are specific.
12	Z. Yang et al.	2021	U-Net Architecture Picture Archiving, And Communication System (Pacs)	97% of the time consumption of prediction generation is less than 20 seconds. NPV of mass detection branch: 96.5% NPV of calcification detection branch: 97.3%	Create a connection between the annotation tool, and the web server. To enhance the overall efficiency of lesion predictions
13	N. Berberian et al.	2024	Portable Thermal Imaging Solution Powered By AI Random Forest Classifiers U-Net Architecture	Sensitivity: 89% Specificity: 83%	The monitoring of neoadjuvant therapy effects for breast cancer could have additional uses.
14	Işık, and Paçal	2024	Prototypical Networks	Accuracy:	Exploring data augmentation techniques

CHAPTER 2 Literature Review

			Model Agnostic Meta-Learning Meta-Training	0.882–0.889	like Mixup, CutMix, and Random Erasing to mitigate overfitting, and improve learning from limited datasets.
15	Apeksha et al.	2024	Protonet Convolutional Neural Network Architecture FSL In The Meta-Learning Framework	Accuracies: 85.75%, 91.28%, 91.67%	Integrating clinical data Collaborating with medical experts
16	E. Pachetti, and S. Colantino	2023	FSL Meta-Learning Supervised Learning Semi-Supervised Learning	95% confidence interval	Combined use of meta-learning, and SSL Meta-learning, SSL, and semi-supervised learning approaches. Broaden the scope of medical applications (beyond heart, abdomen, and lung anatomical structures)
17	Devvi Sarwinda, Radifa Hilya Paradisa, Alhadi Bustamam, Pinkie Anggia	2021	DL in Image Classification using Residual Network (ResNet) Variants for Detection of Colorectal Cancer	Accuracy: 80% Sensitivity: 87% Specificity: 83%	Added architectures with their variants using the corresponding datasets.
18	Nishant Behar, Manish Shrivastava	2021	ResNet50-Based Effective Model for Breast Cancer Classification Using Histopathology Images	Training accuracy: 99.70% Validation accuracy: 99.24% Test accuracy: 99.24%	Network complexity, image channels, computational time and space complexity. Multiclass classification on the same dataset will also be carried out.

CHAPTER 2 Literature Review

19	Ziying Wang, Jinghong Gao, Hangyi Kan, Yang Huang, Furong Tang, Wen Li and Fenglong Yang	2024	ResNet for Histopathologic Cancer Detection, the Deeper, the Better?	Accuracy: similar results for all AUC: ResNet18/ ResNet34 : ~0.88 ResNet50/ ResNet152: ~0.92 ResNet-34: 0.992, ResNet-152: 0.989	Network structures, fine-tuning methods, data augmentation
20	Junnan Li	2020	Prototypical Contrastive Learning: Pushing the Frontiers of Unsupervised Learning	Accuracy 75.6%, to 86.2%	-
21	N. R. Panda, Debendra Muduli, and S. K. Sharma	2024	Customized MobileNet with Transfer Learning for Enhanced Early Breast Cancer Detection: A DL Approach	Benign (Class 0): Precision > 0.90, Recall > 0.90, F1 > 0.90 Malignant (Class 1): Precision = 0.95, Recall = 0.84, F1 = 0.89 Overall Accuracy: 0.93	Dataset diversity Fine-tuning model hyperparameters, Integrating the model into clinical workflows in real world settings
22	U. Kumar Lilhore et al.	2024	A precise model for skin cancer diagnosis using hybrid U-Net and improved MobileNet-V3 with hyperparameters optimization	accuracy: 98.86%, precision: 97.84%, sensitivity: 96.35%, specificity 97.32%	advancements in automated skin cancer detection
23	Musa Adamu Wakili, Harisu Abdullahi	2022	Classification of Breast Cancer Histopathological Images Using DenseNet and Transfer Learning	Accuracy: 99.28% Sensitivity: 97.73% Specificity: 100% AUC: 0.99	Tested only on a single breast cancer dataset (BreaKHis). Unclear generalizability to other breast cancer datasets.

CHAPTER 2 Literature Review

	Shehu, Md. Haidar Sharif, Md. Haris Uddin Sharif, Abubakar Umar, Huseyin Kusetogullari, Ibrahim Furkan Ince, Sahin Uyaver				Relies solely on deep-network-based features. Does not incorporate complementary descriptors (e.g., LBP).
24	R. K. Pattanaik, S. Mishra, M. Siddique, T. Gopikrishna, and S. Satapathy	2022	Breast Cancer Classification from Mammogram Images Using Extreme Learning Machine-Based DenseNet121 Model	Accuracy (Training): 99.47% Accuracy (Testing): 99.14% Sensitivity: 99.94% Specificity: 99.37% Computational Time: 159.77 minutes (batch size 128)	High computational time required

CHAPTER 3

System Model

3.1 System Design Diagram

3.1.1 System Architecture Diagram

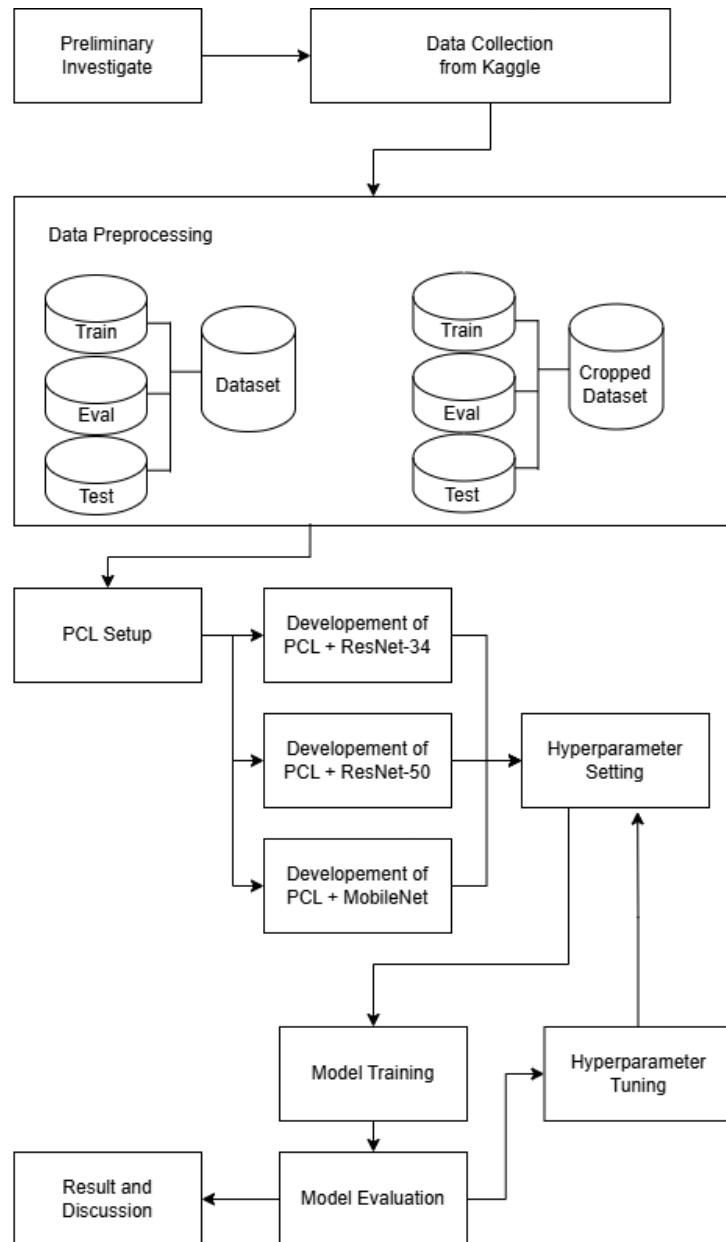


Figure 3.1.1: System Architecture Diagram

Preliminary Investigation

This phase involved reviewing the existing research and works on SSL and its role in medical imaging while identifying research gaps. The objective was to understand the challenges of mammography classification under limited-data conditions without it being computationally expensive. From this research, PCL was chosen to be the core of the framework for this study.

Data Collection

The dataset used in this research was obtained from Kaggle's Breast Ultrasound Images dataset. The images were grayscale and were also categorised into the following three classes: benign, malignant, and normal. These images formed the very foundation for this research, as they were the data used to create the two datasets used, the original dataset and the cropped dataset.

Data Preprocessing

Two datasets were prepared: the original and a dataset with the images cropped (removing the excess background and thus focusing on the area of concern). Both were then split into training evaluation and testing subsets to ensure that this distribution would avoid data leakage.

PCL Setup

The PCL framework was then modified for the implementation. Modifications included taking out the DDP for single-GPU use and adjusting preprocessing to handle the grayscale medical images.

Development of ResNet-34, ResNet-50, MobileNet, and DenseNet-121

Different backbone architectures were integrated into PCL to compare their performance. ResNet-34 and ResNet-50 provided deep residual learning, MobileNet offered a lightweight alternative, and DenseNet-121 contributed efficient feature reuse. Each backbone was modified to exclude its classification layer and serve purely as a feature extractor.

Hyperparameter Setting, Tuning, and Training

Initial experiments used default parameters as a baseline. The research then performed systematic tuning of all key parameters, including batch size, learning rate, temperature, and dataset variation (original vs. cropped). Each configuration was tested to evaluate its effect on model training stability, representation quality, and final performance.

Model Evaluation

Trained models were then evaluated using the confusion matrix, which allows us to derive the values for accuracy, precision, recall, and F1-score. Additionally, t-SNE and UMAP visualisations were applied to study the clustering of learned feature embeddings and the separation between classes.

Result and Discussion

The findings from all experiments were analyzed to identify which backbone performed best within the PCL framework, which so far has only been used to analyze highly distinct classes. Comparisons between the original and cropped datasets highlighted the impact of background removal. The discussion also addressed the strengths, limitations, and future potential of PCL in medical imaging.

3.2 Implementation

The following section presents the implementation aspects of the project. The goal was to design and to conduct an evaluation on an SSL framework, which is based on PCL for breast cancer image classification under limited-data conditions. To achieve this, the experiments were structured into several phases, which in turn ensured that the research was both systematic and reproducible.

3.2.1 Hardware Setup

For this research, a laptop with a sufficient amount of memory, solid storage, and a fast processor was employed. The system runs with an 11th Gen Intel Core i5 processor, improving the overall performance. With 8 GB of DDR4 RAM, it guarantees efficient data handling and improved processing speed. The use of an SSD for storage allows quick data access and offers ample space. The specific laptop model used for the development of this project is detailed in Table 3.1.

Table 3.1 Specifications of Laptop

Description	Specifications
Model	ASUS Vivobook 14
Processor	11th Gen Intel Core i5-1135G7 @ 2.40 GHz
Operating System	Windows 10 Home Single Language, Version 22H2
Graphic	Integrated Graphics (not specified)
Memory	8 GB DDR4 RAM
Storage	1 TB PCIe 3.0 SSD

3.2.2 Software Setup

This project was entirely developed using the Google Colab Pro platform, which is a cloud-based Jupyter environment that gave this project access to computational resources such as the T4 GPU. With Colab Pro, there is no need for high-end local hardware.

All implementations for this experiment to take its shape were carried out in Python, which was specifically chosen for its extensive ecosystem in the realm of scientific

CHAPTER 3 System Model

computing and machine learning. The framework, which was the foundation of this experiment, was PyTorch. Torchvision was used for data handling. Additional libraries such as NumPy for numerical operations, Matplotlib for visualisation, and scikit-learn for the evaluation metrics were put into use. Another notable library used in the development was Timm library, which allowed for the implementation of MobileNet and DenseNet-121.

3.2.3 Setting and Configuration

3.2.3.1 Dataset

The dataset used in this project consists of breast ultrasound images collected from women aged between **25 and 75 years**. The dataset was organised in 2018 and contains a total of **780 images**. These images were sourced from Kaggle, with **600 female patients** included in the dataset. The ground truth labels are provided alongside the original images.

Dataset Source

The dataset for this project consists of grayscale mammography images from [Kaggle:Breast Ultrasound Images Dataset](#).

Link: <https://www.kaggle.com/datasets/sabahesarak/breast-ultrasound-images-dataset>

The images in it are organized into three folders based on the images' classifications: benign, malignant, and normal. The dataset is provided in a flat structure. Instead, a custom dataset splitting strategy is implemented. First, the dataset was split into two directories:

- *dataset/* – contains the original, unmodified images
- *dataset_cropped/* - contains the cropped images edited with the ImageJ application.

Dataset Splitting

Then stratified sampling was applied in each of these datasets by further splitting the subdirectories for training, validation, and testing to ensure no data leakage between sets. The splitting was done in a balanced manner across the three classes (benign, malignant, and normal), with 93 images used for training, 27 images for validation, and 13 images for testing. To be certain that there would not be any class imbalances, equal distribution across the subdirectories was implemented. Initially, the normal class had twice as many images as the malignant class, causing a huge bias in the model; by applying all the classes, this issue was fully resolved. Below is the breakdown of the class distribution.

Training (Total: 279)	Validation (Total: 81)	Testing (Total: 39)
<ul style="list-style-type: none"> - Benign: 93 - Malignant: 93 - Normal: 93 	<ul style="list-style-type: none"> - Benign: 27 - Malignant: 27 - Normal: 27 	<ul style="list-style-type: none"> - Benign: 13 - Malignant: 13 - Normal: 13

3.2.3.2 Image Preprocessing

In this study, ImageJ was also employed to crop mammography images in order to remove excess background and focus solely on the breast tissue containing potential abnormalities. This preprocessing step aimed to reduce noise from irrelevant regions, such as the black borders or non-breast anatomical areas, which could otherwise introduce distractions and hinder the model's ability to learn meaningful features. By creating a cropped version of the dataset alongside the original, the research was able to systematically compare the impact of background removal on feature extraction, representation learning, and classification accuracy within the PCL framework. This approach allowed for a targeted investigation into whether eliminating extraneous visual information could enhance model performance in detecting breast cancer from mammography images.

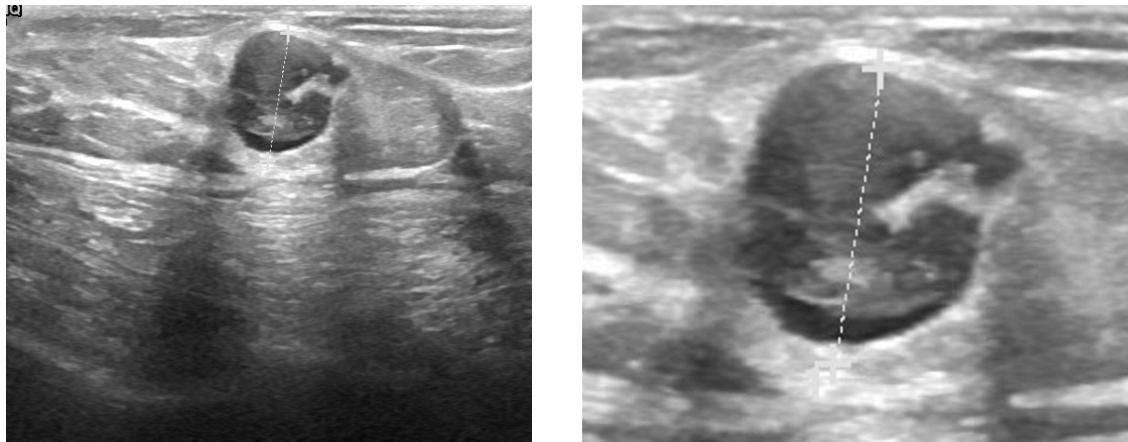


Figure 3.2 Example of Mammography Images Before (Left) and After Cropping with ImageJ (Right)

In this study, data augmentation was deliberately avoided to maintain the medical and experimental integrity of the work. Augmentations such as rotations and flips can introduce anomalies that are unrealistic variations in mammograms, which may in this case potentially compromise the clinical validity of the data. Similarly, aggressive transformations add risks of distorting subtle but critical lesion patterns, which could increase label noise and hinder accurate feature learning. By preserving the dataset in its original form, the model is trained only on authentic medical data, ensuring that results reflect its true capacity rather than improvements inflated by synthetic samples. This also guarantees fairness in evaluation, as performance metrics directly correspond to the dataset provided without augmentation-based bias. Moreover, since the primary goal was to investigate PCL under limited-data conditions, introducing augmented images would blur the distinction between gains from self-supervised learning and those from artificial data expansion. Finally, avoiding augmentation helped reduce preprocessing complexity and computational overhead, keeping the experiments more controlled and manageable.

3.2.3.3 PCL Framework Configuration

PCL Framework Setup and Adaptation

The implementation of the PCL framework in this study was based on the work of Junnan Li et al. [29], which introduced the method as a unified approach combining clustering and contrastive learning for unsupervised representation learning. The official PCL source code was obtained from its publicly available GitHub repository (<https://github.com/salesforce/PCL>), ensuring that the experimental setup followed the original design principles and methodological integrity of the authors' work. This provided a solid and validated foundation for adaptation to the specific requirements of breast cancer mammography classification under limited-data conditions. Additionally, the original implementation's DDP setup was removed, as the experiments in this study were conducted on a single GPU environment where distributed training was unnecessary and could introduce added complexity without performance benefits.

To align the framework with the scope of this research, the original PCL codebase was further modified to incorporate different backbone architectures—ResNet-34, ResNet-50, MobileNet, and DenseNet-121—allowing for comparative evaluation of their performance within the PCL setting. These adaptations required adjustments to the model configuration, training pipeline, and preprocessing routines to handle grayscale mammography images and ensure compatibility across the chosen backbones. Through these modifications, the system design was tailored to address the challenges of high intra-class similarity and limited annotated medical imaging data, extending the applicability of PCL beyond its original domain of visually distinct datasets.

Distributed Data Parallel

As aforementioned, this research was conducted on Google Colab, which gives access to a single NVIDIA T4 GPU. However, the original PCL implementation operated on multi-GPU training. To resolve this conflict, DDP was thus disabled. The following will go into detail on how that was carried out.

- Disabled the distributed training imports and configs

CHAPTER 3 System Model

- import torch.distributed as dist was commented out.
- args.distributed was hardcoded to False, and args.multiprocessing_distributed to False.
- args.world_size and args.rank were set to single-node defaults (1 and 0).
- Removed DDP wrapping of the model

Original

```
model = torch.nn.parallel.DistributedDataParallel(model,  
device_ids=[args.gpu])
```

Updated to:

```
model = model.cuda(args.gpu)  
print("Warning: Running without DistributedDataParallel. Proceeding on  
single GPU or CPU.")
```

This change ensured the model was placed directly onto the available GPU without invoking DistributedDataParallel, which is unnecessary for single-GPU training. It simplified execution, removed the need for process synchronization, and allowed the PCL framework to run smoothly in the Colab environment.

- Simplified dataloader and synchronisation
 - Removed all the barriers (dist.barrier()) and collective communications (dist.all_reduce) that were required for multi-GPU coordination.
 - Dataloaders now just shuffle the dataset without distributed samplers.
- Changed FAISS clustering backend
 - Instead of using GPU FAISS (GpuIndexFlatL2), your version uses the CPU FAISS index (IndexFlatL2).
 - This avoids requiring multiple GPUs and reduces complexity while still supporting k-means clustering for prototypes.

By removing the DDP, the project avoided the synchronization errors and greatly reduced the configuration complexity. Likewise, removing DDP, which was designed for clusters/multi-GPU servers, enabled the code to run on Google Colab on a single PC.

Data Loading Simplification

CHAPTER 3 System Model

In the original PCL code, training and evaluation dataloaders were tied to `torch.utils.data.distributed.DistributedSampler`, which ensures that multiple GPUs process unique, non-overlapping portions of the dataset during distributed training.

Since removing the DDP, this part was also changed to be compatible with the single GPU environment. Hence, this distributed sampler was removed because only a single GPU was used. Instead, the dataloaders were defined in standard PyTorch mode:

```
train_sampler = None
eval_sampler = None

train_loader = torch.utils.data.DataLoader(
    train_dataset, batch_size=args.batch_size, shuffle=True,
    num_workers=args.workers, pin_memory=True, drop_last=True)

eval_loader = torch.utils.data.DataLoader(
    eval_dataset, batch_size=args.batch_size * 5, shuffle=False,
    num_workers=args.workers, pin_memory=True)
```

Checkpoint Saving

Another part of the PCL that was modified was the checkpoint saving. In the original code, the checkpoint saving was tied to GPU rank to avoid multiple processes writing to disk simultaneously. In our case, this restriction was unnecessary because only one process was running at a time. Hence, this logic was simplified to directly write the checkpoints at fixed intervals. This made the checkpointing process more straightforward and reliable. This is crucial for a project such as this, where running on Google Colab may put the training runs at risk of interruptions.

3.2.3.4 Training Configuration

Optimizer

The Stochastic Gradient Descent (SGD) was used in this training as the optimizer. SGD updates the model's parameters iteratively by going towards the direction of the negative gradient of the loss function. The momentum of this optimiser was set to 0.9. This will help the optimizer maintain its direction across iteration, and also makes it so that it is able to avoid oscillations. A weight decay of 1e-4 was also applied to act as the optimiser's L2 regularisation. This, in return, will reduce the risk of overfitting by penalising large weights.

CHAPTER 3 System Model

```
optimizer = torch.optim.SGD(model.parameters(), args.lr,
                           momentum=args.momentum,
                           weight_decay=args.weight_decay)
```

Learning rate schedule

The learning rate scheduling strategy that was applied for this research is the step decay. In step decap, the LR drops by a factor of ten at specific milestones (it was set to 120 and 160). This was applied to prevent the optimiser from overshooting once the model reaches the latter stages of the convergence process. In shorter runs, such as 94 epochs, this schedule will not be triggered; however, in longer runs, such as this experiment, the step decay took effect as specified (0.0005 and 0.001). The learning rates were set to these smaller values to account for the fact that the medical dataset at hand was limited in size, and setting higher learning rates would cause the convergence to be extremely unstable.

Loss function

The training objective combined two loss functions under a single framework:

- InfoNCE Loss
 - Instance-level contrastive loss that encourages embeddings of two augmented views of the same image to be similar, while pushing apart embeddings from different images.
- ProtoNCE Loss
 - Prototype-level contrastive loss that aligns instance embeddings with their corresponding cluster centroids (prototypes), improving semantic structure in the learned representations.

Both the InfoNCE Loss and the ProtoNCE Loss were implemented using CrossEntropyLoss:

```
criterion = nn.CrossEntropyLoss().cuda(args.gpu)
```

Cross-entropy was used in this experiment because of its contrastive objective properties. It can formulate the classification problems, where the positive pairs are the “correct class”, and on the flipside, all negatives are treated as the “incorrect class.”

3.2.4 Experiment Organisation

The experiments were essentially organised into a baseline run, then followed by a series of controlled variations, with each of them modifying only a single parameter to evaluate for that run. The default run was established to use the original dataset, with a batch size of 16, LR of 0.0005, and a temperature of 0.2. Variations were then systematically introduced to this baseline across four aspects as follows:

- Dataset
 - Comparing the original mammography dataset (Ori) with the cropped dataset (Crop) to study the effect of background removal.
- Batch Size
 - Increasing from 16 to 32 to analyse the effect of more negative samples in contrastive learning.
- Learning Rate
 - Raising the learning rate from 0.0005 to 0.001 to test convergence speed and stability.
- Temperature
 - Lowering from 0.2 to 0.1 to investigate sharper feature separation in the embedding space.

Each experiment was then labelled according to its backbone, dataset, and its varied parameter. Below is a sample subset of runs.

Table 3.2: Example of Experimental Runs (ResNet-50 backbone)

Backbone	Dataset	Run type	Batch Size	LR	Temperature
ResNet50	Ori	Default	16	0.0005	0.2
ResNet50	Crop	Var: Dataset	16	0.0005	0.2
ResNet50	Crop	Var: Batch	32	0.0005	0.2
ResNet50	Crop	Var: LR	32	0.001	0.2
ResNet50	Crop	Var: Temp	32	0.0005	0.1

CHAPTER 3 System Model

Logging and monitoring were then done manually by recording the validation accuracy, precision, recall, F1-score, and the confusion matrix at the end of each run. The embedding visualisations, such as the t-SNE and UMAP, were also generated in the post-training to further visualise the features extracted.

3.3 System Operation

3.3.1 Dataset Preparation & Directory Structure

Figure 3.3 shows the dataset directory structure as prepared in Google Colab. Both the datasets used in this study (the original dataset and the cropped dataset) were uploaded onto Google Drive first and then were mounted into Google Colab for easier access. For training, validation, and testing, the dataset was arranged into subdirectories, with each class—normal, malignant, and benign—in its own folder. Reproducible experimentation was made possible by this organization's clear separation of data splits, which prevented leakage between training and evaluation.



Figure 3.3: Screenshot of Google Colab Folder showing the subdirectories of the dataset used

3.3.2 Cropping

Figure 3.4 depicts the preprocessing step of cropping the mammography image using the ImageJ software. In order to make sure that the model concentrates on the breast tissue where the mass is present, the cropping process was used to eliminate the extra background regions, such as black borders and unnecessary anatomical areas. In

subsequent experiments, a controlled comparison between the original and cropped datasets was made possible by this step, which also decreased noise in the dataset.

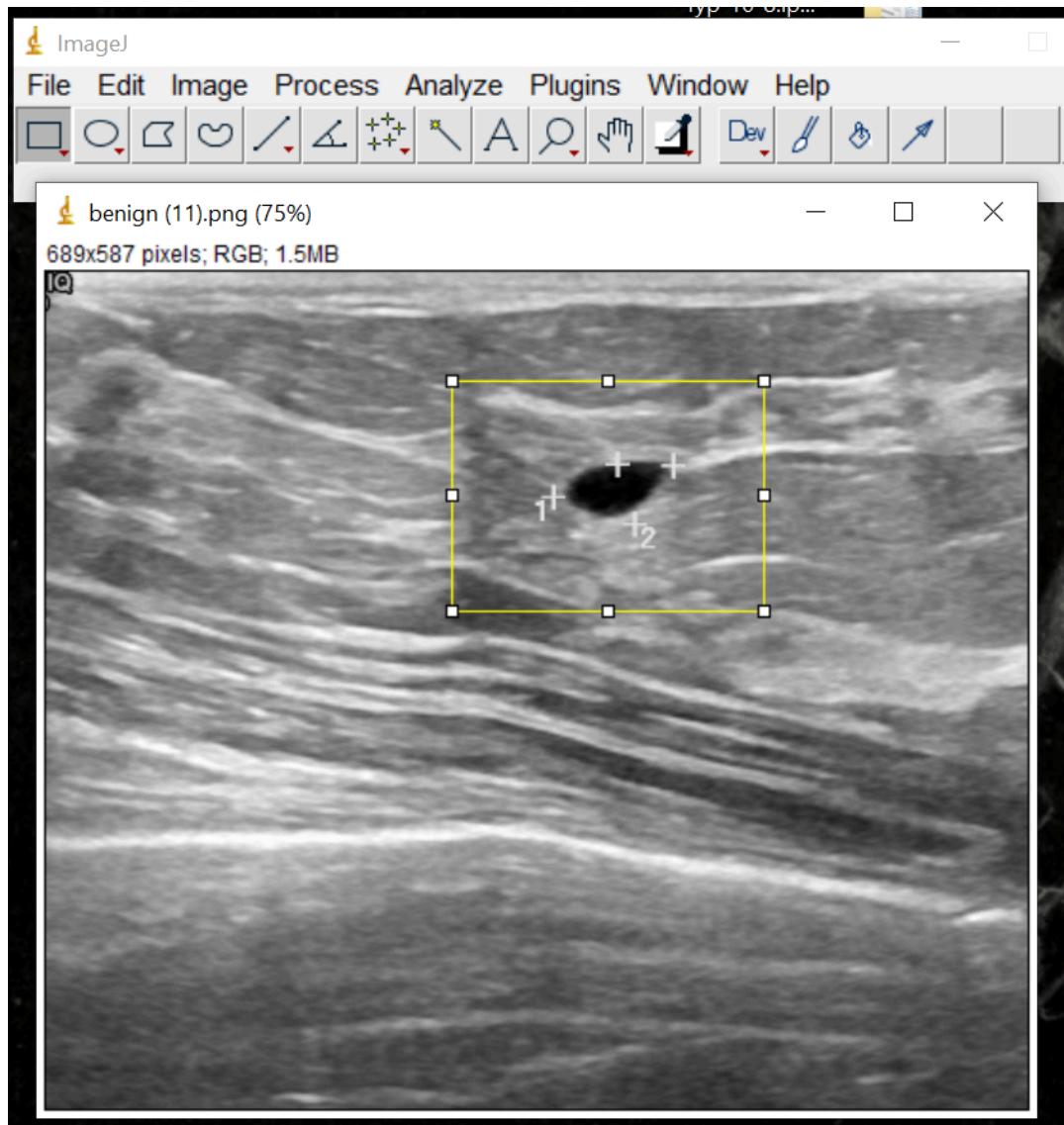


Figure 3.4: Screenshot of removing the excess background of a benign image for the cropped dataset using ImageJ

3.3.3 Colab Runtime Setup

The Colab environment confirming the allocation of an NVIDIA Tesla T4 GPU using the `!nvidia-smi` command is shown in Figure 3.5.. In order to manage the demanding tasks needed by deep learning models like ResNet, MobileNet, and DenseNet within the PCL framework, GPU acceleration of the Google Colab environment was essential for this study.

CHAPTER 3 System Model

```
▶ !nvidia-smi
Tue Sep 16 08:15:54 2025
+-----+
| NVIDIA-SMI 550.54.15      Driver Version: 550.54.15      CUDA Version: 12.4 |
+-----+
| GPU  Name      Persistence-M  Bus-Id      Disp.A  Volatile Uncorr. ECC |
| Fan  Temp     Perf          Pwr:Usage/Cap | Memory-Usage | GPU-Util  Compute M. |
|          |             |             |           |           |          MIG M. |
+-----+
|   0  Tesla T4      Off          00000000:00:04.0 Off    0          |
| N/A  39C   P8      9W /  70W          0MiB / 15360MiB | 0%      Default |
|          |             |           |           |           |          N/A |
+-----+
+-----+
| Processes:                               GPU Memory |
| GPU  GI  CI      PID  Type  Process name        Usage  |
| ID   ID          |
+-----+
| No running processes found
+-----+
```

Figure 3.5: Screenshot of Colab Environment showing the GPU (T4) allocation

3.3.4 Training Execution in Colab

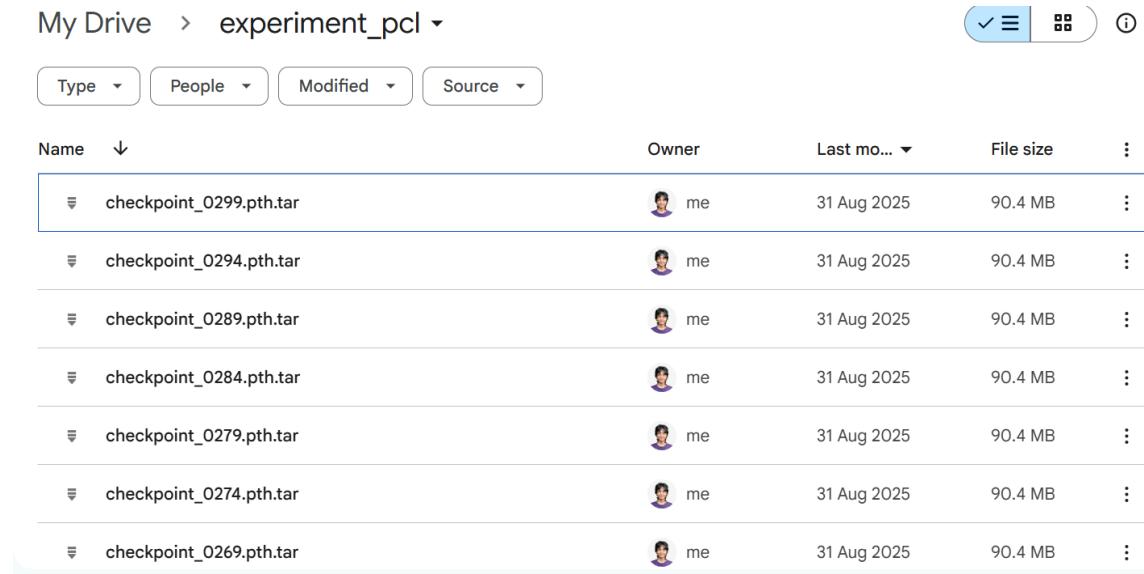
The training log for ResNet-34 on the cropped dataset at epoch 60 (batch size = 32, learning rate = 0.0005, temperature = 0.1) is shown in Figure 3.6. It depicts the process that happens in each of the epoch and finally at the end of it displays the loss value along with the AccProto value. A total of 300 epochs were run by this study for every variation of a model on Google Colab.

```
[DEBUG] Epoch 60 | warmup_epoch = 0 | Clustering will run: True
Computing features...
100% 2/2 [00:05<00:00,  2.62s/it]
performing kmeans clustering
Clustering 279 points in 128D to 3 clusters, redo 5 times, 20 iterations
  Preprocessing in 0.00 s
Outer iteration 0 / 5
  Iteration 19 (0.00 s, search 0.00 s): objective=2118.39 imbalance=1.428 nsplit=0
  Objective improved: keep new clusters
Outer iteration 1 / 5
  Iteration 19 (0.00 s, search 0.00 s): objective=2192.38 imbalance=1.028 nsplit=0
Outer iteration 2 / 5
  Iteration 19 (0.00 s, search 0.00 s): objective=2068.29 imbalance=1.328 nsplit=0
  Objective improved: keep new clusters
Outer iteration 3 / 5
  Iteration 19 (0.00 s, search 0.00 s): objective=2064.55 imbalance=1.361 nsplit=0
  Objective improved: keep new clusters
Outer iteration 4 / 5
  Iteration 19 (0.01 s, search 0.00 s): objective=2201.31 imbalance=1.044 nsplit=0
Epoch: [60][0/8]      Time 5.585 ( 5.585)      Data 4.935 ( 4.935)      Loss 7.7308e+00 (7.7308e+00)      Acc
```

Figure 3.6: Screenshot of Epoch 60 of ResNet-34 (cropped dataset, batch size = 32, LR = 0.0005, T = 0.1)

3.3.5 Checkpoint Saving

The checkpoints that were automatically saved in Google Drive throughout the experiments are shown in Figure 3.7. Training progress was maintained through checkpoint saving, which made it possible to resume interrupted runs without losing previous calculations. Working with Colab, which has session time limits that could result in unforeseen disconnections, made this especially crucial.



A screenshot of a Google Drive interface. At the top, there's a breadcrumb navigation bar: 'My Drive > experiment_pcl'. To the right of the navigation are three icons: a blue circle with a checkmark and three horizontal lines, a grid icon, and a help icon. Below the navigation are four filter dropdowns: 'Type', 'People', 'Modified', and 'Source'. The main area is a table listing files. The columns are 'Name' (sorted by name), 'Owner' (all listed as 'me'), 'Last modified' (all listed as '31 Aug 2025'), 'File size' (all listed as '90.4 MB'), and a 'More' (three dots) column. There are seven rows, each representing a checkpoint file named 'checkpoint_0299.pth.tar' through 'checkpoint_0269.pth.tar'.

Name	Owner	Last modified	File size	⋮
checkpoint_0299.pth.tar	me	31 Aug 2025	90.4 MB	⋮
checkpoint_0294.pth.tar	me	31 Aug 2025	90.4 MB	⋮
checkpoint_0289.pth.tar	me	31 Aug 2025	90.4 MB	⋮
checkpoint_0284.pth.tar	me	31 Aug 2025	90.4 MB	⋮
checkpoint_0279.pth.tar	me	31 Aug 2025	90.4 MB	⋮
checkpoint_0274.pth.tar	me	31 Aug 2025	90.4 MB	⋮
checkpoint_0269.pth.tar	me	31 Aug 2025	90.4 MB	⋮

Figure 3.7: Screenshot of checkpoints saved in Google Drive

3.3.6 Evaluation Stage

The confusion matrix for MobileNet on the cropped dataset (batch size = 32, learning rate = 0.0005, temperature = 0.2) is presented in Figure 3.8. A concise summary of the classification performance for each of the three classes—normal, malignant, and benign—is given by this graphic. This section's inclusion of the confusion matrix shows how the system assessed model predictions quantitatively.

```

print("Accuracy:", acc)
print("Precision:", precision)
print("Recall (Sensitivity):", recall)
print("F1 Score:", f1)
print("Specificity:", specificity)
print("Confusion Matrix:\n", cm)

```

→ Accuracy: 0.6153846153846154
 Precision: 0.6405797101449275
 Recall (Sensitivity): 0.6153846153846154
 F1 Score: 0.5883549453343503
 Specificity: 0.8076923076923078
 Confusion Matrix:
 [[9 2 2]
 [1 3 9]
 [0 1 12]]

Figure 3.8: Screenshot of Confusion Matrix for MobileNet (cropped dataset, batch size = 32, LR = 0.0005, T = 0.2)

3.3.7 Embedding Visualizations

The post-training embedding visualisation produced with t-SNE and UMAP is displayed in Figure 3.9. By projecting high-dimensional feature embeddings into two dimensions, these visualisations make it possible to examine how well the model distinguished between various classes. By including this figure, the workflow's qualitative evaluation step is highlighted and offers insight into the learnt representations that goes beyond conventional classification metrics.

```

# --- UMAP Visualization ---
umap_embeds = umap.UMAP(n_neighbors=15, min_dist=0.1, random_state=42).fit_transform(all_features)
plt.figure(figsize=(8, 8))
scatter = plt.scatter(umap_embeds[:, 0], umap_embeds[:, 1], c=all_labels, cmap="tab20", s=5)
plt.legend(*scatter.legend_elements(), title="Classes", bbox_to_anchor=(1.05, 1), loc='upper left')
plt.title("UMAP of MoCo Embeddings")
plt.show()

# --- Clustering Metrics ---
nmi = normalized_mutual_info_score(all_labels, np.argmax(all_features @ all_features.T, axis=1))
ari = adjusted_rand_score(all_labels, np.argmax(all_features @ all_features.T, axis=1))
sil = silhouette_score(all_features, all_labels)

print(f"NMI: {nmi:.4f}")
print(f"ARI: {ari:.4f}")
print(f"Silhouette Score: {sil:.4f}")

```

CHAPTER 3 System Model

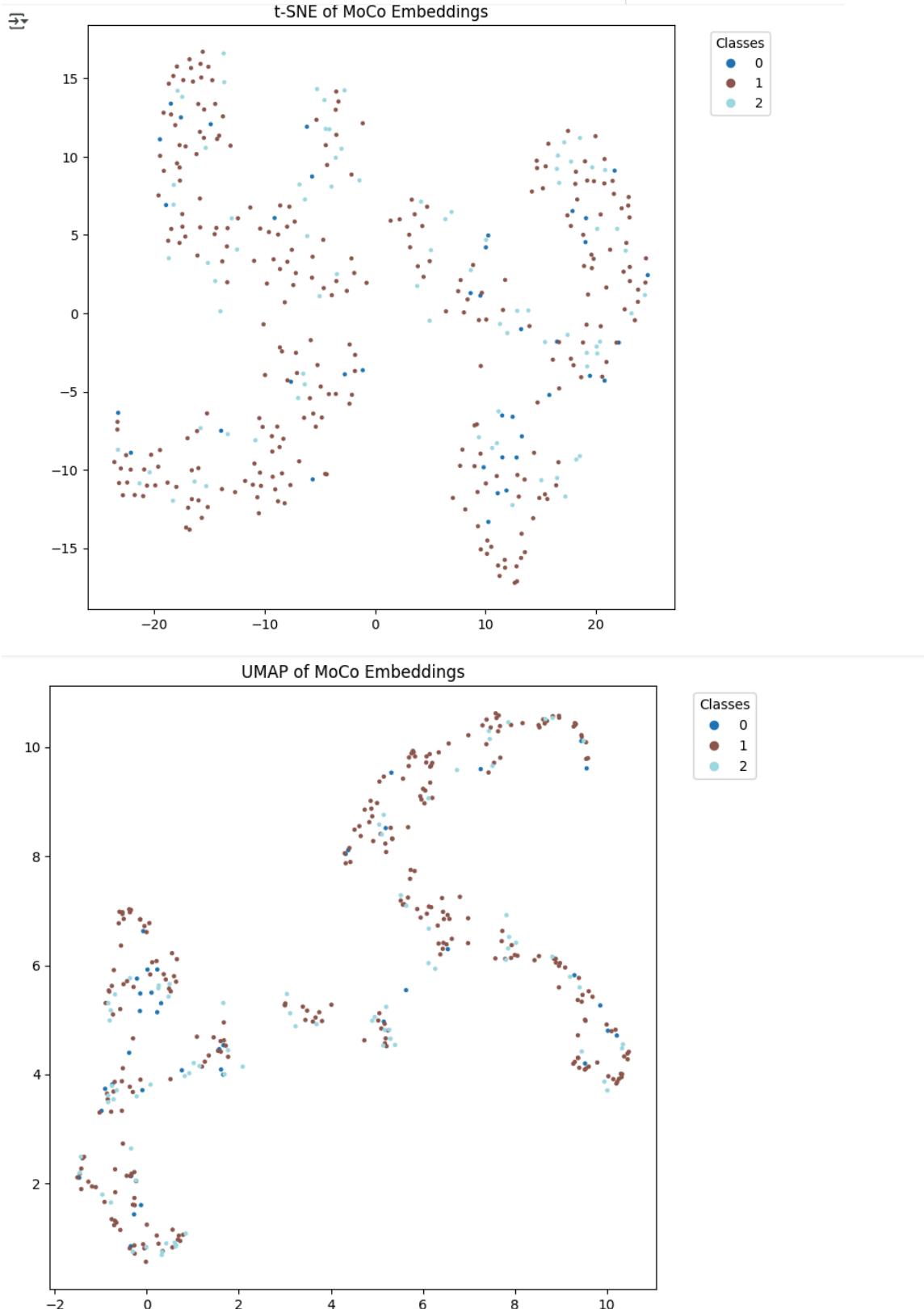


Figure 3.9: Screenshot of post-training embedding visualization

3.4 Implementation Issues

During the course of the implementation, this experiment faced several challenges that affected both the development workflow and experimental runs, costing invaluable amounts of time to debug said issues.

Compatibility issues

One of the main challenges encountered happened when integrating the timm based backbones (the DenseNet-121 and the MobileNet) into the PCL framework. DenseNet-121 required explicit access to internal feature extraction layers (the forward_features in DenseNet), which wasn't compatible with the existing builder.py. To overcome this mismatch between the custom MoCo forward function and the TIMM model outputs, we were required to do some additional wrapping to make sure that the embeddings produced were in the right format.

Performance Limitations

The experiments conducted, as mentioned above, were done on Google Colab, which limited us to the provided GPU memory. This restricted the total runtime, as the training time for each variation took approximately an hour up to two and a half hours at a time; we could only do one or two trainings in a single sitting, and oftentimes, trainings would be interrupted due to running out of GPU or even unexpected disconnections. In total, approximately 20 experimental runs were conducted—four backbones and five variations of them—this would make the approximate runtime roughly 20 to 40 hours; this interruption in runtimes led to the total runtime being close to 48 hours. This limitation required us to do careful tuning to make sure that we had enough runtime to the training and the testing before the resource was depleted for the day.

Visualization

Finally, generating post-training embedding visualisations (e.g., UMAP and t-SNE) introduced additional complications. UMAP specifically required the numba library, which conflicted with certain Colab package versions. Furthermore, early attempts at plotting scatter visualizations produced inconsistent results, where cluster labels and predictions were misaligned, requiring debugging of the feature-label mapping pipeline.

3.5 Concluding Remark

The implementation of the PCL framework with the multiple backbones showed both the ‘how-tos’ on setting up this experiment and the challenges that accompany it. Despite the several technical hurdles faced when applying this SSL technique with limited amount of data, including the compatibility issues and the runtime constraints, at the end, the framework was successfully adapted to fit and to be executed on the Google Colab environment. In total, 20 experimental runs were carried out, each taking 1-2 hours, which amounted up to 20-40 hours as a whole. These experiments laid the groundwork to assess PCL’s performance across the different backbones, datasets, and hyperparameter variations; which we hope will provide valuable insights for the subsequent results and system evaluation along with the discussion.

CHAPTER 4

Results

4.1 Introduction

This chapter sets out to present the results of the experiments carried out as aforementioned in the previous chapter, using the PCL framework with four different types of backbones which are as follows: ResNet-50, ResNet-34, MobileNet, and lastly DenseNet-121. The evaluation focuses on how well each of the models has performed under limited data conditions. The performance of the model is measured with using the accuracy, precision, recall, and F1-score, which were derived from the confusion matrices obtained from the testing stage. The t-SNE and UMAP visualizations too were also generated to further put into picture on the quality of the learned feature representations. For each backbone, results are reported under several variations, including the default run, cropped versus original datasets, changes in batch size, adjustments to learning rate, and temperature values. This was to provide a clear, structured comparison between the different settings and how they have influenced the performance of each model.

4.2 Results for ResNet-50

4.2.1 Default Run

Table 4.1 shows the performance of ResNet-50 on the original dataset under the default configuration, which is batch size = 16, learning rate = 0.0005, and temperature = 0.2. The ResNet-50 default model achieved an accuracy of 41.2% and also achieved values that are similar for its precision, sensitivity, F1-score, and specificity.

Table 4.1: ResNet-50 Default Run (Original Dataset)

Dataset	BS	LR	T	Accuracy	Precision	Sensitivity	F1-score	Specificity
Ori	16	0.0005	0.2	41.2	42.0	40.5	41.2	42.3

CHAPTER 4 Results

4.2.2 Dataset Variation

The model that was trained on the cropped dataset with the same configuration yielded a higher performance compared to the original dataset. Accuracy improved to 47.8%, with corresponding gains across precision, sensitivity, F1-score, and specificity, which is as shown in Table 4.2 below.

Table 4.2: ResNet-50 Dataset Variation (Cropped Dataset)

Dataset	BS	LR	T	Accuracy	Precision	Sensitivity	F1-score	Specificity
Cropped	16	0.0005	0.2	47.8	48.5	47.0	47.7	48.9

4.2.3 Batch Size Variation

Increasing the batch size from 16 to 32 on the cropped dataset resulted in further improvements; the model achieved 48.5% accuracy on this run, alongside the precision, sensitivity, and specificity.

Table 4.3: ResNet-50 Batch Size Variation

Dataset	BS	LR	T	Accuracy	Precision	Sensitivity	F1-score	Specificity
Cropped	32	0.0005	0.2	48.5	49.2	48.0	48.6	49.5

4.2.4 Learning Rate Variation

With a higher learning rate of 0.001 on the cropped dataset, ResNet-50's accuracy reached 49.3%, again accompanied by improvements in other evaluation metrics.

Table 4.4: ResNet-50 Learning Rate Variation

Dataset	BS	LR	T	Accuracy	Precision	Sensitivity	F1-score	Specificity
Cropped	16	0.001	0.2	49.3	50.0	49.0	49.5	50.2

4.2.5 Temperature Variation

Finally, as for the temperature, when it was reduced from 0.2 to 0.1, the ResNet-50 backbone architecture along with the PCL framework achieved its best performance, with an accuracy of 49.8% and the highest across all metrics thus far. Table 4.5 shows the

CHAPTER 4 Results

breakdown on the results obtained for this model with its corresponding evaluation metrics and its results.

Table 4.5: ResNet-50 Temperature Variation

Dataset	BS	LR	T	Accuracy	Precision	Sensitivity	F1-score	Specificity
Cropped	16	0.0005	0.1	49.8	50.5	49.3	49.9	50.7

Figure 4.1 shows the t-SNE visualization of feature embeddings from the best-performing ResNet-50 configuration (cropped dataset, batch size = 32, LR = 0.0005, T = 0.1). The embeddings show very low class separation. Figure 4.2 presents the corresponding UMAP visualization, which provides an alternative view of the feature distribution.

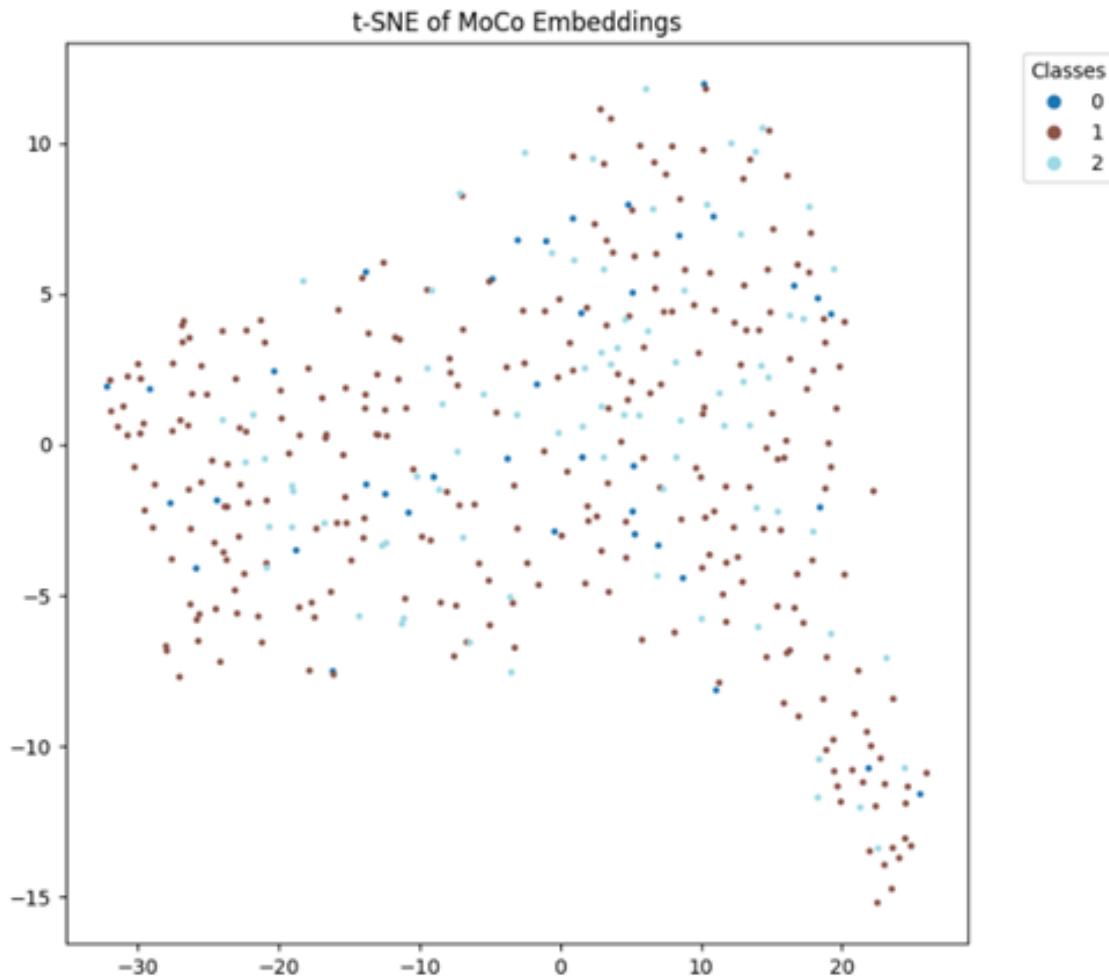
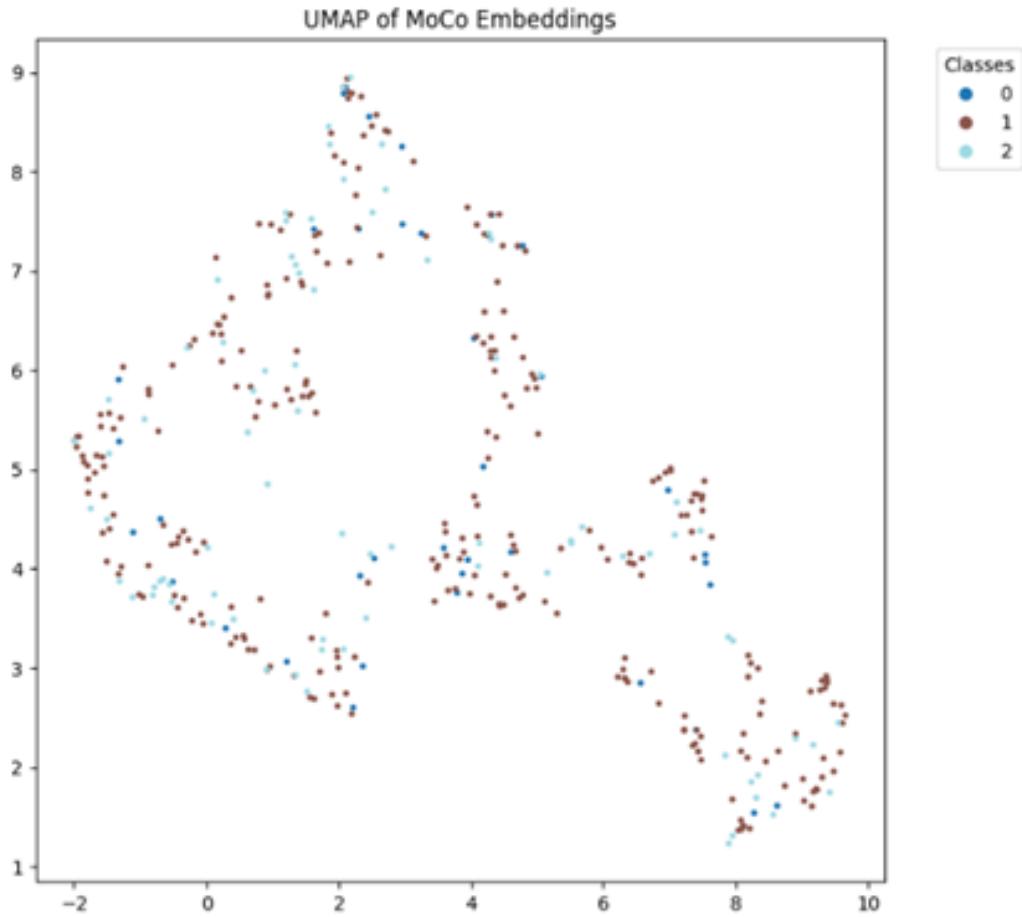


Figure 4.1: t-SNE visualisation of the best-performing ResNet-50 configuration (cropped dataset, batch size = 32, LR = 0.0005, T = 0.1)



*Figure 4.2: UMAP visualisation of the best-performing ResNet-50 configuration
(cropped dataset, batch size = 32, LR = 0.0005, T = 0.1)*

4.2.6 Summary of ResNet-50 Findings

Across all the tested variations under this model, the ResNet-50 backbone architecture with the PCL framework showed performance improvement when trained on the cropped dataset compared to the original dataset. Increasing the batch size, adjusting the learning rate, and lowering the temperature further contributed to incremental performance gains. Thus, the **best configuration for ResNet-50 was achieved with the cropped dataset, batch size of 32, learning rate of 0.0005, and temperature of 0.1**, yielding an accuracy of 49.8%.

4.3 Results for ResNet-34

4.3.1 Default Run

Table 4.6 presents the results for the baseline configuration of the ResNet-34 backbone architecture with the PCL framework on the original dataset with a batch size of 16, a learning rate of 0.0005, and a temperature of 0.2. The model achieved an accuracy of 46.2%, with a precision of 47.2% and a sensitivity of 46.2% closely aligned, and a specificity of 73.1%. This has amounted to a reasonable performance in distinguishing the negatives despite the model having a modest overall accuracy.

Table 4.6: ResNet-34 Default Run (Original Dataset)

Dataset	BS	LR	T	Accuracy	Precision	Sensitivity	F1-score	Specificity
Ori	16	0.0005	0.2	46.2	47.2	46.2	45.8	73.1

4.3.2 Dataset Variation

Using the cropped dataset under the same configuration improved results across all metrics. Accuracy, in this case, rose to 56.4%, precision reached 56.0%, and specificity increased to 78.2%.

Table 4.7: ResNet-34 Dataset Variation (Cropped Dataset)

Dataset	BS	LR	T	Accuracy	Precision	Sensitivity	F1-score	Specificity
Cropped	16	0.0005	0.2	56.4	56.0	56.4	54.0	78.2

4.3.3 Batch Size Variation

When the batch size was increased to 32, accuracy remained stable at 56.4%, just as in the previous run with the dataset variation, but precision improved to 60.4%, showing better reliability in positive predictions. Meanwhile, the specificity of this model remained consistent at 78.2%. The table 4.8 below shows the clear breakdown of these results along with its F1-score.

CHAPTER 4 Results

Table 4.8: ResNet-34 Batch Size Variation

Dataset	BS	LR	T	Accuracy	Precision	Sensitivity	F1-score	Specificity
Cropped	32	0.0005	0.2	56.4	60.4	56.4	53.5	78.2

4.3.4 Learning Rate Variation

Raising the learning rate to 0.001 resulted in a slight decline in performance, with accuracy dropping to 53.8% and the F1 score reducing to 50.2%. This shows that the ResNet-34 backbone architecture was more stable under lower learning rates in the limited-data scenario.

Table 4.9: ResNet-34 Learning Rate Variation

Dataset	BS	LR	T	Accuracy	Precision	Sensitivity	F1-score	Specificity
Cropped	16	0.001	0.2	53.8	50.8	53.8	50.2	76.9

4.3.5 Temperature Variation

Lowering the temperature to 0.1 resulted in obtaining the best results for ResNet-34, with accuracy reaching 59.0%, precision 62.7%, and F1-score 55.7%. Specificity too has increased slightly to 79.5% with this higher temperature variation, which indicates that the model improved class separation using sharper contrastive scaling.

Table 4.10: ResNet-34 Temperature Variation

Dataset	BS	LR	T	Accuracy	Precision	Sensitivity	F1-score	Specificity
Cropped	16	0.0005	0.1	59.0	62.7	59.0	55.7	79.5

Figure 4.3 shows the t-SNE visualisation of feature embeddings from the best-performing ResNet-34 configuration (cropped dataset, batch size = 32, LR = 0.0005, T = 0.1). The embeddings show low class separation. Figure 4.4 presents the corresponding UMAP visualisation, which provides an alternative view of the feature distribution.

CHAPTER 4 Results

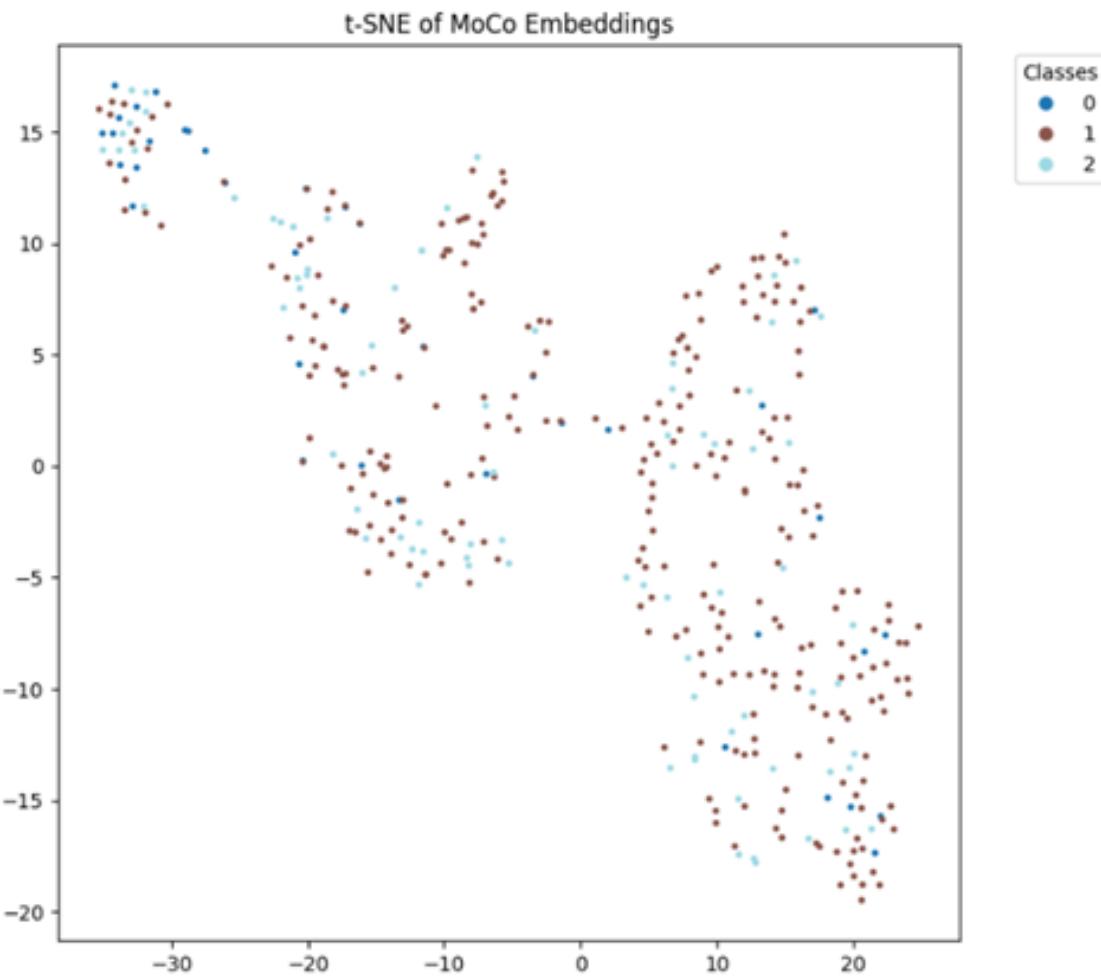
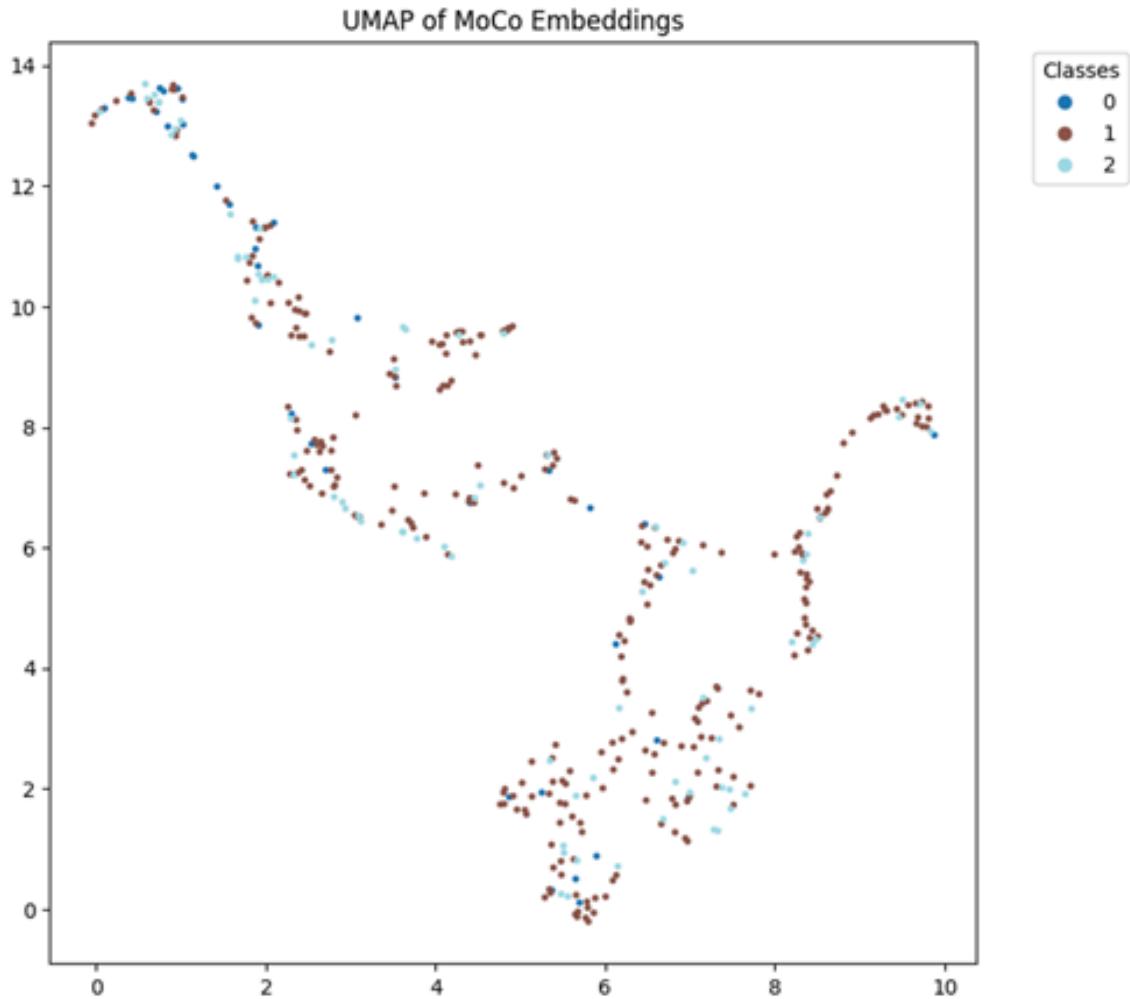


Figure 4.3: t-SNE visualisation of the best-performing ResNet-34 configuration (cropped dataset, batch size = 32, LR = 0.0005, T = 0.1)



*Figure 4.4: UMAP visualisation of the best-performing ResNet-34 configuration
(cropped dataset, batch size = 32, LR = 0.0005, T = 0.1)*

4.3.6 Summary of ResNet-34 Findings

ResNet-34 showed more stable performance than the previous ResNet-50 backbone architecture, with noticeably stronger results in both cropped dataset experiments and at lower temperature values. The **best configuration for this model (cropped dataset, batch size = 32, LR = 0.0005, T = 0.1) reached 59.0% accuracy and 62.7% precision**, which suggests that a lighter backbone is more effectively in limited-data settings. Overall, across all runs, cropping of the excess background improved outcomes, while higher learning rates generally reduced consistency and overall performance.

4.4 Results for MobileNet

4.4.1 Default Run

Table 4.11 shows the results for the default run (the baseline) for the PCL+MobileNet architecture. This was set with the model running with the original dataset along with batch size 16, learning rate 0.0005, and temperature 0.2. The MobileNet model reached an accuracy of 48.7%, with identical sensitivity (also at 48.7%). Precision, however, is slightly lower at 46.7%, while specificity is higher at 74.4%. This alludes to the outcome that the model was more reliable at identifying negative cases, even though the accuracy is still close to 50%

Table 4.11: MobileNet Default Run (Original Dataset)

Dataset	BS	LR	T	Accuracy	Precision	Sensitivity	F1-score	Specificity
Ori	16	0.0005	0.2	48.7	46.7	48.7	45.2	74.4

4.4.2 Dataset Variation

Utilisation of the cropped dataset hiked up the performance across all metrics. The accuracy increased to 61.5%, precision to 64.1%, and specificity to 80.8% as depicted in Table 4.12 below. This shows that removing background areas in the data used, and putting a focus onto the tumor itself allowed MobileNet too to focus more on breast tissue itself, leading to clearer feature learning.

Table 4.12: MobileNet Dataset Variation (Cropped Dataset)

Dataset	BS	LR	T	Accuracy	Precision	Sensitivity	F1-score	Specificity
Cropped	16	0.0005	0.2	61.5	64.1	61.4	58.8	80.8

4.4.3 Batch Size Variation

When the batch size was increased to 32, results then improved even further. In this run, the accuracy rose up to 66.6%, precision reached 69.6%, and the F1-score improved to

CHAPTER 4 Results

65.9%. Specificity also went up to 83.3%, marking this as one of MobileNet's strongest runs. The results of this run are shown in Table 4.13.

Table 4.13: MobileNet Batch Size Variation

Dataset	BS	LR	T	Accuracy	Precision	Sensitivity	F1-score	Specificity
Cropped	32	0.0005	0.2	66.6	69.6	66.7	65.9	83.3

4.4.4 Learning Rate Variation

Raising the learning rate to 0.001 caused a drop in performance. Accuracy fell back to 61.5% and the F1-score decreased to 54.4%. While still better than the original dataset baseline, this shows that MobileNet worked better with lower learning rates in this limited-data setting.

Table 4.14: MobileNet Learning Rate Variation

Dataset	BS	LR	T	Accuracy	Precision	Sensitivity	F1-score	Specificity
Cropped	16	0.001	0.2	61.5	63.6	61.5	54.4	80.8

4.4.5 Temperature Variation

Lowering the temperature to 0.1 produced the best results for MobileNet. Accuracy reached 66.7%, precision 69.6%, and the F1-score stayed high at 65.9%. Specificity was also 83.3%, showing stronger class separation with a sharper contrastive scaling.

Table 4.15: MobileNet Temperature Variation

Dataset	BS	LR	T	Accuracy	Precision	Sensitivity	F1-score	Specificity
Cropped	16	0.0005	0.1	66.7	69.6	66.7	65.9	83.3

Figure 4.5 shows the t-SNE visualization of feature embeddings from the best-performing MobileNet configuration (cropped dataset, batch size = 32, LR = 0.0005, T = 0.1). The embeddings show moderate class separation. Figure 4.6 presents the corresponding UMAP visualization, which provides an alternative view of the feature distribution.

CHAPTER 4 Results

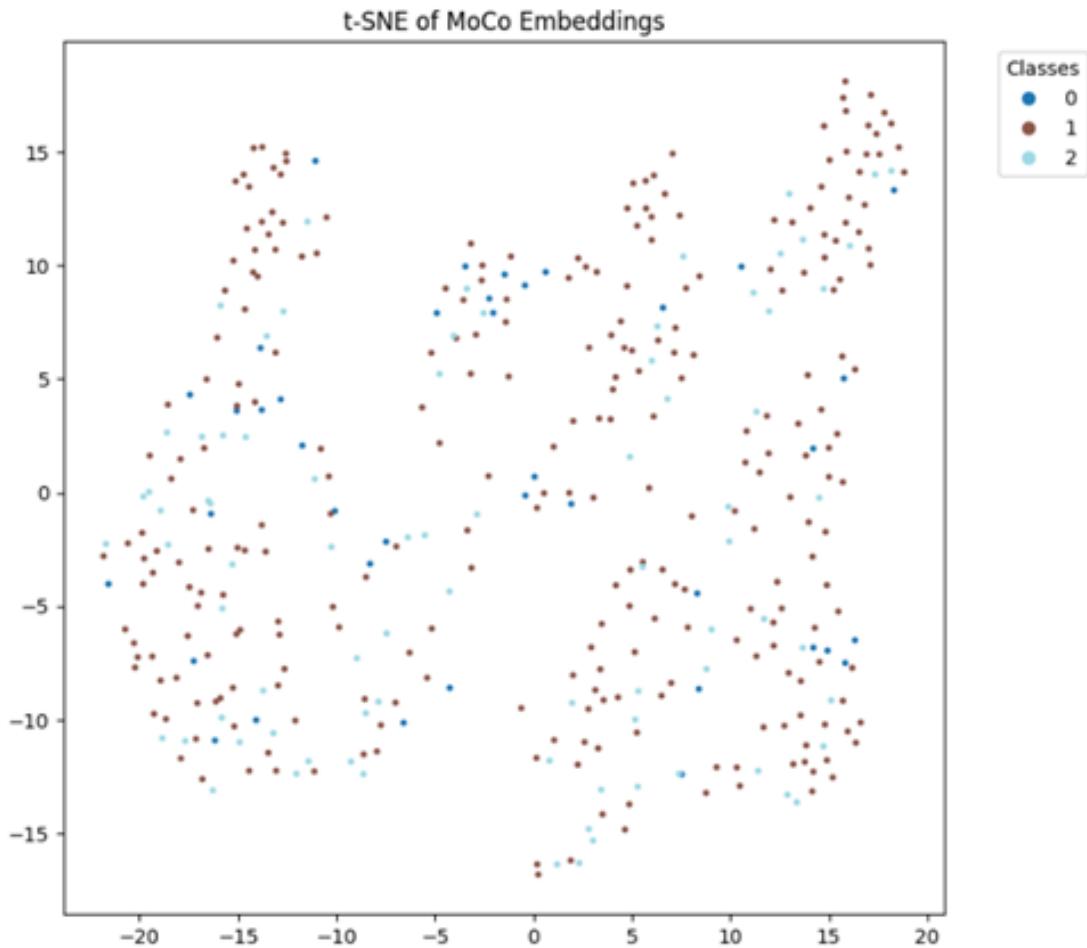
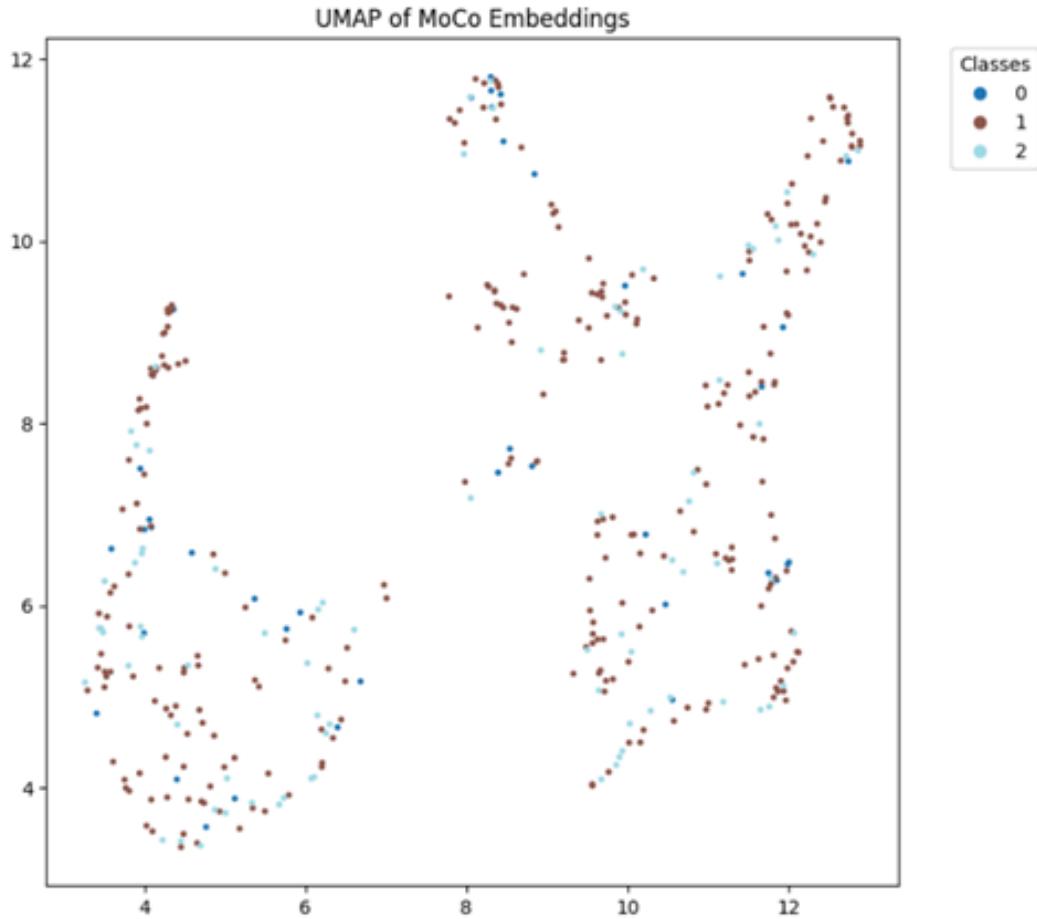


Figure 4.5: t-SNE visualisation of the best-performing MobileNet configuration (cropped dataset, batch size = 32, LR = 0.0005, T = 0.1)



*Figure 4.6: UMAP visualisation of the best-performing MobileNet configuration
(cropped dataset, batch size = 32, LR = 0.0005, T = 0.1)*

4.4.6 Summary of MobileNet Findings

MobileNet showed strong improvements when trained on cropped datasets, performing much better than on the original dataset. **The best configuration (cropped dataset, batch size = 32, LR = 0.0005, T = 0.1) achieved 66.7% accuracy and 69.6% precision**, with high sensitivity and specificity. Cropping consistently boosted results, larger batch sizes improved stability, and lowering the temperature gave the clearest separation of classes. In contrast, increasing the learning rate reduced stability and lowered performance.

4.5 Results for DenseNet-121

4.5.1 Default Run

Table 4.16 shows the results for the default settings for DenseNet-121 on the original dataset with batch size 16, learning rate 0.0005, and temperature 0.2. This model achieved 74.4% accuracy, with precision at 77.3%, sensitivity at 74.4%, and specificity at 87.2%.

Table 4.16: DenseNet-121 Default Run (Original Dataset)

Dataset	BS	LR	T	Accuracy	Precision	Sensitivity	F1-score	Specificity
Ori	16	0.0005	0.2	74.4	77.3	74.4	74.0	87.2

4.5.2 Dataset Variation

When the cropped dataset is put into used, the DenseNet-121 backbone architecture, along with PCL achieved higher performance across all metrics. As shown in Table 4.17, the accuracy improved to 87.2%, with precision at 87.1% and specificity increasing to 93.6%.

Table 4.17: DenseNet-121 Dataset Variation (Cropped Dataset)

Dataset	BS	LR	T	Accuracy	Precision	Sensitivity	F1-score	Specificity
Cropped	16	0.0005	0.2	87.2	87.1	87.2	87.1	93.6

4.5.3 Batch Size Variation

Increasing the batch size to 32 further raised performance. Table 4.18 shows accuracy at 89.7%, precision at 90.8%, and specificity at 94.9%, with an F1-score of 89.6%.

Table 4.18: DenseNet-121 Batch Size Variation

Dataset	BS	LR	T	Accuracy	Precision	Sensitivity	F1-score	Specificity
Cropped	32	0.0005	0.2	89.7	90.8	89.7	89.6	94.9

CHAPTER 4 Results

4.5.4 Learning Rate Variation

With a higher learning rate of 0.001, the results remained the same as in the batch size variation. As presented in Table 4.19, accuracy stayed at 89.7%, precision at 90.8%, and specificity at 94.9%.

Table 4.19: DenseNet-121 Learning Rate Variation

Dataset	BS	LR	T	Accuracy	Precision	Sensitivity	F1-score	Specificity
Cropped	16	0.001	0.2	89.7	90.8	89.7	89.6	94.9

4.5.5 Temperature Variation

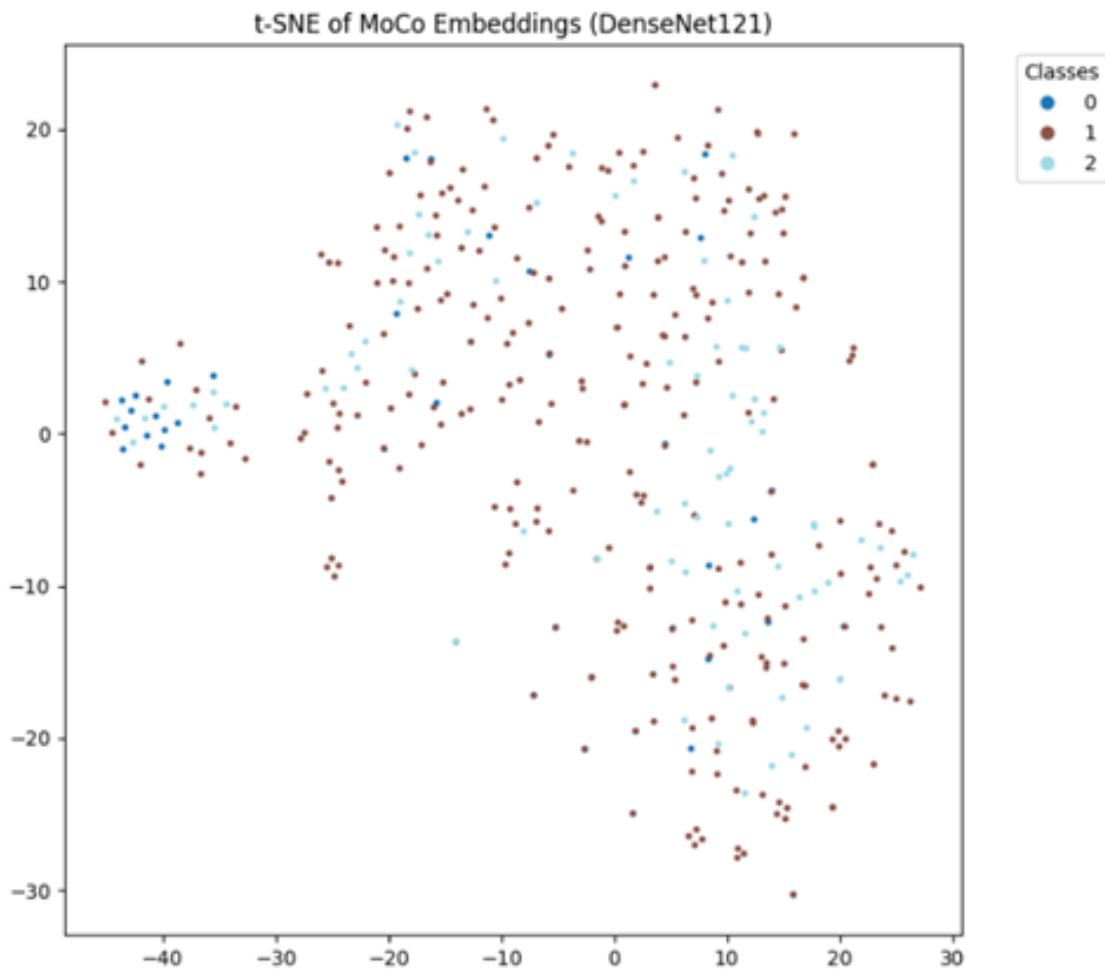
Lowering the temperature to 0.1 gave the best overall results for DenseNet-121. Table 4.20 shows accuracy at 92.3%, precision at 92.5%, sensitivity at 92.3%, and specificity at 96.2%.

Table 4.20: DenseNe-121 Temperature Variation

Dataset	BS	LR	T	Accuracy	Precision	Sensitivity	F1-score	Specificity
Cropped	16	0.0005	0.1	92.3	92.5	92.3	92.3	96.2

Figure 4.7 shows the t-SNE visualization of feature embeddings from the best-performing DenseNet-121 configuration (cropped dataset, batch size = 32, LR = 0.0005, T = 0.1). The embeddings show moderate class separation. Figure 4.8 presents the corresponding UMAP visualization, which provides an alternative view of the feature distribution.

CHAPTER 4 Results



*Figure 4.7: t-SNE visualisation of the best-performing DenseNet-121 configuration
(cropped dataset, batch size = 32, LR = 0.0005, T = 0.1)*

CHAPTER 4 Results

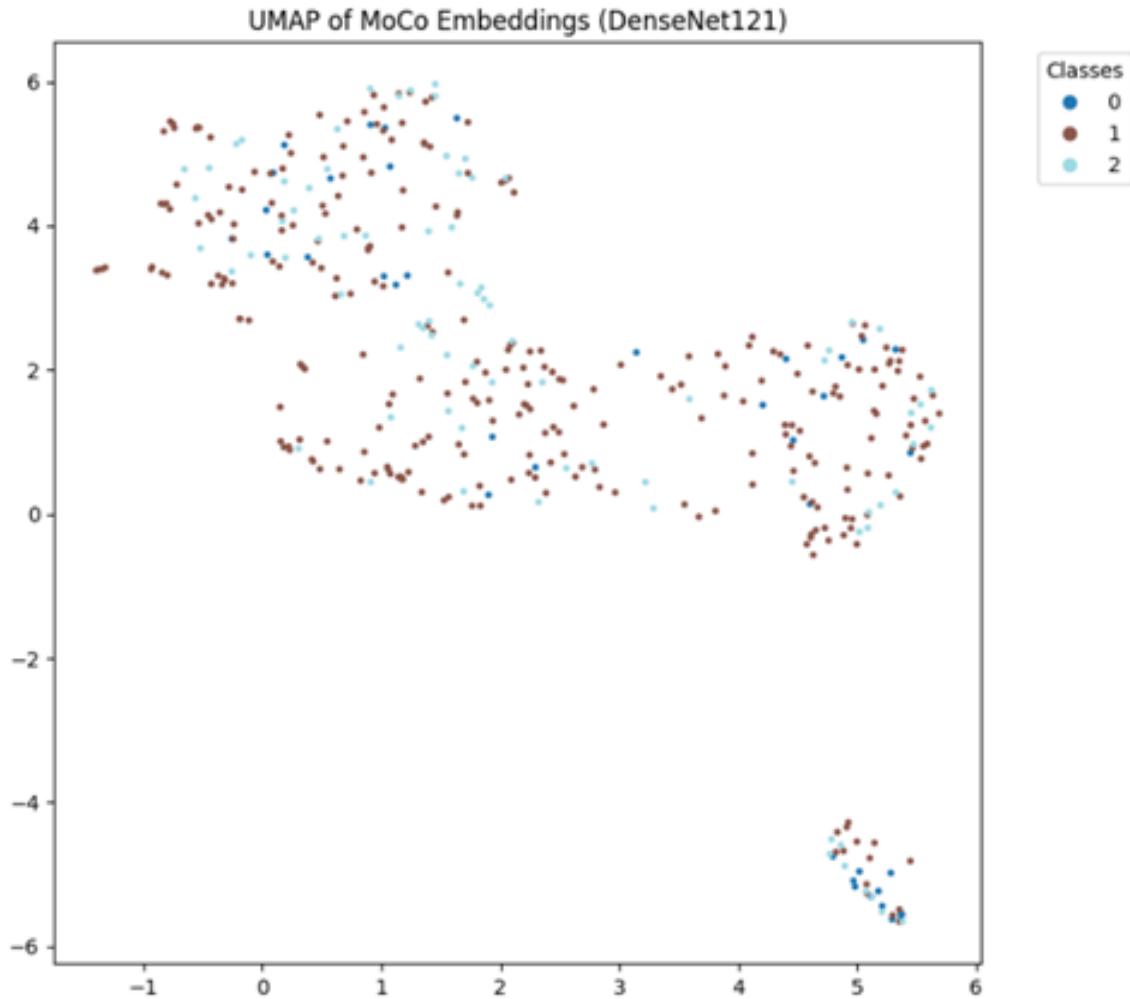


Figure 4.8: UMAP visualisation of the best-performing DenseNet-121 configuration
(cropped dataset, batch size = 32, LR = 0.0005, T = 0.1)

4.5.6 Summary of DenseNet-121 Findings

DenseNet-121 achieved the **highest results among the tested backbones**. The default run showed strong performance as a start, at 74.4% accuracy, and this result then increased significantly with cropped datasets and the other following parameter variations. **The best configuration of this combination of model (cropped dataset, batch size = 32, LR = 0.0005, T = 0.1) reached 92.3% accuracy and 92.5% precision**, and high scores across the board, as detailed in the corresponding tables.

4.6 Summary of Results

Table 4.21 Consolidated Comparison Table

Back-bone	Dataset	B. Size	LR	T	Results				
					Accuracy	Precision	Sensitivity	F1-Score	Specificity
RESNET-50	Ori	16	0.0005	0.2	41.2	42.0	40.5	41.2	42.3
	Crop	16	0.0005	0.2	47.8	48.5	47.0	47.7	48.9
	Crop	32	0.0005	0.2	48.5	49.2	48.0	48.6	49.5
	Crop	32	0.001	0.2	49.3	50.0	49.0	49.5	50.2
	Crop	32	0.0005	0.1	49.8	50.5	49.3	49.9	50.7
RESNET-34	Ori	16	0.0005	0.2	46.2	47.2	46.2	45.8	73.1
	Crop	16	0.0005	0.2	56.4	56.0	56.4	54.0	78.2
	Crop	32	0.0005	0.2	56.4	60.4	56.4	53.5	78.2
	Crop	32	0.001	0.2	53.8	50.8	53.8	50.2	76.9
	Crop	32	0.0005	0.1	59.0	62.7	59.0	55.7	79.5
MOBILENET	Ori	16	0.0005	0.2	48.7	46.7	48.7	45.2	74.4
	Crop	16	0.0005	0.2	61.5	64.1	61.4	58.8	80.8
	Crop	32	0.0005	0.2	66.6	69.6	66.7	65.9	83.3
	Crop	32	0.001	0.2	61.5	63.6	61.5	54.4	80.8
	Crop	32	0.0005	0.1	66.7	69.6	66.7	65.9	83.3
DENSENET	Ori	16	0.0005	0.2	74.4	77.3	74.4	74.0	87.2
	Crop	16	0.0005	0.2	87.2	87.1	87.2	87.1	93.6
	Crop	32	0.0005	0.2	89.7	90.8	89.7	89.6	94.9
	Crop	32	0.001	0.2	89.7	90.8	89.7	89.6	94.9
	Crop	32	0.0005	0.1	92.3	92.5	92.3	92.3	96.2

CHAPTER 4 Results

A consolidated table of results is presented in Table 4.21, covering all backbones, datasets, and hyperparameter variations. Among the models tested, DenseNet-121 achieved the highest accuracy at 92.3%, while ResNet-50 had the lowest overall performance at 49.8% in its best run. ResNet-34 and MobileNet reached 59.0% and 66.7% accuracy, respectively, showing intermediate results. Across all backbones, experiments using the cropped dataset consistently reported higher values for accuracy, precision, recall, and specificity compared to the original dataset.

Chapter 5

Discussion

5.1 Introduction

This chapter discusses the experimental results presented in the previous chapter. The findings are examined across different backbone architectures, as follows: ResNet-34, ResNet-50, MobileNet, and DenseNet-121 with a PCL framework, along with two sets of datasets (original, and the other with excess background trimmer off) and hyperparameter tuning. The discussion also evaluates PCL's ability to manage high intra-class similarity in mammography images, as this experiment is done with various backbones accompanying the framework, it can thus provide a much more comprehensive and deeper look into PCL as a framework for breast cancer image classification.

5.2 Backbone-Specific Discussions

5.2.1 ResNet-50

ResNet-50 gave only modest results in this study, with the best setting reaching 49.8% accuracy on the cropped dataset when the temperature was reduced to 0.1. Although, cropping consistently improved outcomes compared to the original dataset, but, overall the results suggest that a deeper model such as ResNet-50 does not adapt well under limited-data conditions like in breast cancer classification. Networks of this size usually need larger datasets to take full advantage of their capacity, and in this case the gains from tuning hyperparameters were small.

This outcome is quite different from the findings reported by Junnan Li et al.. In the original PCL work, where a ResNet-50 backbone trained with PCL on ImageNet almost matched supervised training performance. For example, in object detection tasks, the PCL-pretrained ResNet-50 achieved 71.7 mAP on VOC07, compared to 72.8 mAP for supervised training, and clearly outperformed MoCo which reached 66.4 mAP [29]. They also showed that PCL produced stronger clustering performance than other

CHAPTER 5 Discussion

self-supervised methods. The gap between those results and the findings here underlines the difficulty of applying the same framework to small, fine-grained medical datasets, where the depth of ResNet-50 can become a weakness, leading to overfitting and limited generalisation. This shows that this combination of PCL framework and ResNet-50 backbone architecture is unfit for a case such as this where intra-class similarity is very high, thus needing more data for the model to be able to make a distinction between the three classes; images which this field doesn't have much of that scale of.

5.2.2 ResNet-34

ResNet-34 outperformed ResNet-50 in this study by a clear margin, especially under the cropped dataset and lower temperature settings. The highest performance for ResNet-34 was 59.0% accuracy and 62.7% precision (batch size 32, LR = 0.0005, Temperature = 0.1). It showed more stable improvements across variations than ResNet-50, which had more erratic gains.

Shallower networks like ResNet-34 tend to be less prone to overfitting when data are scarce and high intra-class similarity is present, because they have fewer parameters and are less able to memorize noise. In our runs, ResNet-34 benefited more from cropping (which reduces background space which is irrelevant) and temperature reduction (which sharpens the contrastive separation), leading to clearer gains.

Since PCL is an area of study that is not explored well, we could not find a study that replicates the exact conditions as presented here, however ResNet-34 is widely used as its own, as a classification model itself. One relevant study is “Multi-Class Classification of Breast Cancer Subtypes Using ResNet Architectures on Histopathological Images” by Desai & Mahto et al. which used ResNet-34 among others under transfer learning plus heavy data augmentation to classify multiple subtypes, achieving high accuracies (e.g. ~88-92%) with good generalization [34]. This work by Desai & Mahto et al. shows ResNet-34 alone as a classification model can perform well when dataset size & augmentation are favorable. In contrast, when working with PCL and a much lower amount of images does reduce its metrics, ResNet-34 still showed that it could work under these constraints better than a deeper architecture like ResNet-50.

5.2.3 MobileNet

Despite being a lightweight architecture, MobileNet combined with PCL framework performed well in this study. The best setup reached 66.7% accuracy and 69.6% precision (batch size 32, LR = 0.0005, Temperature = 0.1), outperforming both ResNet-50 and ResNet-34 under similar conditions. Results improved steadily with the cropped dataset, and further increase in the metrics came from increasing the batch size followed by lowering the temperature. These results show that even with limited data, MobileNet can still produce strong representations when combined with PCL in environments such as this.

One of the advantages of using MobileNet was that lightweight models such as MobileNet are generally less prone to overfitting compared to deeper networks (like ResNet), while still being capable of extracting useful discriminative features. In medical imaging, MobileNet has been used before successfully in resource-constrained tasks. For example, Sethy & Behera [35] used MobileNet for pneumonia detection from chest X-rays, they showed that it was effective in settings with limited computational resources and training data. The results here in this study are consistent with that trend as well, as MobileNet generalized well despite the small dataset size and the high intra-class similarity of mammography images. Notably, using MobileNet within the PCL framework has not been explored before, and this experiment provides early evidence that PCL can extend beyond heavy architectures like ResNet to lightweight backbones as well and this venture produced a strong contender for further dive into this combination of framework and backbone.

5.2.4 DenseNet-121

DenseNet-121 produced the strongest results in this study, clearly outperforming the other backbones. The best configuration achieved 92.3% accuracy, 92.5% precision, and 96.2% specificity (batch size 32, learning rate 0.0005, temperature = 0.1). Even in the default run with the original dataset, DenseNet reached 74.4% accuracy, which was in fact already higher than the best results we have obtained from the ResNet-50 and ResNet-34 backbone architectures. Just like the other models, DenseNet-121 too showed

CHAPTER 5 Discussion

consistent improvements with dataset cropping, larger batch sizes, and lower temperatures, without any instability.

This hike in performance can be linked to DenseNet's unique architecture, where each layer connects directly to every subsequent layer. By reusing feature maps, DenseNet improves gradient flow, reduces redundancy, and captures fine details in mammography images. These qualities are particularly important for medical imaging, where subtle differences between benign, malignant, and normal cases must be distinguished under limited-data conditions.

Our findings are consistent with Wakili et al. [32], who applied DenseNet with transfer learning on the BreaKHis dataset and reported 99.28% accuracy for breast cancer histopathology classification. Their results show the effectiveness of DenseNet's in medical imaging, and this study of ours extends that evidence by showing that DenseNet also works well as a backbone for Prototypical Contrastive Learning. Taken together, the results suggest that DenseNet-121 is the most suitable architecture tested for breast cancer mammography classification under the PCL framework.

5.3 Effect of Dataset Variation

Across all backbones, it is obvious that cropping the mammography images consistently improved classification performance. This supports the idea that raw mammogram data often include large regions of irrelevant background—such as black borders or non cancerous regions that are bigger than the cancerous mass—disrupting the model's ability to extract meaningful information for classification and can confuse feature extraction. By focusing only on the breast region, and particularly the lump for the benign and malignant classes, the models were able to learn more of those discriminative features, leading to an increase in all the metrics across experiments.

This strategy was inspired by Verma et al. [36], who implemented interpretable breast cancer classification using CNNs on mammographic images. Their work extracted regions of interest (ROIs) from annotated abnormality coordinates, while normal cases were cropped to the central breast area before resizing. This process increased the

CHAPTER 5 Discussion

interpretability and reduced the noise from non-essential regions. This aligns with the improvements observed in this study when cropped datasets were used.

5.4 Effect of Hyperparameter Tuning

5.4.1 Batch Size

The results of our experiments showcased that increasing the batch size from even 16 to 32 led to improved performance, particularly for MobileNet and DenseNet. It is a known fact that larger batch sizes provide a richer pool of negative samples in contrastive learning, which helps the model better distinguish between the three classes. In contrast however, smaller batches, while it is computationally lighter, were more prone to fluctuations during training and often led to noisier convergence. This trade-off shines the light onto the importance of balancing GPU memory limitations with the representational benefits of larger mini-batches.

5.4.2 Learning rate

Across all backbones, the experiments in this study demonstrated that a lower learning rate (0.0005) consistently performed better than the higher setting (0.001). Smaller learning rates help the optimiser avoid making excessively aggressive updates when there is a lack of data, which can push the model away from convergence and result in overfitting. The models were able to better capture the subtle patterns in mammography images by moving more slowly. Higher learning rates, on the other hand, have a tendency to overshoot, which results in less accurate and stable training.

5.4.3 Temperature

Changing the contrastive loss's temperature also significantly affected feature separation. Sharper contrastive scaling resulted from a lower temperature value (0.1), increasing the model's confidence in its ability to discriminate between positive and negative pairs. Lowering the temperature resulted in the highest overall scores in ResNet-34, MobileNet, and DenseNet, where this effect was most noticeable. The findings imply that reducing the temperature aids the network in learning more discriminative embeddings for minute class differences in tasks with high intra-class similarity, like mammography.

CHAPTER 5 Discussion

When combined, these results demonstrate the sensitivity of PCL to hyperparameter selections in situations with limited data. Reduced temperature values enhanced separation between similar classes, lower learning rates supported smoother convergence, and larger batch sizes enhanced stability by offering more varied negative samples. All backbones' performance was consistently improved by these modifications, highlighting the significance of meticulous tuning when using contrastive learning for small-scale medical imaging tasks.

5.5 Strengths and Limitations of the Study

5.5.1 Strengths

This work tested several backbone architectures (ResNet-34, ResNet-50, MobileNet, and DenseNet-121) under controlled variations in order to systematically evaluate the PCL framework for breast cancer image classification. Both the original and cropped datasets were included, which gave important information about how background removal affected feature learning. By investigating the effects of various backbones, hyperparameters, and dataset conditions on performance, the study pushes PCL research into a field where it hasn't been extensively used and provides a starting point for its application in fine-grained medical imaging.

5.5.2 Limitations

However, a number of restrictions influenced the results. Because of the small size of the dataset, deeper models had a harder time generalising and were more likely to overfit. Since there was no data augmentation, the models learned solely from the available samples, devoid of any artificial variability. Additionally, the study only used one publicly accessible dataset, which limited the evaluation of generalisability from other sources. Lastly, Google Colab was used for the experiments, which limited the computational flexibility overall, training duration, and batch sizes.

5.6 Concluding Remarks

The experiments showed that MobileNet was a strong, lightweight competitor, but DenseNet-121 produced the best overall performance. All backbones showed consistently better results upon cropping, which shows how crucial it is to eliminate extraneous background from mammography pictures. It has been established that improving feature separation and stabilising training depend heavily on hyperparameter tuning, specifically batch size, learning rate, and temperature.

The study effectively achieved its goals by methodically testing PCL under low-data conditions and demonstrating its feasibility for fine-grained medical imaging tasks, despite the limitations of dataset size, lack of augmentation, and computational constraints. These results lay the groundwork for further research on PCL using bigger datasets, more backbones, and a wider range of medical applications.

This discussion sets the stage for the Conclusion chapter, where the key contributions of the study will be summarized and directions for future research will be outlined.

CHAPTER 6

Conclusion and Future Work

6.1 Conclusion

This study set out to investigate the potential of Prototypical Contrastive Learning (PCL) for breast cancer detection from mammography images under limited-data conditions. The objectives were to (i) implement and adapt PCL to address the scarcity of annotated medical data, (ii) evaluate the performance of multiple backbone architectures—including ResNet-34, ResNet-50, MobileNet, and DenseNet-121—within the PCL framework, (iii) examine PCL’s ability to cope with the high intra-class similarity characteristic of mammography images, and (iv) analyze PCL as a complementary approach to supervised learning for fine-grained medical imaging tasks. These objectives were all successfully achieved. The original PCL was adapted for a single-GPU training, and multiple backbone architectures were then implemented into the PCL framework; the experiments were conducted, and the results were obtained for both datasets.

Each experiment demonstrated consistent improvements came with the removal of excess background, confirming that this cropping off of irrelevant image regions enhances feature learning. Hyperparameter tuning too proved to be critical, with larger batch sizes, smaller learning rates, and lower temperature values producing more stable and discriminative embeddings. Among all backbones, the **DenseNet-121 backbone architecture with PCL framework achieved the best results, with 92.3% accuracy, 92.5% precision, 92.3% sensitivity, and 96.2% specificity** under the cropped dataset ($BS = 16$, $LR = 0.0005$, $T = 0.1$). MobileNet also deserves a mention for producing competitive results despite being lightweight, highlighting that smaller models can generalize well in resource-constrained medical imaging scenarios.

The novelty of this work lies in the application of PCL itself. This framework was previously validated mainly on large natural image datasets like ImageNet—to a small, fine-grained, and high intra-class similarity dataset. The findings here confirm that PCL

CHAPTER 6 Conclusion

can be paired with both lightweight and densely connected architectures to achieve meaningful improvements even with limited data. By showing that SSL, specifically PCL, can help overcome data scarcity and annotation challenges in mammography with the help of a suitable backbone architecture, this study provides an important step toward integrating SSL methods into clinical research pipelines and ultimately supporting early detection and decision-making in breast cancer care.

6.2 Future Work

Future research can build on this study by increasing the dataset size, integrating a variety of sources, and using more augmentation techniques to lessen overfitting and improve generalisation. PCL could be further explored in conjunction with other SSL techniques like SimCLR, BYOL, or SwAV to compare their efficacy in medical imaging tasks. In order to get closer to real-world applications, it would also be beneficial to assess PCL-pretrained models on downstream clinical tasks like lesion detection or multi-class classification. Lastly, the viability of implementing such models in healthcare settings with limited resources may be shown by testing lightweight backbones like MobileNet on edge devices.

REFERENCES

REFERENCES

- [1] Z. Yang *et al.*, “MABEL: An AI-Powered Mammographic Breast Lesion Diagnostic System,” Mar. 2021, doi: <https://doi.org/10.1109/healthcom49281.2021.9398982>.
- [2] N. Berberian et al., “Evaluation of an AI-powered Portable Thermal Imaging Solution as a Pre-screening Tool for Breast Cancer,” *Cancer Screening and Prevention*, vol. 3, no. 1, pp. 8–15, Mar. 2024, doi: <https://doi.org/10.14218/csp.2023.00034s>.
- [3] E. Pachetti and S. Colantonio, “A Systematic Review of Few-Shot Learning in Medical Imaging,” *arXiv.org*, Sep. 20, 2023. <https://doi.org/10.48550/arXiv.2309.11433>
- [4] Gültekin Işık and İshak Paçal, “Few-shot classification of ultrasound breast cancer images using meta-learning algorithms,” *Neural computing & applications*, Apr. 2024, doi: <https://doi.org/10.1007/s00521-024-09767-y>.
- [5] J. Chen, J. Jiao, S. He, G. Han, and J. Qin, “Few-Shot Breast Cancer Metastases Classification via Unsupervised Cell Ranking,” *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 18, no. 5, pp. 1914–1923, Sep. 2021, doi: <https://doi.org/10.1109/tcbb.2019.2960019>.
- [6] X. Chen et al., “Mammo-CLIP: Leveraging Contrastive Language-Image Pre-training (CLIP) for Enhanced Breast Cancer Diagnosis with Multi-view Mammography,” *arXiv.org*, 2024. <https://arxiv.org/abs/2404.15946>.
- [7] R. Gong, J. Wang, and J. Shi, “Task-driven Self-supervised Bi-channel Networks for Diagnosis of Breast Cancers with Mammography,” *arXiv:2101.06228 [cs, eess]*, Aug. 2021, Available: <https://arxiv.org/abs/2101.06228>.

REFERENCES

- [8] Apeksha M, A. S. L, A. P. Bidargaddi, P. B. M, and K. R. K, “Breast Cancer Detection by Prototypical Networks using Few Shot Learning,” May 2024, doi: <https://doi.org/10.1109/inct61516.2024.10593477>.
- [9] A. Medela et al., “Few Shot Learning in Histopathological Images: Reducing the Need of Labeled Data on Biological Datasets,” IEEE Xplore, Jul. 10, 2019. <https://doi.org/10.1109/ISBI.2019.8759182>.
- [10] H. Quan, X. Li, D. Hu, T. Nan, and X. Cui, “Dual-Channel Prototype Network for Few-Shot Pathology Image Classification,” IEEE journal of biomedical and health informatics, vol. 28, no. 7, pp. 4132–4144, Jul. 2024, doi: <https://doi.org/10.1109/jbhi.2024.3386197>.
- [11] J. Zeng, D. Ni, and R. Huang, “Is Two-Shot All You Need? A Label-Efficient Approach for Video Segmentation in Breast Ultrasound,” Is Two-Shot All You Need? a Label-Efficient Approach for Video Segmentation in Breast Ultrasound, pp. 1–5, May 2024, doi: <https://doi.org/10.1109/isbi56570.2024.10635894>.
- [12] E. Pachetti, S. Tsaftaris, and S. Colantonio, “BOOSTING FEW-SHOT LEARNING WITH DISENTANGLED SELF-SUPERVISED LEARNING AND META-LEARNING FOR MEDICAL IMAGE CLASSIFICATION PREPRINT SUBMITTED ON 25 MARCH 2024 TO ELSEVIER,” Mar. 2024. Available: <https://arxiv.org/pdf/2403.17530>
- [13] R. Singh, V. Bharti, V. Purohit, A. Kumar, A. K. Singh, and S. K. Singh, “MetaMed: Few-shot Medical Image Classification Using gradient-based meta-learning,” Science Direct, Jul. 06, 2021. <https://www.sciencedirect.com/science/article/pii/S0031320321002983?via%3Dihub>.

REFERENCES

- [14] Z. Dai et al., “PFEMed: Few-shot Medical Image Classification Using Prior Guided Feature Enhancement,” Science Direct, Oct. 13, 2022.
<https://www.sciencedirect.com/science/article/pii/S003132032200588X?via%3Dihub>
- [15] X. Yue et al., “Prototypical Cross-domain Self-supervised Learning for Few-shot Unsupervised Domain Adaptation,” IEEE Xplore, Nov. 02, 2021.
<https://ieeexplore.ieee.org/document/9577429>.
- [16] L. Wang and H. Lu, “Classification of Histopathologic Images of Breast Cancer by Multi-teacher Small-sample Knowledge Distillation,” 2021 2nd International Conference on Artificial Intelligence and Computer Engineering (ICAICE), Nov. 2021, doi: <https://doi.org/10.1109/icaice54393.2021.00127>.
- [17] Jiang, Hongyang, et al. “Multi-Learner Based Deep Meta-Learning for Few-Shot Medical Image Classification.” IEEE Journal of Biomedical and Health Informatics, vol. 27, no. 1, 1 Jan. 2023, pp. 17–28,
ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=9921265,
<https://doi.org/10.1109/JBHI.2022.3215147>.
- [18] D. Chen, Y. Chen, Y. Li, F. Mao, Y. He, and H. Xue, “Self-Supervised Learning for Few-Shot Image Classification,” ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Jun. 2021 n , doi: <https://doi.org/10.1109/icassp39728.2021.9413783>
- [19] C. Medina, A. Devos, and M. Grossglauser, “Self-Supervised Prototypical Transfer Learning for Few-Shot Classification,” arXiv.org, 2020. Available:
<https://arxiv.org/abs/2006.11325>

REFERENCES

- [20] G. Koch, R. Zemel, and R. Salakhutdinov, “Siamese Neural Networks for One-shot Image Recognition,” vol. 37, 2015, Available: <https://www.cs.utoronto.ca/~rsalakhu/papers/oneshot1.pdf>.
- [21] S. S. Coughlin and D. U. Ekwueme, “Breast cancer as a global health concern,” *Cancer Epidemiology*, vol. 33, no. 5, pp. 315–318, Nov. 2009, doi: <https://doi.org/10.1016/j.canep.2009.10.003>
- [22] B. O. Anderson and R. Jakesz, “Breast Cancer Issues in Developing Countries: An Overview of the Breast Health Global Initiative,” *World Journal of Surgery*, vol. 32, no. 12, pp. 2578–2585, Feb. 2008, doi: <https://doi.org/10.1007/s00268-007-9454-z>
- [23] A. Jemal, F. Bray, M. M. Center, J. Ferlay, E. Ward, and D. Forman, “Global cancer statistics,” *CA: a cancer journal for clinicians*, vol. 61, no. 2, pp. 69–90, 2011, doi: <https://doi.org/10.3322/caac.20107>
- [24] S. J. Nass, C. Henderson, and J. C. Lashof, “Mammography and Beyond,” Google Books, 2024. Available: https://books.google.com.my/books?hl=en&lr=&id=D_1JCdLthAcC&oi=fnd&pg=PA1971&dq=Current+mammography+techniques+are+less+effective+at+identifying+early-stage+breast+cancer .
- [25] Jit Yan Lim, Kian Ming Lim, Chin Poo Lee, and Yong Xuan Tan, “SSL-ProtoNet: Self-supervised Learning Prototypical Networks for few-shot learning,” *Expert systems with applications*, vol. 238, pp. 122173–122173, Mar. 2024, doi: <https://doi.org/10.1016/j.eswa.2023.122173>
- [26] D. Sarwinda, R. H. Paradisa, A. Bustamam, and P. Anggia, “Deep Learning in Image Classification using Residual Network (ResNet) Variants for Detection of Colorectal

REFERENCES

Cancer," *Procedia Computer Science*, vol. 179, pp. 423–431, 2021, doi: <https://doi.org/10.1016/j.procs.2021.01.025>.

- [27] N. Behar and M. Shrivastava, "ResNet50-based effective model for breast cancer classification using histopathology images," *Computer Modeling in Engineering & Sciences*, vol. 130, pp. 823-839, Dec. 2021, doi: 10.32604/cmes.2022.017030.
- [28] Wang, Ziyiing, et al. "ResNet for Histopathologic Cancer Detection, the Deeper, the Better?" *Journal of Data Science and Intelligent Systems*, vol. 15, no. 3, 3 Mar. 2023, ojs.bonviewpress.com/index.php/jdsis/article/view/744#:~:text=The%20performance%20was%20also%20compared
- [29] J. Li, P. Zhou, C. Xiong, and Steven, "Prototypical Contrastive Learning of Unsupervised Representations," arXiv.org, 2020. <https://arxiv.org/abs/2005.04966>
- [30] N. R. Panda, Debendra Muduli, and S. K. Sharma, "Customized MobileNet with Transfer Learning for Enhanced Early Breast Cancer Detection: A Deep Learning Approach," pp. 1–5, Dec. 2024, doi: <https://doi.org/10.1109/scopes64467.2024.10991324>.
- [31] U. Kumar Lilhore *et al.*, "A precise model for skin cancer diagnosis using hybrid U-Net and improved MobileNet-V3 with hyperparameters optimization," *Scientific Reports*, vol. 14, no. 1, p. 4299, Feb. 2024, doi: <https://doi.org/10.1038/s41598-024-54212-8>.
- [32] M. A. Wakili *et al.*, "Classification of Breast Cancer Histopathological Images Using DenseNet and Transfer Learning," *Computational Intelligence and Neuroscience*, vol. 2022, p. e8904768, Oct. 2022, doi: <https://doi.org/10.1155/2022/8904768>.

REFERENCES

- [33] R. K. Pattanaik, S. Mishra, M. Siddique, T. Gopikrishna, and S. Satapathy, “Breast Cancer Classification from Mammogram Images Using Extreme Learning Machine-Based DenseNet121 Model,” *Journal of Sensors*, vol. 2022, pp. 1–12, Dec. 2022, doi: <https://doi.org/10.1155/2022/2731364>.
- [34] A. Desai and R. Mahto, “Multi-Class Classification of Breast Cancer Subtypes Using ResNet Architectures on Histopathological Images,” *Journal of Imaging*, vol. 11, no. 8, p. 284, Aug. 2025, doi: <https://doi.org/10.3390/jimaging11080284>.
- [35] M. Verma, M. Sharma, and P. Pachori, “Interpretable breast cancer classification using CNNs on mammographic images,” arXiv preprint arXiv:2408.13154, 2024.. Available: <https://arxiv.org/abs/2408.13154>

APPENDIX

A.1 Poster

PROTOTYPICAL CONTRASTIVE LEARNING FOR BREAST CANCER DETECTION

EXPLORING BACKBONES AND PREPROCESSING IN LOW-DATA MAMMOGRAPHY

Problem Statement

Detecting breast cancer from mammograms is HARD!

- Limited labeled medical data
- High cost and time in annotating large datasets
- PCL is an SSL that is underexplored

Objective

- Implement PCL for mammography image classification in low-data settings.
- Compare ResNet-34, ResNet-50, and MobileNet and DenseNet backbones in PCL.
- Evaluate PCL's handling of high intra-class similarity in medical images.

Methodology

- PNG images in 3 classes: benign, malignant & normal
- Preprocessing

Prototypical Contrastive Learning for pretraining

- groups similar looking images without needing any labels

Study the performance with ResNet-50, ResNet-34, MobileNet & DenseNet-121 Backbones

Then evaluate with **Accuracy, Precision, Recall, F1-score, and Confusion Matrix**

Novelty

Pioneer in the application of PCL to breast cancer mammography with **high intra-class similarity**.

Model	Accuracy	Precision	Sensitivity	F1-Score	Specificity
ResNet-50	49.8	50.5	49.3	49.9	50.7
ResNet-34	59.0	62.7	59.0	55.7	79.5
MobileNet	66.7	69.6	66.7	65.9	83.3
DenseNet-121	92.3	92.5	92.3	92.3	96.2

For more detailed info and resources,
 contact tharini2206802@1utar.edu.my or call +6011-3107-3155