BIRMINGHAM CITY UNIVERSITY

DOCTORAL THESIS

Data Decomposition Methods for Medical Image Classification

Author:
Asmaa Husien

Supervisors:
Prof. Mohamed Medhat GABER
Dr. Mohammed ABDELSAMEA

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

in the

School of Computing and Digital Technology
Birmingham City University

June 30, 2025

Declaration of Authorship

I, Asmaa Husien, declare that this thesis titled, "Data Decomposition Methods for Medical Image Classification" and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

Abstract

Although deep learning methods have achieved outstanding success in the medical image field, they face several challenges that can significantly impact training effectiveness, learning stability and meaningful generalisations. One of these challenges is the limited availability of annotated samples for certain diseases, which are often difficult and expensive to obtain. Another important issue is the presence of overlapping class distributions, where similarities between features of different classes make it difficult for the model to distinguish between them accurately.

To address these issues, the thesis aims to develop deep learning solutions that improve classification performance when faced with limited annotations and irregular class distributions. The study specifically focuses on three key objectives: (1) designing a convolutional neural network that improves feature learning from a generic domain to a more specific task with small annotated samples, (2) developing a deep learning model that effectively mitigates class overlap by refining class boundaries, and (3) enhancing the generalisation strategy to improve learning stability and simplify complex patterns within datasets.

To achieve these objectives, the thesis presents three main contributions. First, the *4S-DT* model and its advanced version, *XDecompo*, are introduced to enhance feature transferability through self-supervised learning with sample decomposition and overcome the limited samples of the dataset. *4S-DT* uses the *k*-means clustering to perform the decomposition process, which may not always align with the true structure of the data. In contrast, *XDecompo* employs an affinity propagation-based class decomposition to automatically enhance the learning of the class boundaries in the downstream task without the need for preset cluster numbers. This clustering process provides more flexibility and adaptability compared to the parametric approach used by *4S-DT*. Moreover, *XDecompo* also incorporates an explainable component to highlight salient pixels that influenced the model's decision and explain the effectiveness of *XDecompo* to enhance the feature extraction and increase the trust of deep learning applications.

The second contribution introduces *CLOG-CD*, a convolutional neural network that integrates curriculum learning with class decomposition to improve classification performance on medical image datasets exhibiting class irregularities. *CLOG-CD* also explores different oscillation steps to evaluate the impact of varying learning speeds on model generalisation at different levels of granularity.

The third contribution of the thesis introduces a novel curriculum learning with a progressive of self-supervised learning called (*CURVETE*) that employs a curriculum learning strategy based on the granularity of sample decomposition during the training of unlabelled samples. Through this process, *CURVETE* enhances the quality of feature representations, extracting rich information across different levels of granularity. These features can then be effectively transferred to a new

downstream task with limited examples, ultimately improving classification performance. *CURVETE* also handles the challenge of irregular class distribution by utilising the curriculum learning strategy with a class decomposition approach in the downstream task.

Extensive experiments have been carried out on various medical image datasets, utilising different evaluation metrics, to validate the effectiveness of our three contributions to the thesis. For the first contribution, *4S-DT* has achieved a high accuracy of 97.54% and 99.80% for detecting COVID-19 cases in dataset-A and dataset-B, respectively. Additionally, *XDecompo* achieved accuracies of 96.16% and 94.30% for colorectal cancer and brain tumour images, respectively, outperforming *4S-DT* and other training strategies. The second contribution, *CLOG-CD*, achieved an accuracy of 96.08% on the chest x-ray dataset, 96.91% on the brain tumour dataset, 79.76% on the digital knee x-ray, and 99.17% on the colorectal cancer dataset using the baseline ResNet-50. In addition, *CLOG-CD* using DenseNet-121 achieved 94.86%, 94.63%, 76.19%, and 99.45% for chest x-ray, brain tumour, digital knee x-ray, and colorectal cancer datasets, respectively. Finally, *CURVETE* showed significant improvements in performance with an accuracy of 96.60% on the brain tumour dataset, 75.60% on the digital knee x-ray dataset, and 93.35% on the Mini-DDSM dataset using the baseline ResNet-50. Furthermore, the classification performance with the baseline DenseNet-121 achieved an accuracy of 95.77%, 80.36%, and 93.22% on the brain tumour, digital knee x-ray, and Mini-DDSM datasets, respectively.

Acknowledgements

The research presented in this thesis was funded by Birmingham City University as part of the scholars program and supported by many individuals.

First and foremost, I am deeply grateful to Allah Almighty, who granted me the confidence and faith to continue my academic journey and finish my PhD study.

I am deeply thankful to my supervisors, Prof. Mohammed Medhat Gaber and Dr. Mohammed Mohammed Abdelsamea, for their continuous support and for giving me the opportunity to be a PhD student at Birmingham City University. Their guidance, constructive feedback, and encouragement have been fundamental in shaping this research and have greatly contributed to my personal and academic growth. I am profoundly thankful to my family for their constant love and encouragement. To my sisters, your belief in me has been a source of strength and motivation, and I could not have achieved this without you.

I also wish to extend my heartfelt gratitude to my dear friends for their support and encouragement, which kept me focused during the challenging times. Your friendship has been a great comfort throughout this process. Additionally, my thanks go to everyone who contributed to this work in any way and played a part in this academic journey, whether through direct support, helpful discussions, or encouragement. I am truly grateful.

Contents

List of Figures

List of Tables

List of Abbreviations

ANN	**A**rtificial **N**eural **N**etwork
DL	**D**eep **L**earning
ML	**M**achine **L**earning
CNN	**C**onvolutional **N**eural **N**etwork
DCNN	**D**eep **C**onvolutional **N**eural **N**etwork
SVM	**S**upport **V**ector **M**achine
SSL	**S**elf **S**upervised **L**earning
AP	**A**ffinity **P**ropagation
CAE	**C**onvolutional **A**uto-**E**ncoder
SAE	**S**tacked **A**uto-**E**ncoder
XAI	**E**xplainable **A**rtificial **I**ntelligence
CD	**C**lass **D**ecompositon
CL	**C**urriculum **L**earning

Dedicated to the soul of my beloved parents, whose prayers, guidance, and sacrifices have shaped me. Though they are no longer with me, their support and belief in my potential continue to inspire me every day.

Chapter 1

Introduction

1.1 Overview

Computer-aided diagnostic (CAD) systems are powerful tools that leverage artificial intelligence (AI). They help healthcare professionals diagnose diseases and predict patient outcomes more efficiently and accurately, leading to better patient care and outcomes [1]. In recent decades, AI has achieved significant advancements, particularly in machine learning (ML) and deep learning (DL), which have transformed the capabilities of CAD systems. These technologies enable systems to automatically learn from large datasets and refine their performance over time without explicit human intervention. ML and DL are key components of AI that have demonstrated significant effectiveness in improving model performance. Traditional ML relies on manually engineered features, where the model's success is tied to the quality of these features. In contrast, DL offers a more advanced approach by learning hierarchical representations of data through multiple layers, enabling it to extract complex patterns automatically. This makes DL especially well-suited for medical image analysis, where high precision and adaptability are critical [2].

Convolutional neural networks (CNNs) are one of the DL algorithms that have experienced rapid success in recent times. CNNs have shown remarkable success in various medical imaging tasks, such as detecting tumours in MRI scans and classifying medical conditions [3]. They can capture both low-level visual features and high-level abstract patterns from the input data, resulting in superior classification performance and generalisation capabilities compared to traditional methods. Moreover, they are also less sensitive to small changes in the image, such as rotation, zoom, or lighting, which makes them more flexible in real-world situations. A typical CNN architecture consists of two main stages: feature extraction and classification. The feature extraction stage comprises multiple convolutional and pooling layers that learn to detect patterns ranging from simple edges to complex textures. The classification stage, usually composed of one or more fully connected layers, interprets these features to make predictions. As the network deepens, it builds increasingly abstract representations, allowing it to recognise intricate patterns in the input data. However, the demand for more generalisable and adaptable deep learning solutions continues to rise, particularly in applications where accuracy and trust are critical.

This chapter presents the research problem, objectives, and key contributions of the thesis. In addition, the chapter outlines the challenges that arise from working with limited annotated data and overlapping class distributions and introduces the proposed solutions designed to address these issues.

1.2 Motivation

Training a CNN model can be either accomplished from scratch using a large labelled dataset or by utilising the rich knowledge encoded in pre-trained networks through a strategy known as transfer learning [4]. In medical image processing, employing transfer learning with pre-trained networks is often the preferred method, as it significantly enhances classification performance. This approach allows models to leverage the learnt features from large datasets, which substantially reduces the need for extensive labelled data and decreases training time. Through utilising knowledge from general tasks, transfer learning can achieve higher accuracy in specialised medical applications such as disease detection and diagnosis. These benefits make transfer learning an attractive option for many medical image analysis problems.

Despite its potential, transfer learning's effectiveness is often constrained by challenges in achieving generalisability [5]. For example, annotating medical datasets demands extensive expertise and effort from medical professionals, making it a resource-intensive, time-consuming task and further limiting the ability to generalise models effectively. Self-supervised learning (SSL) has become an alternative solution to transfer learning techniques. It addresses the challenge of limited labelled data by leveraging large amounts of unlabelled data and creating tasks that generate labels from the data itself [6]. This approach helps to improve model generalisability and performance on downstream tasks without heavily relying on manual annotations.

Another common challenge in medical image processing is the issue of overlapping distributions between classes, which can lead to high variance and poor generalisation [7]. Overlapping distributions arise when different classes share similar feature spaces, making it challenging for the model to distinguish between them. Fig. 1.1 illustrates the overlapping issues in an imbalanced dataset.

Based on these challenges, we previously proposed a novel deep convolutional neural network (DCNN) based on the class decomposition method called *DeTraC* [8]. To the best of our knowledge, employing the class decomposition approach within a CNN model was the first step toward unbiased medical image classification. Class decomposition is a method used in ML and DL models as a pre-processing step to address overlapping distributions within classes, improving the model's learning ability and generalisation. When different classes overlap, the decision boundaries between them can become unclear, especially if the data points within a class vary significantly or resemble data from other classes. This makes the training process

more difficult for the model to correctly classify new data points. For example, in brain tumour classification, the "tumour" class might include images of different types of tumours, sizes, or early-stage tumours. If the model tries to learn this class as a whole, the diversity within the class may blur the boundaries, making it difficult to distinguish between different types of tumours. By decomposing the broad tumour class into smaller, more homogeneous groups, the model can focus on learning specific patterns. This simplification makes it easier for the model to differentiate between similar data points, resulting in better performance on unseen datasets.

The motivation of this thesis is to leverage the strengths of the super-sample decomposition and class decomposition approaches to develop DCNN models with more generalisation capabilities. These models aim to address common challenges in medical image classification, such as the scarcity of labelled samples in some medical datasets and the presence of overlapping distributions within classes. By addressing these issues, the proposed DCNN models will have the ability to enhance accuracy and generalisability across various medical image datasets. Moreover, we proposed a novel CNN model that incorporates a curriculum learning strategy with the decomposition process to improve feature transferability through different levels of granularity. Leading to enhancing the training process and improving the classification performance on medical image datasets affected by class irregularities.

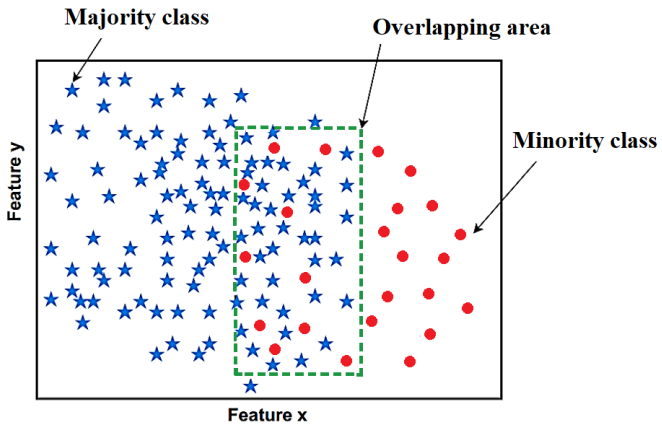

FIGURE 1.1: Illustration of class overlap in an imbalanced dataset. The blue stars represent the majority class, the red circles denote the minority class, and the green rectangle indicates the overlapping region.

1.3 Key Concepts

This section introduces the essential concepts and terminology used in this thesis, providing a foundational understanding of the concepts which will be discussed later in the following chapters.

- **Data decomposition:** This technique is used as a pre-processing step before training the model, aiming to enhance the performance of classification models, particularly in scenarios with overlapping or complex class structures. Its root was first introduced in [9] to enhance low-variance classifiers and increase the flexibility of the decision boundary. There are two types of data decomposition: class decomposition and sample decomposition. Class decomposition focuses on dividing labelled classes in downstream datasets into smaller ones, which is particularly useful when dealing with complex class structures. The method involves dividing a class into smaller, more homogeneous sub-classes based on certain features or characteristics, where each sub-class is assigned a new label related to its original class and considered as a separate new class [10]. After training, those sub-classes are recollected to compute the error correction of the final prediction. This simplification of the local structure within each class enables the model to better capture the specific relationships and boundaries between different data points [11]. By focusing on smaller, well-defined sub-classes, models can achieve improved accuracy and generalisation, making this approach a powerful tool in the medical field. Fig. 1.2 and Fig. 1.3 demonstrate the concept of the class decomposition method and the error correction operation, respectively, for binary datasets.

  In contrast, sample decomposition targets unlabelled data to serve as an initial stage for labelling or further analysis. This approach is a key component in SSL, enabling the model to learn useful representations or extract latent structures within the data without depending on explicit labels. These representations can later be utilised with a small labelled dataset to improve the generalisation of the model. This approach reduces overlap in feature space, enabling the model to better differentiate between sub-classes and create more distinct boundaries. As a result, the performance improves when the sub-classes are recombined for final classification [12].

- **Feature transferability:** This concept refers to the model's ability to learn useful features from one task or training stage and apply them to improve performance on another task or at a different level of complexity [13]. Improving the feature transferability is particularly important in the training model because it enables the model to build on previously acquired knowledge, reducing the need to learn from scratch. This makes training more efficient, especially in complex or data-scarce scenarios [14]. In this thesis, we introduced DCNN models to enhance feature transferability across different stages or new tasks. For example, we enhanced feature representation in the source domain by combining SSL with a sample decomposition process, enabling the model to effectively learn relevant features in the source before fine-tuning them for

a different task. Additionally, we integrated curriculum learning and data decomposition strategies in a progressive manner to enhance feature transferability at different levels of granularity. The model begins by learning specific features and detailed representations, then gradually transfers this knowledge to solve more complex tasks. This structured progression helps the model build a robust understanding of the dataset's structure, ultimately improving generalisation on unseen data.

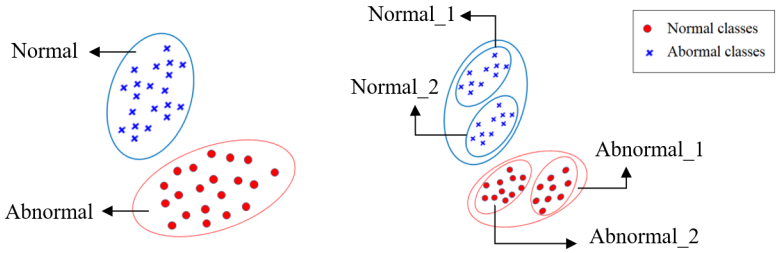

FIGURE 1.2: An example of the class decomposition method applied to a binary classification task on a chest x-ray image dataset. In this approach, each class in the original dataset is divided into two smaller, more homogeneous sub-classes using a clustering algorithm, resulting in a new dataset with four sub-classes. Each sub-class is assigned a label that corresponds to its original class.

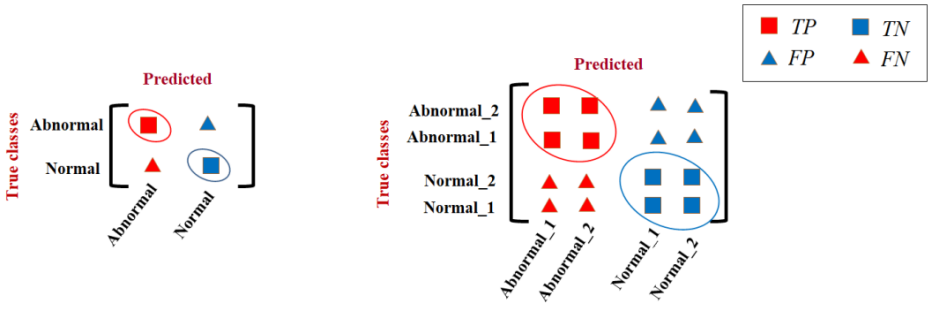

FIGURE 1.3: The figure represents the confusion matrix of a binary classification task and after the error correction process. In the binary confusion matrix, the true positive (TP) refers to correctly identified abnormal cases, while the true negative (TN) refers to correctly identified normal cases. False positive (FP) and false negative (FN) represent incorrect predictions, where FP refers to normal cases misclassified as abnormal and FN refers to abnormal cases misclassified as normal. After class decomposition error correction, the corrected TP is calculated by summing all the correct predictions of classes Abnormal_1 and Abnormal_2 (represented as red squares). Similarly, TN is calculated by summing the correct predictions of Normal_1 and Normal_2 (represented by blue squares). This process is called error correction, where the sub-classes are aggregated back into their respective original classes and remapped to the original problem.

1.4 Problem Statement

The field of medical image classification faces several challenges that affect the effectiveness of machine learning models. With the increasing need for trustworthy and accurate diagnostic tools, it is crucial to tackle these issues to improve the performance of classification systems and enhance patient care. This problem statement highlights the main obstacles that need to be addressed to create more effective and generalised medical image classification solutions.

1. **Irregularity distribution:** One of the notable challenges in the medical imaging domain is the overlap between classes, where different categories share similar features, making it difficult to distinguish between them [15]. Class decomposition is a powerful solution to address this issue by defining clear boundaries between classes. and breaking down complex class structures into simpler, more homogeneous sub-classes, helping the model to identify the differences more effectively. In this way, the model builds a clearer understanding of the overall class structure, allowing the model to first focus on detailed patterns/features within sub-groups before moving to more general features [16].

2. **Inefficient learning under limited annotated data:** Medical image datasets frequently suffer from scarce and expensive labelled data. The lack of sufficient labelled samples can lead to overfitting, poor feature learning, and ultimately limited generalisation of unseen data. To address this challenge, SSL has emerged as a promising solution to deal with such a problem [17]. Rather than depending solely on labelled data, SSL allows the model to learn useful feature representations from unlabelled samples through pretext tasks. These representations can then be fine-tuned using the limited available annotations for downstream classification. As demonstrated in prior works [18], this approach significantly improves model robustness in data-scarce environments while reducing reliance on costly manual labelling in medical imaging applications. Fig. 1.4 demonstrates the fundamental concept behind SSL.

3. **Difficulty in learning from complex dataset structures:** Deep learning models often struggle when the dataset contains a wide range of complex variations and fine-grained details [19]. When training data is presented in a random order, the model is exposed to both easy and hard examples. This can cause the model to focus too much on simple patterns and miss important features in more difficult samples [20]. As a result, this might lead to unstable learning and slower convergence to unseen data. Curriculum learning is an effective solution to improve the training process by changing the model's learning behaviour [21]. This strategy introduces training samples progressively, starting from easy examples and gradually moving to more complex ones. By following this structured progression, the model can build its understanding

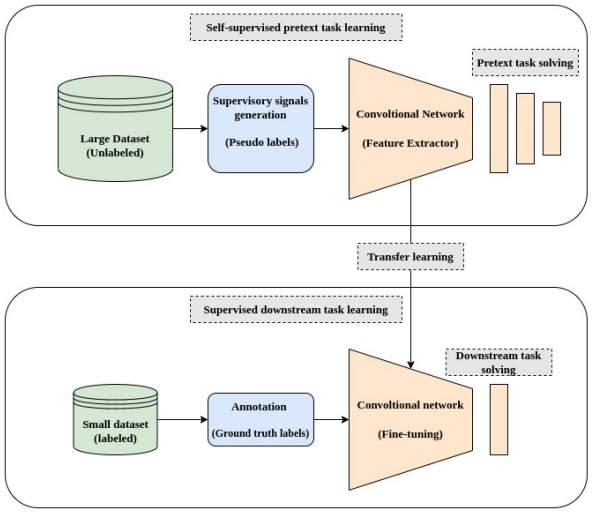

FIGURE 1.4: The pipeline of the SSL process. First, the model is trained on a large set of unlabelled data by solving a pretext task, which generates pseudo-labels from the data itself. Then, an ImageNet pre-trained network is used to train the pretext task to learn rich representations and meaningful features within the data. Finally, the acquired useful features are fine-tuned on a small set of labelled data in a downstream task to improve performance for the final prediction.

incrementally, which leads to more stable training and improved generalisation. Fig. 1.5 illustrates the basic concept of curriculum learning and how this structured progression can guide the model's learning and enhance its overall performance.

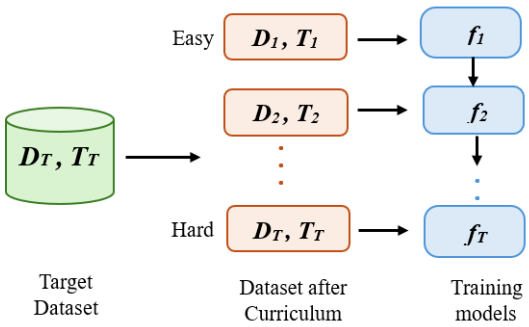

FIGURE 1.5: Illustration of the curriculum learning strategy. The target dataset D_τ is organised into a meaningful order, starting with easier examples and gradually progressing to more difficult ones. The model addresses the target task $\mathcal{T}_T$ by training on a series of sub-tasks $(\mathcal{T}_1, \mathcal{T}_2..., \mathcal{T}_\tau)$. Each sub-task is associated with a corresponding predicted function (f), which helps the model improve step by step until it reaches the final target function f_τ for the downstream task.

1.5 Aim and Objectives

This thesis aims to develop DCNN models for medical image classification. These models will effectively address data distribution irregularities, overcome the limitations of annotated samples, and enhance the training process across diverse medical imaging datasets. As a result, the models improve performance and facilitate the accurate detection of diseases. The objectives of this thesis can be broken down as follows:

- Develop a CNN model capable of improving feature transferability to new tasks by simplifying the structure of classes, enabling the model to learn complex patterns within medical datasets more effectively;
- introduce explainable and interpretable techniques to the machine learning model to increase trustworthiness and usability across various applications in medical imaging processing;
- design CNN models that can effectively resolve overlapping class boundaries and mitigate the impact of limited sample sizes in medical image datasets; and
- build a generalisation and adaptability model capable of enhancing the extraction of feature representations from the latent space, making them more effective for other tasks.

1.6 Solution Approaches and Contributions

In this thesis, we integrated data decomposition with different elements to enhance the performance of DCNN models and address challenges in training medical image datasets. As shown in Fig. 1.6, this thesis presents three main contributions to achieve the objectives mentioned above, which are outlined as follows:

- ***4S-DT* and its advanced version, *XDecompo*:** These models are designed to learn class boundaries and simplify the complex structure in downstream datasets. *4S-DT* combines SSL with super sample decomposition to train a large number of unlabelled samples and fine-tune the knowledge on a small labelled dataset decomposed by k-means clustering. Here, the class decomposition process has been applied using predefined clusters, which may influence the transferability of features and limit adaptability to new tasks. To address this limitation, *XDecompo* employs a dynamic clustering technique that enhances the clustering quality and improves feature transferability on downstream tasks. It extracts deep features from unlabelled images using a convolutional autoencoder and generates pseudo-labels through a clustering algorithm. Then, the ResNet-50 pre-trained network was employed as a backbone for training the pretext model and extracting meaningful information. Finally,

each class in the downstream dataset is divided into smaller, more homogeneous groups using an automatic clustering method. *XDecompo* also integrates a post hoc explainable AI method for feature visualisation, showing its effectiveness in improving feature transferability compared to the *4S-DT* model.

- ***CLOG-CD* Model:** This model was introduced to enhance the training process and address irregular class distributions. *CLOG-CD* combines anti-curriculum learning and class decomposition for training the downstream tasks based on different levels of granularity. The model starts training at the highest granularity of class decomposition, where the dataset is decomposed into the maximum number of smaller groups. The knowledge gained at this level is then used to learn progressively lower granularities until reaching the original classes and then returning to the highest granularity level. In this process, the class decomposition method simplifies complex challenges by breaking down the structure of the classes into more homogeneous sub-classes. This allows the model to initially focus on learning the most relevant features between data points, making the classification task easier. The model was evaluated using three different oscillation steps, which controlled the step size as the model transitioned to the next level. In addition, its performance was compared with other training strategies, including traditional curriculum learning, anti-curriculum learning over a single iteration, and traditional transfer learning from pre-trained networks.

- ***CURVETE* Model:** This model brings together the strengths of all elements: SSL with sample decomposition, curriculum learning, and class decomposition to address challenges such as limited labelled data and irregular class distributions. *CURVETE* achieves this through three main stages: First, SSL is applied in combination with sample decomposition to process a large volume of unlabelled data, enabling the extraction of meaningful feature representations without relying on explicit labels. Second, an ImageNet pre-trained network is adapted for training the pretext task and classifying the pseudo-labels. At this stage, the model leverages the anti-curriculum learning strategy with different granularities of sample decomposition to make the training process more effective. This strategy broadens the solution space, allowing the model to discover new patterns and meaningful representations, which can later be fine-tuned on smaller labelled datasets. Finally, the learnt representations are fine-tuned for classifying a new downstream task, which also utilises the anti-curriculum learning approach based on the class decomposition method to simplify complex structures and establish clear class boundaries. This enables the model to improve feature transferability and effectively handle irregular data distributions.

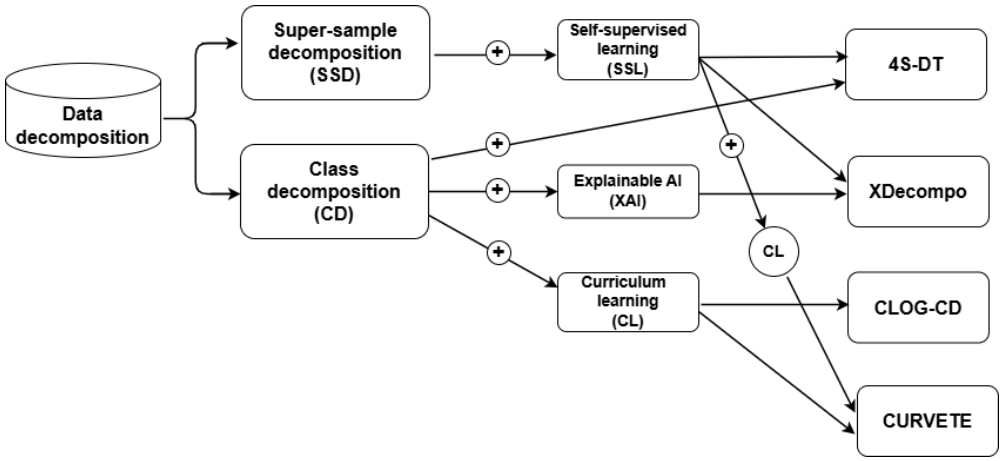

FIGURE 1.6: A framework of the proposed solutions to achieve the thesis objectives.

1.7 Dataset Used in the Thesis

Medical imaging datasets often present unique challenges due to their complexity, high dimensionality, and the need for expert annotation [22]. In this thesis, a diverse set of publicly available medical datasets was used to evaluate and validate the proposed contributions, such as chest x-ray images, brain tumours, colorectal cancer histopathology, digital knee x-ray images, and digital mammogram datasets. These datasets were selected due to their clinical relevance and the presence of various data irregularities, such as class overlap in the morphological structure of medical images. In the experimental work, we used both labelled and unlabelled datasets, depending on the research contribution. Labelled data is used for supervised learning tasks, while unlabelled data is leveraged through self-supervised learning to address limited annotation. The specific usage of each dataset is detailed in the respective contribution chapters. Table 1.1 provides a summary of the basic characteristics of both the labelled and unlabelled data for each dataset.

- **COVID-19 chest x-ray dataset:** This dataset was employed only in the first contribution for detecting COVID-19 cases under the constraint of limited annotated cases at that time. Two different labelled datasets were used, referred to as Dataset-A and Dataset-B. Dataset-A contains 105 COVID-19, 11 SARS, and 80 Normal images, while Dataset-B includes 576 COVID-19, 4,273 Pneumonia, and 1,583 Normal images. In addition, a large number of unlabelled chest radiograph samples were collected from different sources to train the pretext model and address the scarcity of COVID-19 annotation examples; see the first row in Table 1.1.

- **General chest x-ray dataset:** This second chest x-ray dataset was used in the second contribution as a labelled dataset to evaluate model generalisability. The dataset was collected by a team of researchers from Qatar University, the

University of Dhaka, and collaborators from Pakistan and Malaysia, in partnership with medical doctors. It consists of 21,165 x-ray images divided into four classes: 3,616 COVID-19, 6,012 Lung Opacity, 10,192 Normal, and 1,345 Viral Pneumonia, see the second row in Table 1.1.

- **Brain tumour dataset:** The labelled dataset was obtained from Nanfang and General Hospitals at Tianjin Medical University, China, comprising 3,034 images across three tumour classes: 1,426 glioma, 708 meningioma, and 930 pituitary tumours. It was used in all three contributions, making it suitable for comparative analysis of the proposed models. Unlabelled data were also sourced from a public Kaggle dataset to assess model performance under limited annotation conditions; see Table 1.1.

- **Colorectal cancer histopathology dataset:** The labelled and unlabelled datasets were obtained from the NCT Biobank (National Center for Tumour Diseases, Heidelberg, Germany) and the UMM pathology archive (University Medical Centre Mannheim, Mannheim, Germany). The labelled dataset, "CRC-VAL-HE-7K," consists of 7,180 image patches divided into nine tissue types: 1,338 Adipose, 847 Background, 339 Debris, and others including Lymphocytes (LYM), Mucus (MUC), Smooth Muscle (MUS), Normal Colon Mucosa (NORM), Cancer-Associated Stroma (STR), and Colorectal Adenocarcinoma Epithelium (TUM). The unlabelled dataset, "NCT-CRC-HE-100K," contains 100,000 samples with diverse representations of colorectal cancer and normal tissues.

- **Digital knee dataset:** The labelled and unlabelled knee x-ray datasets were sourced from different repositories, as summarised in Table 1.1. The labelled dataset consists of five classes: 514 Normal, 477 Doubtful, 232 Mild, 221 Moderate, and 206 Severe cases. These images were acquired from reputable hospitals and diagnostic centres using the PROTEC PRS 500E X-ray machine, with expert annotation from two medical specialists. In this study, only images from the MedicalExpert-I sub-folder were used. The unlabelled dataset, the Knee Osteoarthritis Initiative (OAI) was used, including 9,786 images categorised into five severity grades.

- **Digital mammograms dataset:** We used the Mini-DDSM dataset, which is a subset of the larger Digital Database for Screening Mammography (DDSM), as a labelled dataset. The dataset is divided into three classes: 2048 Normal, 2,716 Cancer, and 2,684 Benign. In addition, the MIAS mammograms dataset was selected as unlabelled samples, which is a well-known public dataset commonly used in breast cancer detection and diagnosis.

TABLE 1.1: Summary of the datasets used in this thesis.

Dataset Name	Reference/ Sources		Dataset Modality	Image Size	Image Formate
	Labelled dataset	Unlabelled dataset			
COVID-19 chest x-ray	[23], Kaggle[1]	[24, 25, 26, 27]	x-ray	1255 × 2199	Jpg
General chest x-ray	[28, 29]	–	x-ray	299 × 299	Png
Brain Tumor	[30]	Kaggle[2]	MRI	400 × 400	Png
Colorectal cancer	[31]	[31][3]	Histopathology	224 × 224	Tif
knee x-ray	[32]	[33]	MRI	300 × 162	Png
Mini-DDSM	[34]	[35]	x-ray	125 × 320	Png, JPEG

[1] https://www.kaggle.com/prashant268/chest-xray-covid19-pneumonia
[2] https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection
[3] No case overlap between the labelled and unlabelled datasets.

1.8 Publications

The following list of publications supports the work presented in this thesis.

- Abbas, A., Abdelsamea, M. M., Gaber, M. M., 4S-DT: Self-supervised super sample decomposition for transfer learning with application to COVID-19 detection. IEEE Transactions on Neural Networks and Learning Systems, 32(7), 2021.
- Abbas, A., Gaber, M.M. and Abdelsamea, M.M., Xdecompo: explainable decomposition approach in convolutional neural networks for tumour image classification. Sensors, 22(24), p.9875, 2022.
- Abbas, A., Gaber, M.M. and Abdelsamea, M.M., CLOG-CD: Curriculum Learning based on Oscillating Granularity of Class Decomposed Medical Image Classification. IEEE Transactions on Emerging Topics in Computing, 2025.
- Abbas, A., Gaber, M.M. and Abdelsamea, M.M., CURVETE: Curriculum Learning and Progressive Self-supervised Training. ICONIP 2025 **(under review)**.

1.9 Thesis Organisation

The rest of this thesis is organised in the following manner.

Chapter 2 provides a comprehensive background on the architecture and functionality of neural networks, as well as the theoretical foundation for DL and its different techniques. In addition, it emphasises the DCNN architecture and provides a detailed description of the layers and operations in DCNNs. The chapter also discusses common challenges associated with the implementation of the DCNN model, along with an exploration of different training strategies and the pre-trained deep learning models utilised in our research.

Chapter 3 reviews the related work on medical image classification using DCNNs. In addition, we pay attention to the other methods that address challenges related to limited data and overlapping class distributions. In addition, the chapter highlights recent works that have utilised SSL and curriculum learning strategies in improving medical image processing.

Chapter 4 introduces our *4S-DT* model and its enhanced version, *XDecompo*, which improves the class decomposition process by automatically learning class boundaries in downstream tasks. The chapter starts by discussing the *4S-DT* model, which utilises a predefined clustering algorithm to decompose the dataset into a fixed number of sub-classes. Next, we provide a detailed description of the stages that comprise the *XDecompo* model. In addition, we integrate post hoc explainable AI to ensure the trustworthiness of the model's decisions, highlighting specific features and regions of interest that the model accurately localised. *XDecompo* was evaluated on two distinct medical image datasets, both of which face challenges with overlapping classes. The chapter also presents various performance metrics and compares our model with other state-of-the-art methods.

Chapter 5 introduces the theoretical and mathematical foundation behind the idea of CL, and the two factors that are responsible for the implementation of the curriculum learning strategy. Next, we go through our method and demonstrate how the curriculum learning strategy can enhance classification performance and address distribution irregularities when combined with the class decomposition method. Our method was evaluated through extensive experiments based on four different medical image datasets using two baseline networks. We examined our method with different steps of oscillation and compared its performance against different training strategies. Finally, we compared our model with other recent methods in the medical imaging field.

Chapter 6 explores the effectiveness of applying a curriculum learning strategy with different levels of granularity in sample decomposition to improve feature representations from generic, unlabelled samples during pretext training. The chapter outlines the key stages of our proposed model, which combines curriculum learning and sample decomposition to enhance the quality of feature learning in the pretext phase. Additionally, the method applies curriculum learning with the class decomposition approach for training the downstream classification task. The model was tested using two different granularity components during pretext training, enabling the model to gradually learn and capture more refined features. Our method was evaluated on three small datasets and compared to other training strategies and state-of-the-art models in the field.

Chapter 7 concludes this dissertation by summarising and discussing the main findings and contributions made throughout the research, reflecting on the initial aims and objectives. It also introduces new directions for future research in the medical image field.

Chapter 2

Background

In the previous chapter, we detailed the problem statement, defined the aims and objectives of the research, and provided an overview of the contributions of this work. This chapter focuses on the essential background information and foundational concepts that are necessary to understand the methodologies and contributions discussed in chapters 4, 5, and 6.

2.1 Artificial Neural Networks (ANN)

Artificial neural networks (ANNs) are computational systems inspired by the structure and function of the human brain. They consist of layers of interconnected nodes (also known as neurons) designed to process and transmit information. This enables the recognition of patterns and the solving of complex problems [36]. ANNs consist of an input layer, one or more hidden layers, and an output layer. Networks with only one hidden layer are often referred to as simple or traditional neural networks, while those with multiple hidden layers are known as deep neural networks (DNNs). In most neural networks, neurons from one layer are connected to neurons in the subsequent layer, with each connection having an associated weight. These weights determine the influence one neuron has on another, processing the input data and transmitting the results to the next layer. Through this layered structure, the network learns progressively more about the data, capturing deeper and more complex patterns at each stage. The final layer produces the output based on the cumulative learning from all preceding layers. This hierarchical depth is essential for tackling complex tasks such as image and speech recognition, as well as natural language processing. By passing data through multiple layers, each layer focuses on different aspects of the input, allowing ANNs to effectively identify and interpret complex features and relationships within the data [37].

DNNs are a fundamental component of deep learning (DL), which itself is a subfield of machine learning (ML) [38]. While ML methods typically require handcrafted features and domain expertise, DL models rely on DNNs to learn features automatically from raw input data. This end-to-end learning process enables DL models to extract increasingly abstract representations through multiple hidden layers.

Each layer captures higher-level patterns, improving the model's ability to recognise complex structures.

2.1.1 Fundamental Structure of DNN

DNN consists of multiple layers: the input layer, which receives raw input data such as images, text, or numerical values. Each layer comprises interconnected nodes (neurons) that process data through weighted connections, and each neuron corresponds to one feature of the input data. Hidden layers are intermediate layers that perform specific computations on the features entered from the input/previous layers to provide more abstract and useful representations before passing the result to the next/output layer. This structure allows the network to learn complex patterns through a series of computations before passing the result to the final output layer [39].

Fig. 2.1 represents the main building block of DNN. As shown, the learning process in DNN requires transforming inputs across numerous layers of neurons to produce the desired output. For each neuron, the training observation x is multiplied by the weight w, and the output is added with a bias b to allow the network to fit the data when all input features are equal to 0. This process is called linear-transformation and can be represented as $z = w_1x_1 + w_2x_2 + \cdots + w_nx_n + b$. The linear transformations can be used to classify data with linear models, which are essential for tasks where drawing a straight decision boundary is sufficient, such as classifying email as spam or not. However, in real-world data, the model needs to learn complex relationships between patterns to generalise such data that cannot be modelled or represented with linear transformations. Using the activation function is essential for the transfer of linear signals into non-linear ones, making the model more powerful and flexible to learn complex patterns in the data [40]. As shown in Fig. 2.1, the sum z is passed through an activation function ϕ to decide whether the information received is important enough to determine the output and make the decision and pass it along or not. The mathematical formula of the activation function can be represented as:

$$y = \phi\left(\sum_{i=1}^{n} x_i w_i + b\right), \tag{2.1}$$

There are several types of activation functions that can be used in DNN to transfer the input signals, such as Sigmoid, ReLU, and Softmax; see Fig. 2.2. The sigmoid function maps any input to a value between 0 and 1, and it is commonly used in binary classification tasks where the model predicts the probability of an input belonging to a certain class. Whereas the softmax function is usually used in multi-classification problems to normalise the output into (n) probabilities based on the categories of the data. The ReLU (Rectified Linear Unit) function is widely used in deep learning models, such as convolutional neural networks (CNNs), particularly in real-world applications. ReLU is a popular activation function because, for

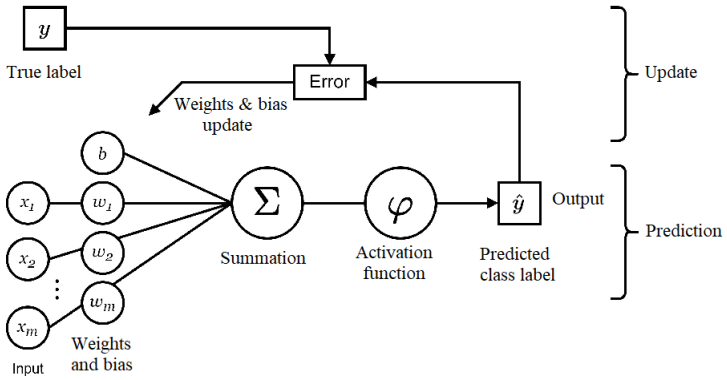

FIGURE 2.1: Learning process in the neural network.

positive input values, it maintains a gradient of 1, preventing the vanishing gradient problem that can occur during back-propagation and allowing for more efficient learning and faster convergence during training. Fig. 2.2 shows examples of activation functions and their mathematical equations.

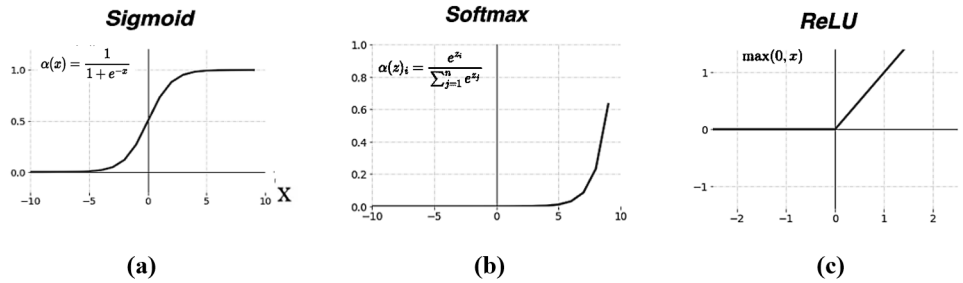

FIGURE 2.2: Examples of activation functions commonly applied to neural networks: a) Sigmoid, b) Softmax, c) Relu.

At this stage, DNN acquires knowledge through feed-forward propagation, where input data moves through the network from the input layer to the output layer in one direction to make initial predictions (predicted class labels) [41]. These predictions are compared to the actual outcomes (true class labels) using a loss function, which measures the error between the predicted and actual outputs. This error is then used to adjust the network's parameters, guiding the learning process. The smaller the value of the loss function, the better the model's predictive accuracy, as it indicates the model is making predictions closer to the true values. Let n be the number of observations, $\hat{y}$ the predicted values, and y the actual values, then the Mean Squared Error (MSE) loss function l can be defined as:

$$l(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} (\hat{y} - y)^2 \tag{2.2}$$

MSE is commonly used in regression tasks, such as predicting continuous values. For classification tasks, cross-entropy loss is the most common loss function, which is particularly suited for multi-class classification problems [42]. Cross-entropy loss measures the difference between the true probability distribution (actual labels) and

the predicted probability distribution (predicted labels). By penalising incorrect predictions based on their confidence, cross-entropy loss encourages the model to assign higher probabilities to correct classes. It is also effective when combined with softmax activation in the output layer, where the model's predictions are turned into probabilities that sum to 1, making it easier to compare them with the actual labels and guiding the model to make better predictions. Cross-entropy loss for multi-class classification can be represented as:

$$l(X,Y) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{c} y_j^{(i)} ln a_j(x^{(i)}) \tag{2.3}$$

Where $X = \left\{x^{(1)}, x^{(2)}, ..., x^{(n)}\right\}$ are the input samples in the training dataset, $Y = \left\{y^{(1)}, y^{(2)}, ..., y^{(n)}\right\}$ are the corresponding labels for these inputs, c is the number of classes, and $a_j(x^{(i)})$ is the predicted probability of class j for the input $x^{(1)}$ from the softmax activation function.

The process of updating the weights of the connections between the neurons to reduce this error is known as back-propagation [43], see Fig. 2.1. After the model makes a prediction, back-propagation calculates the difference between the predicted and actual output (the error). This error is then propagated backwards through the network, layer by layer, to update the weights using gradient descent. The gradient of the loss function with respect to each weight is calculated using the chain rule of calculus. These gradients are then used to adjust the network's parameters (weights and biases), guiding the model toward an optimal solution. This process is critical for training DNN as it allows the model to learn the relationship between features and complex patterns in real-world data. As a result, the model achieves better performance and higher accuracy. By repeating this process during training, the model gradually learns the best weights that minimise the error, refining its predictions over multiple training cycles (epochs). This iterative process allows the model to improve its performance on both training data and unseen test sets, leading to higher overall accuracy. The general equation for updating the weights during back-propagation is:

$$w_{ij} \leftarrow w_{ij} - \eta \frac{\partial l}{\partial w_{ij}} \tag{2.4}$$

Where η is the learning rate, $\frac{\partial l}{\partial w_{ij}}$ is the partial derivative of the loss function l with respect to the weight w_{ij}, which gives the gradient of the loss with respect to that weight.

In addition, there are external adjustable parameters, known as hyperparameters, that are defined prior to the start of the learning process and have a significant impact on the model's performance. For example, the learning rate must be chosen by the user to determine the size of the updates made to the model's weights during training. Selecting the learning rate is essential for controlling how quickly or slowly

the model learns [44]. A low learning rate can cause the loss function to decrease very slowly, increasing the risk of the model getting stuck in a bad local minimum. On the other hand, a high learning rate can prevent the model from learning useful patterns, causing the loss function to fluctuate or even diverge, missing the optimal solution. Therefore, finding an appropriate learning rate is crucial for effective training. In practice, techniques like step decay, which gradually reduce the learning rate at regular intervals, are often used to help the model converge more efficiently over time.

2.2 Challenges in Training DNN

The effectiveness of a deep learning model is primarily assessed by its performance on unseen datasets rather than merely its accuracy on the training data. This subsection discusses common problems encountered during training deep learning models and the techniques that can be used to overcome these issues. One significant challenge that should be taken into consideration before training the DNN is the high demand of a large volume of labelled data, which can be expensive or scarce, particularly in fields like medical imaging. To address this, data augmentation (AUG) is often employed as a strategy to artificially increase the size and diversity of the dataset by applying transformations to the existing data [45]. These transformations include cropping, flipping (both vertically and horizontally), sharpening, shifting, adding noise, and rotating images at different angles. Another common challenge is the frequent occurrence of overfitting and underfitting in machine learning and deep learning [46].

Fig. 2.3 illustrates the relationship between model complexity, represented by the number of weights and parameters included in the model, and the prediction error displayed on the y-axis. As illustrated in the figure, the relationship between model complexity and error rates shows distinct trends. From left to right, as model complexity increases, the training error (green curve) consistently decreases. Conversely, the test error (red curve) initially declines but eventually starts to rise, illustrating the phenomenon known as overfitting. This occurs when the model exhibits high variance, becoming overly sensitive to fluctuations in the training data. As a result, the model memorises the training examples, including noise, rather than learning the underlying patterns. This leads to a low training error but a high test error. In other words, the model is too complex and fits the training data well but performs poorly on unseen test data.

On the left side of the figure, where model complexity is low, we encounter underfitting, characterised by high bias. Here, the model lacks sufficient capacity to capture the relationships between patterns in the data, resulting in high error rates and poor performance on both training and test datasets. This issue is often observed in shallow neural networks, where the model does not contain enough parameters or is oversimplified in its architecture. Therefore, to address underfitting, one can

increase the complexity of the model by adding more neurones to the hidden layers or by increasing the number of hidden layers, allowing the model to capture more intricate patterns. Achieving an optimal fit requires developing a robust model that strikes a balance between complexity and simplicity, ensuring low test error and strong generalisation to new unseen data.

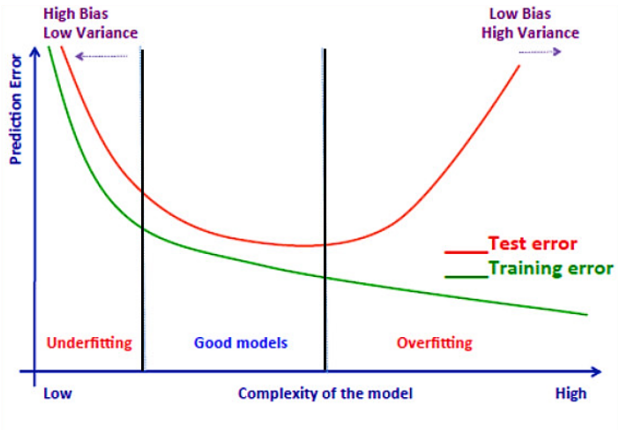

FIGURE 2.3: The relationship between model complexity and loss function error, insights into underfitting, overfitting, and the bias-variance trade-off.

There are several techniques to prevent or mitigate overfitting, including AUG, early stopping, dropout, and regularisation techniques. 1) An early-stopping training technique can be used when the validation loss no longer improves after a certain number of epochs, preventing the model from fitting the noise in the data [47]. 2) AUG, as discussed earlier, helps increase the dataset size by applying several transformation processes. 3) Dropout [48] is another common technique in which, during each iteration, a fraction of neurons and their connections are randomly dropped from the network based on a probability p (commonly set to 0.5). This means that their weights cannot be updated nor affect the learning of the other network nodes. 4) Regularisation techniques [49], such as L1 and L2 regularisation, control the model's complexity and prevent overfitting by adding a penalty term λ to the loss function, where for every weight w in the network, we add the term λ to the loss function to minimise the loss on the training set. L1 regularisation keeps only the useful features and drives the weights of some features to be zero. It means if the input features have weights closer to 0, L1 norm would be sparse during optimisation. This can be useful to focus on the most relevant features, but it may also cause the loss of some useful information that influences the final output. In comparison, L2 regularisation, also known as weight decay, does not force any weights to zero but instead penalises larger weights. This allows the model to learn as much information as possible and forces some weights to be small without rejecting or making them exactly zero. As a result, L2 regularisation helps the model generalise better by returning a non-sparse solution, where all the weights remain non-zero. In practice, L2 regularisation is well-suited for medical analysis tasks because medical

images often contain subtle and complex features spread across different regions. By retaining non-zero weights, L2 regularisation ensures that no potentially important features are completely ignored, helping the model capture specific patterns that are essential for accurate diagnosis or classification. The modified loss function after adding the L2 regularisation term is represented as:

$$l(X,Y) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{c} y_j^{(i)} ln\alpha_j(x^{(i)}) + \frac{\lambda}{2n}\sum_{k=1}^{m}\omega_k^2 \tag{2.5}$$

Where the first term represents the cross-entropy loss in Eq. 2.3, the second term is the L2 regularisation, ω_k is the weights of the model, m is the total number of weights, and λ is the regularisation hyper-parameter that controls the strength of the L2 regularisation. Choosing λ properly is critical for the model's performance. If λ is too small, the regularisation will have little effect, allowing the model to potentially overfit the training data. On the other hand, if λ is too large, the regulariser will dominate, forcing the weights to become too small or even zero, which could lead to underfitting, where the model fails to capture important patterns in the data.

2.3 A Taxonomy of ML and DL Techniques

ML and DL techniques can be broadly categorised into three major types: (i) supervised or discriminative learning, (ii) unsupervised or generative learning, and (iii) hybrid learning and other advanced strategies. Each of these categories serves distinct purposes depending on the task and data requirements [50].

1. **Supervised or discriminative learning [51]:** These techniques learn from labelled data to predict target outputs, enabling classification, regression, and detection tasks. Common ML algorithms include Support Vector Machines, Decision Trees, and Random Forests. In DL, architectures such as CNNs and Recurrent Neural Networks are widely used.

2. **Unsupervised or generative learning [52]:** In this category, models are provided with no information about the target class labels of the dataset. They are capable of discovering the inherent structure of unlabelled data and learning meaningful representations without relying on labelled samples. This approach is used for tasks such as clustering, association, and dimensionality reduction. Include k-Means clustering and principal component analysis in ML, while DL methods include Self-Organising Maps, and Restricted Boltzmann Machines.

3. **Hybrid learning and other advanced strategies [53]:** These approaches combine aspects of both supervised and unsupervised learning to leverage the strengths of each method, resulting in more flexible and effective models. These models can use labelled data (supervised learning) to make accurate predictions while also using unlabelled data (unsupervised learning) to uncover

hidden patterns or structures within the data. This approach is especially useful in scenarios where labelled data is limited but large amounts of unlabelled data are available. There are several techniques in this category, such as: semi-supervised learning, and self-supervised learning.

2.3.1 Self-supervised Learning Technique

For many years, the development of learning methods in computer vision has focused on improving model architectures, often assuming access to high-quality annotated data. However, in practice, obtaining such data is costly and labour-intensive, which frequently leads to models being trained on suboptimal datasets. SSL is a type of ML in which models learn to understand data without relying on labelled examples [54]. It can be used in a wide range of applications, including computer vision, natural language processing, and speech recognition. Unlike supervised learning, which requires large volumes of manually annotated data, SSL leverages the natural structure and properties of the data itself to generate pseudo-labels. This enables the model to learn useful features without explicit supervision, making it particularly effective in domains like medical imaging, where annotated datasets are limited due to the need for expert input and high annotation costs. The SSL process typically follows a two-stage pipeline. In the pretext stage, the model is trained using automatically generated pseudo-labels to learn meaningful representations from unlabelled data. In the downstream stage, these learnt representations are transferred to target tasks that have limited labelled data, such as disease classification in medical imaging. This approach enables the model to generalise better, particularly when annotated data is scarce or expensive to obtain [55].

SSL includes several types, each defined by the nature of the pretext task used to extract meaningful features from unlabelled data [56], as follows:

- **Contrastive learning task:** The model learns to distinguish between similar and dissimilar data instances by minimising the distance between similar pairs and maximising the distance between dissimilar ones. This helps the model capture key features and build robust representations by comparing pairs of samples.

- **Non-contrastive learning task:** Involves training a model only using non-contrasting pairs, also known as positive sample pairs. Rather than a positive and negative sample, as is the case with contrastive learning.

- **Generative learning task:** The model learns the structure of the data by trying to reconstruct or generate data that is similar to the input. For example, in autoencoders, the encoder compresses the input data into a lower-dimensional representation (latent space), forcing the model to capture the most important features. The decoder then reconstructs the original data from this compressed

representation. By minimising the difference between the original and reconstructed data, the model learns to capture key patterns, structures, and relationships in the data.

- **Predictive learning tasks:** The model learns the data's structure by predicting certain properties or transformations of the input data, such as rotation angles or spatial positions. This helps the model capture meaningful patterns and features by understanding how changes to the input data affect its representation.
- **Clustering-based SSL:** involves first grouping unlabelled samples into clusters and treating these clusters as pseudo-labels. The model is then trained on these pseudo-labelled samples to learn discriminative features during the pretext stage. These learnt features can be fine-tuned or directly applied to downstream tasks such as classification or detection. This clustering-based approach is particularly effective when labels are unavailable but data contains latent semantic structures that can be uncovered through unsupervised grouping.

2.3.2 Curriculum Learning Strategy

Curriculum learning is also another type of ML designed to mimic the way humans learn, where individuals tend to learn simpler concepts first, and as their understanding grows, they gradually tackle more complex tasks. It was first introduced by Bengio et al. in 2009 and has become popular in many areas of machine learning [21]. Unlike traditional training behaviours in ML and DL models that present data in random order, curriculum learning introduces a structured progression starting with learners beginning with basic concepts and gradually advancing to more difficult material. The main goal of curriculum learning is to help the model learn better and faster by building a strong base with easy samples before facing more complex data. This method can improve the model's performance, help it generalise better to new data, and make it more stable during training [57]. Curriculum learning relies on two key factors to organise the training process: the curriculum schedule and the pacing function. The curriculum schedule determines the right moment to introduce new samples or tasks based on their difficulty. While the pacing function controls how quickly the model moves from easier to harder examples, guiding the learning progression in a structured manner. Curriculum learning has extended to include various forms of curriculum as follows:

- **Vanilla curriculum learning:** This type is the basic form of curriculum learning, where the model starts training with the easiest examples and gradually progresses to more difficult ones based on a predefined difficulty order.
- **Self-paced learning (SPL):** This type eliminates the need for prior knowledge about the order of the training samples. Instead, the model is allowed to reorder the samples dynamically based on its learning progress. The model

starts with easier examples and progressively selects harder ones as it gains more experience, enabling more flexible learning without relying on external guidance.

- **Balanced curriculum learning (BCL):** This approach builds on the idea that the model shouldn't bias one class or image over another. It ensures that sample selection is diversified and balanced under additional constraints. By incorporating various training examples into the CL framework, BCL ensures the model learns from a broader range of data, enhancing its generalisation capabilities.

- **Self-paced curriculum learning (SPCL):** This approach combines aspects of both vanilla curriculum learning and SPL. Initially, the training samples are ordered based on their complexity. Then, during the training process, the model is allowed to reorder the data, adjusting its focus as it learns. This provides a balance between predefined structure and dynamic adjustment, helping the model to adapt to the data more effectively.

- **Progressive curriculum learning (PCL):** This type allows the model to develop gradually without explicitly sorting the data by difficulty. Instead of focusing on organising the data in a specific order, PCL adjusts the model or task complexity over time. This can involve techniques such as using higher dropout rates or focusing on coarse patterns initially. As training progresses, these simplifications are gradually removed, making the task more challenging and allowing the model to learn progressively more complex representations.

- **Teacher-student curriculum learning (TSCL):** This approach involves two models: a teacher and a student. The teacher model learns to adjust optimal learning parameters based on feedback from the student model and creates a training schedule. The student model is then trained according to the schedule set by the teacher, allowing for more efficient learning through guided progression.

Curriculum learning has been applied across a wide range of machine learning tasks and domains. For example, in computer vision, it helps improve performance in tasks like image classification, object detection, and medical image analysis by gradually introducing complex examples. In natural language processing, it has been used for tasks such as language modelling, machine translation, and text summarisation, where training begins with simple sentences and gradually includes more complex structures [58]. In addition, it is used in reinforcement learning to help agents learn better strategies by starting with easier environments or goals. These applications show that curriculum learning can make models more stable, faster to train, and better at generalising unseen data across different scenarios.

2.4 Deep Learning Approaches for the Classification Task

Image classification is a fundamental task in computer vision, where an algorithm is tasked with assigning a collection of images to predefined classes or categories based on shared characteristics or distinctive features. For example, in medical image classification, the goal might be to determine whether an image represents a normal or abnormal condition. The classification model is trained using labelled data, allowing it to learn and recognise patterns, features, or characteristics that correspond to specific categories. Once trained, the model can predict the class of new, unseen images by analysing their features and assigning them to the appropriate category.

Image classification has been extensively explored using a range of techniques, from traditional machine learning algorithms such as Support Vector Machines, k-nearest Neighbours, and Random Forests to more advanced deep learning-based approaches [59]. Traditional methods typically rely on hand-crafted features (e.g., texture, shape, intensity histograms), which often limit their performance, especially on complex medical imaging data. Therefore, the performance of such methods is often constrained by the quality and relevance of the features chosen, making them less robust. In contrast, DL models eliminate the need for manual feature extraction by automatically learning a hierarchy of discriminative features from the raw pixel data through multiple layers. As a result, most deep learning algorithms have garnered significant attention from researchers.

There are several DL models, such as CNNs, Vision Transformers (ViTs), and Recurrent Neural Networks (RNNs), that serve as the backbone of many state-of-the-art applications in computer vision. CNNs have demonstrated exceptional performance and have become the dominant architecture for image classification tasks, particularly in medical imaging [60]. ViTs, on the other hand, are a more recent architecture that leverages self-attention mechanisms to model global relationships between image patches, showing promising results in large-scale image classification tasks [61]. Another commonly used DL model is RNN, which is effective for processing sequential data and has been applied to tasks involving temporal or slice-based medical imaging [62].

In this thesis, CNNs were selected as the best choice for building an accurate and efficient classification model for several reasons. First, CNN has the ability to detect local elements in an image in a hierarchical manner, where the low-level layers are designed to learn general representations, while the high-level layers capture more complex features and patterns. Second, they improve translation invariance due to their reliance on local patterns in the data and the use of weight sharing. This means that the same filter is applied across different regions of the input image, allowing the model to recognise patterns regardless of their position. Another key advantage of CNNs is their parameter efficiency, as weight sharing and local receptive fields significantly reduce the number of parameters during training. This makes them less sensitive to overfitting and allows them to work effectively even with smaller

datasets.

In comparison, ViTs are computationally intensive due to the quadratic complexity of the self-attention mechanism, which scales with the number of patches in an image [63]. Although ViTs are effective at capturing global context and have gained popularity in large-scale tasks, they generally require larger datasets and more computational resources to achieve optimal performance. Finally, ViTs may struggle with tasks that demand attention to local details, especially when data is limited, making CNNs a more practical choice for many medical imaging applications. Consequently, CNNs are widely adopted in various computer vision tasks, including image classification, object detection, and medical image analysis [64].

2.5 The Architecture of CNNs: Building Blocks and Structure

CNNs are a popular discriminative deep learning architecture, and their structure demonstrates remarkable success in a wide range of applications, especially at recognising patterns and structures [60]. CNNs operate through a sequence of layers that automatically learn spatial hierarchies of information from input images, with each layer performing specific functions to process and extract relevant features. In this section, we discuss the fundamental components of CNN, different strategies for training it, and the pre-trained networks utilised in our contributions.

2.5.1 Feature Extraction Layers in CNN

The primary building blocks of CNNs involve two key tasks: feature learning and classification, see Fig. 2.4. The first part of the CNN structure consists of a sequence of layers: a convolutional layer, a ReLU layer, and a pooling layer. Convolutional layers are responsible for detecting certain local features in all locations of the input images by performing convolutional operations through the collection of filters (also known as kernels) [65]. As shown in Fig. 2.5, the filter slides over the input image, creating feature maps that highlight various elements such as edges and textures, and conducts the inner product between two matrices, one representing the set of learnable parameters (i.e., kernel) and the other representing the weight vector section of the receptive field. The sliding size of the kernel is called a stride. The convolution operation of the 2D input image can be represented below:

$$Out_{ij} = \sum_{m=0}^{k-1} \sum_{n=0}^{k-1} x_{i+m,j+n} w_{mn} + b_{ij}, \tag{2.6}$$

Where $x_{i+m,j+n}$ is the input features, w_{mn} represents the kernel/ filter weights at position (m,n), and k is the size of the kernel/filter. The formula output size can be computed by:

$$Conv_{out} = \frac{W - F + 2P}{S} + 1, \tag{2.7}$$

Where F is the filter, and S refers to the slide size of the kernel, (known as a stride). When the stride is 1, then we move the filters one pixel at a time. When the stride is 2, then the filters jump 2 pixels at a time as we slide them around. Having a larger stride will produce smaller feature maps. P is the amount of padding, and it handles the elements on the edges or that would fall outside of the border of the matrix, which are taken to be zero. (+1) is related to the "Bias", which adds to each layer with the activation neuron. This operation is repeated over many deep layers, with each layer learning to detect different features.

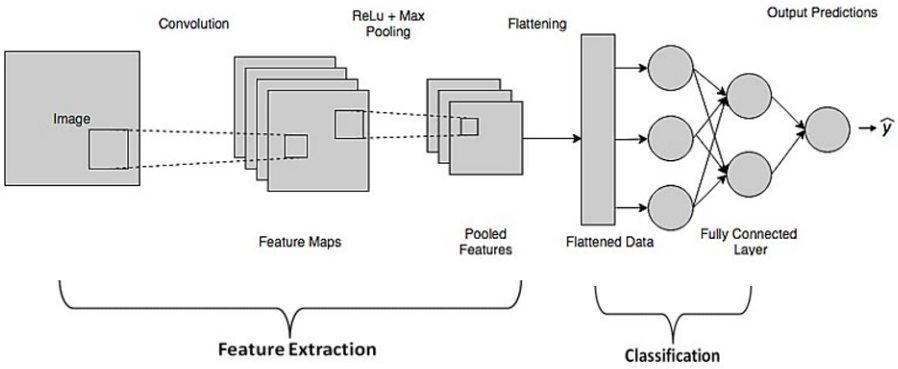

FIGURE 2.4: The design architecture of CNN.

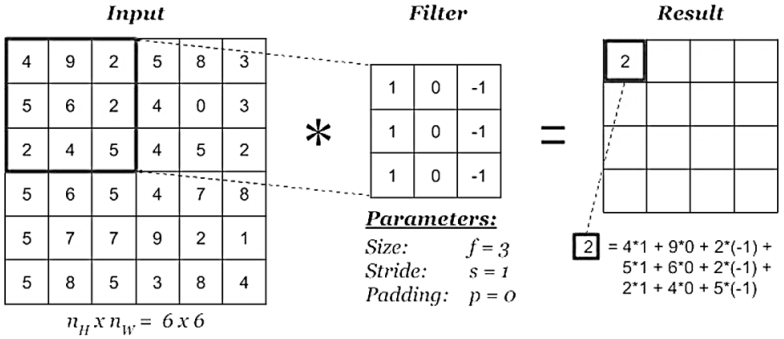

FIGURE 2.5: An example of the 2D convolution operation in the convolution layer. The operation is performed by sliding the filter 3×3 over the input matrix, computing the element-wise multiplication between the filter and the region of the input it overlaps, and summing the results. The result of applying the filter is shown as the first value of the output matrix.

Convolutional layers also use an activation function, like ReLU, to transform the linear operations from the previous step into a non-linear output, enabling the network to discover more complex relationships in the data and eliminating the vanishing gradient problem during model training. The pooling layer is applied separately to each feature map to reduce the dimension and the number of parameters that the

network needs to learn about [66]. This makes the model more computationally efficient and robust to variations in the input data. The most common pooling operations are maximum and average pooling. As can be seen in Fig. 2.6, the input image is divided into a set of non-overlapping rectangles. For each sub-region, a 2×2 window slides across the feature map with a stride of 2 cells. Depending on the pooling method used, either max pooling or average pooling is applied. Max-pooling selects the maximum value within the window, while average-pooling computes the average value. The rest of the values in the sub-region are discarded.

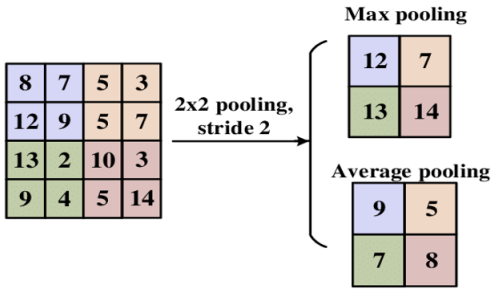

FIGURE 2.6: An example of a down-sampling operation in the pooling layer is illustrated using two of the most popular methods, with a filter size of 2×2 and a stride of 2, which extracts 2×2 patches from the input data. In the max-pooling operation, the output is determined by selecting the maximum value from each patch while discarding the other values. In the average-pooling operation, the average value of the elements within each patch is calculated, providing a different representation of the data.

2.5.2 Classification Layers in CNN

The second part of the CNN architecture is responsible for the classification task. As shown in Fig. 2.7, this part consists of several dense layers that flatten the output feature maps into a vector. This vector is then passed to the output layer, which typically includes a fully connected layer and a softmax activation function. The softmax function provides probabilities for each class, allowing the input image to be classified into the class with the highest probability. The Fully Connected layer is a traditional multi-layer perceptron that takes an input volume from the previous output layer and multiplies it by a vector of weights. The result is an N-dimensional vector, where N is the number of classes in the target data, and each element in the N-dimensional vector represents the probability of a certain class. The softmax layer calculates the probabilities for each class label, highlighting the features that most correlate with a particular class by calculating the corresponding output, where the probability for the correct class is higher and the probabilities for other classes are significantly smaller. Finally, the last layer, known as the loss layer or cost function, provides feedback to the neural network about its predictions. It indicates whether the inputs were identified correctly and, if not, how far off the predictions were. This

feedback helps manage the adjustments of weights across the network to minimise the difference between the predicted probability distribution and the true distribution.

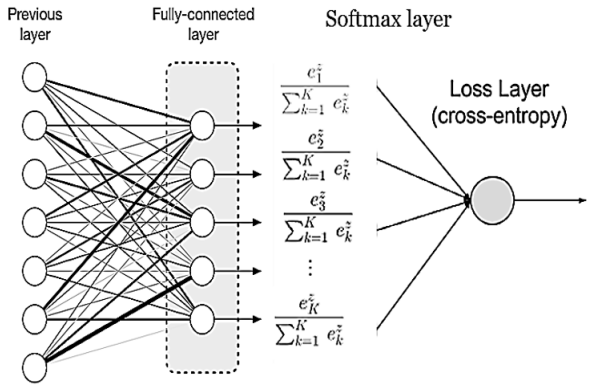

FIGURE 2.7: Illustration of the structure of a fully connected layer with a softmax activation layer and cross-entropy loss for the classification task.

2.5.3 Different Strategies for Training CNN

CNNs are highly effective for processing and analysing visual data, making them essential in numerous image-related applications. There are two main strategies for training CNN models [67]: 1) Training from scratch (or full training), which involves designing a custom network architecture from end to end. It includes selecting the number and types of layers, choosing appropriate activation functions, and configuring the learning process to fit the specific task. Training from scratch is particularly suited for tasks that demand a custom model and involve a large number of output categories. However, this method is less common because it often requires long training times (days or weeks) to train the network and can be challenging when labelled data is scarce or expensive to obtain. 2) The second method is transfer learning from a pre-trained network, which utilises existing CNN models that have been trained on large-scale datasets, called ImageNet. Instead of building a model from scratch, the transfer learning strategy adapts the learnt general features from large datasets to a new specific dataset. This approach is widely used in most DL applications due to its ability to reduce the training time and computational resources needed [68]. Fig. 2.8 illustrates the transfer learning process.

There are three major transfer learning scenarios: a) shallow transfer training, which treats the pre-trained ImageNet as a fixed feature extractor for the new dataset. In this scenario, the weights are not updated during training, except for the last classification layers, which are replaced to match the new task. This approach is quick and computationally inexpensive, making it suitable for tasks where the new dataset is similar to the original dataset the model was trained on. However, performance may be limited when there is significant variation between the datasets.

b) Deep tuning strategy: involves training the entire CNN architecture, including both the pre-trained layers and the newly added classification layers. By allowing all layers to be updated, the model can extract more complex and abstract features from the new dataset. This method is more computationally intensive but provides better performance for tasks with greater differences between the pre-trained and new datasets. c) The fine-tuning strategy: starts the training from the last layer in the model and then incrementally includes more layers in the update process until the desired performance is reached.

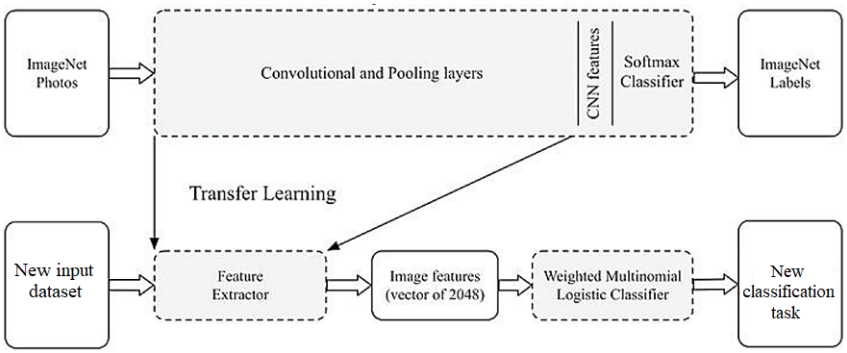

FIGURE 2.8: The framework of the transfer learning technique. The CNN model is trained using an ImageNet pre-trained network, and then the learnt weights are reused when training on a new dataset.

2.5.4 ImageNet Pre-trained Models in CNN

The ImageNet dataset contains a large variety of image datasets belonging to various classes or labels that can be used for training machine learning and deep learning models in various computer vision tasks. Popular deep learning architectures, such as VGG [69], AlexNet [70], Xception [71], DenseNet [72], Inception [73], ResNet [74]. Here, we discuss the architectures we have used in our experimental work.

AlexNet: The network consists of eight layers with five convolution layers, and the remaining three layers are fully connected; see Fig. 2.9. The main advantage of AlexNet is that it minimises the vanishing gradient problem by using the ReLU activation function. So, it can be trained better and faster. The pre-trained network was used as a feature extraction of an input image in our contributions presented in Chapter 4. Fig. 2.9 represents the architecture diagram of the AlexNet pre-trained network.

ResNet: is a short form of residual network and has many variants. It was originally developed to handle problems, such as the vanishing gradient and degradation problem. It uses skip connections, which allow the network to preserve information by adding the output of an earlier layer to the output of a later layer.

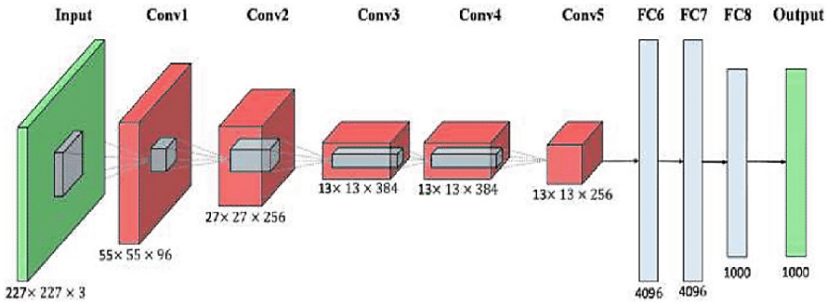

FIGURE 2.9: The architecture of the AlexNet network [75].

This mechanism helps the network train more effectively and learn better representations of the input data. For example, if layer (n) is not learning anything (the output is zero) or gives no useful information. By the skip connection mechanism, the output of layer (n-1) is directly added to the output of layer (n), so the network still receives information from layer (n-1). This means that the network still moves forward using the output from the previous layer (n-1), ensuring no loss of critical data. Then the activation function is applied to this combined output (from both layer (n-1) and layer (n)). This technique allows the network to "skip" unimportant layers when they don't contribute much, while still using their outputs if they do, which prevents the vanishing gradient problem and potentially boosts the network's overall performance. In this study, we used ResNet-50 in the experimental work as a baseline network in our contributions presented in chapters 4, 5, and 6. ResNet-50 is a DNN consisting of an input layer, followed by four stages of residual blocks, each containing three layers (a 1×1 convolution, a 3×3 convolution, and another 1×1 convolution) per block. The network starts with a 7×7 convolution and a 3×3 max-pooling layer and concludes with a global average pooling layer before the fully connected classification layer. This architecture improves accuracy and efficiency in image recognition tasks by simplifying the training of very deep networks. Fig. 2.10 shows the architecture of the ResNet-50 and explains how a skip connection works, where F(x) denotes the ReLU activation function applied on x, and x is the output after applying the convolution operation.

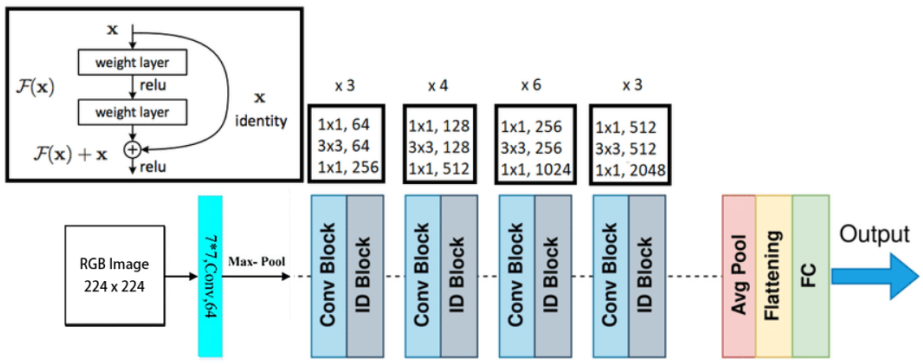

FIGURE 2.10: Structure of deep residual learning [76].

DenseNet: another type of DNN architecture that connects each layer to every other layer in a feed-forward manner. Where each layer receives inputs from all preceding layers and passes its output to all subsequent layers. This direct information flow promotes feature reuse and reduces the risk of vanishing gradients, making it possible to train deeper networks effectively. It also requires fewer parameters since there is no need to learn redundant features, improving model efficiency. In our experimental work, we used DenseNet-121 in our contributions presented in chapters 5 and 6. DenseNet-121 is one of the different variants of DenseNet and is widely used for tasks such as image classification due to its ability to achieve high accuracy with fewer parameters, making it a powerful model for handling complex visual recognition tasks [72]. It consists of 120 convolutional layers followed by a fully connected layer. The network is composed of four dense blocks, each containing several convolutional layers. These dense blocks are interconnected, ensuring that each layer receives inputs from all preceding layers. Between these blocks are transition layers that include 1×1 convolution and 2×2 average pooling operations, which help reduce the number of feature maps and maintain computational efficiency while preserving the dense connectivity that is key to the network's performance.

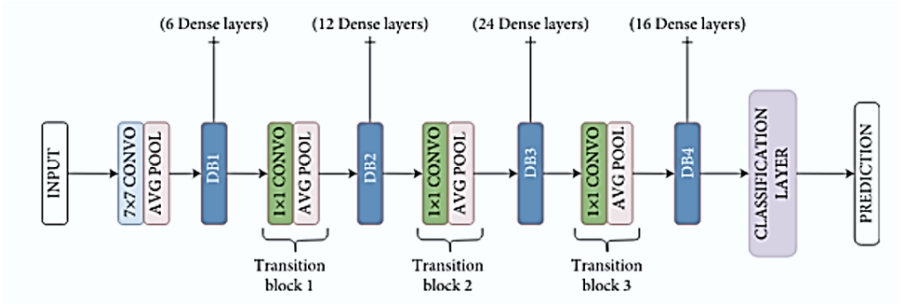

FIGURE 2.11: DenseNet-121 Architecture [77].

2.6 Summary

In this chapter, we provided an overview of AI and the structure of the neural network. We explained the fundamental structure of DNN and the challenges they may encounter during training. Then, we presented the CNN architecture in detail and discussed various strategies for training CNNs. Additionally, we introduced an overview of the pre-trained deep DCNN architectures used in our work. In the next chapter, we conduct a literature review of recent methods that use CNNs for classifying different medical imaging datasets. We will also highlight previous studies that address challenges in training models for medical image analysis.

Chapter 3

A Review on Medical Image Classification

In the previous chapter, we explored the fundamental concepts of deep learning networks and the architecture of deep convolutional neural networks (DCNNs), which serve as the foundation for the models developed in this work. In this chapter, we provide an overview of some of the applications of convolutional neural networks (CNNs) in medical image classification, highlighting the methodologies, advancements, and challenges within this domain.

3.1 Overview

Rapid development in the medical imaging field has significantly improved the ability to identify and detect a wide range of diseases. CNNs, as a powerful tool in the medical imaging domain, have achieved state-of-the-art performance in processing and analysing image datasets [78]. Their hierarchical architecture, which includes layers of convolutions, pooling, and fully connected layers, is designed to automatically learn and extract relevant features from input images. This capability makes CNNs highly effective for various tasks, including medical image classification, enabling accurate classification of various diseases. In Chapter 2, we discussed two primary strategies for training CNNs: full training (commonly referred to as training from scratch) and transfer learning using pre-trained networks. Transfer learning can be implemented through three scenarios: 1) Shallow tuning, in which the pre-trained network is treated as a fixed feature extractor and only the final classification layers are replaced and trained for the new task; 2) deep tuning, where the entire network is retrained on the new dataset to comprehensively fine-tune its weights; and 3) fine-tuning, where selected layers are frozen while others are retrained to adapt to the new task.

In this chapter, we explore the other efforts for medical image classification and recent innovations that have used CNN architectures with transfer learning strategies. Then we discuss other models that utilise class decomposition, supervised learning, and curriculum learning strategy. These approaches have shown considerable promise in enhancing model performance, particularly in scenarios with limited

labelled datasets and irregular distributions. Finally, we conclude with a discussion and overview of the key findings, analysis of ideas, motivations behind the research, and proposed solutions to address the challenges in the field.

3.2 Transfer Learning with CNN for Medical Imaging

CNNs have been successfully applied in various medical imaging modalities, including MRI, CT, X-rays, and histopathological images. They have demonstrated exceptional performance in detecting brain tumours in MRI scans, diagnosing pneumonia from chest x-rays (CXR), classifying malignant tissues in histopathology slides, and detecting colon cancer (CRC) [79]. These applications highlight the effectiveness of CNNs in enhancing diagnostic accuracy and assisting clinical decision-making across multiple medical domains, outperforming traditional machine learning algorithms. There are many works where the authors used the concept of transfer learning strategy to train a CNN and transfer the learnt weights to solve different tasks. This eliminates the need for huge data sets and decreases the training time required for deep learning algorithms constructed from scratch. For example, Sethy et al. [80] utilised a CNN architecture as a feature extraction using 13 pre-trained networks. These features were then fed separately into a support vector machine (SVM) classifier to distinguish between COVID-19, pneumonia, and normal cases. The method demonstrated high accuracy in detecting COVID-19 when training a CNN with ResNet50 compared to other pre-trained networks and traditional machine learning techniques. In [22], Apostolopoulos et al. proved the efficiency of using a transfer learning strategy to enhance the accuracy of COVID-19 detection from CXR images by transferring the pre-trained weights from pre-trained models to a small COVID-19 dataset. The experiment was conducted on two different COVID-19 datasets with three different CNN networks.

Rahman et al. [81] applied three different augmentation processes to generate additional CXR images for the detection of pneumonia. They used the augmented images in a CNN model for binary classification (normal vs. pneumonia) and multi-class classification (normal, bacterial, and viral pneumonia). The model was trained using four different pre-trained networks, with DenseNet201 achieving the highest accuracy. Mabrouk et al. [82] used an ensemble learning (EL) technique of several pre-trained models to enhance the performance of detecting pneumonia from CXR images. Their method utilised the meaningful features extracted from each model to provide a more robust and accurate classification. Wang et al. [83] used the transfer learning strategy using a data set labelled in the same domain as the target data set. The model was fine-tuned to classify four classes of lung cancer using a fine-tuning strategy. This method was compared to traditional machine learning classifiers and deep learning models with pre-trained ImageNet. The results showed that the proposed approach achieved significant performance, surpassing all other models.

Another significant disease following lung cancer is breast cancer, which ranks as the most common cancer among women. Clinical studies emphasise that early detection greatly improves survival rates. Consequently, extensive research has been conducted on computer-aided breast cancer diagnosis from various perspectives. Samala et al. [84] suggested that training a CNN model to classify masses from mammograms using multi-task learning is more effective than single-task learning, which allows the model to share knowledge between related tasks and improves overall performance. In addition, they investigated how varying the depth of convolutional layers impacted the model's performance for classifying masses in two types of mammography images. The experiments found that freezing the initial convolution layer achieved the best results. In [85], three different CNN architectures were employed to extract meaningful features, which were then integrated and fed into three fully connected layers along with an average pooling layer. The results demonstrated that this integrated approach achieved better performance than using the features from the pre-trained models individually. Alkhaleefah et al. [86] investigated the impact of applying different data augmentation (AUG) techniques on the performance of transfer learning models using the VGG-19 pre-trained network for classifying breast lesions into two classes. Their results demonstrated that incorporating AUG techniques improves model robustness and classification accuracy. Ayana et al. [87] used a multistage transfer learning technique with three pre-trained networks to classify breast cancer into two classes. The method started with general feature extraction from ImageNet pre-trained models, then fine-tuning the learnt features on a related dataset of cancer cell images. Finally, these features were then used as a pre-trained model for the specific task of classifying ultrasound images of breast cancer. The authors also used several AUG techniques to increase the dataset and compared the results to a single-stage transfer learning process.

Saber et al. [88] proposed a comprehensive study for breast tumour classification by incorporating several preprocessing techniques and leveraging five pre-trained models for feature extraction with SVM as a classifier. The preprocessing step involved resizing images, noise removal, and applying various AUG methods to enhance dataset quality. Moreover, two segmentation tools were employed to isolate tumours in the images and improve the relevance of the input features. The results demonstrated that fine-tuning specific layers while freezing others yielded better classification accuracy compared to other approaches. Alzubaidi et al. [89] proposed two transfer learning strategies based on the source dataset. The first approach involves pre-training a model on a labelled dataset from the same domain as the target dataset to address the issue of limited training images. The pre-trained model is then fine-tuned on the target dataset to adapt to its specific features. The second approach utilises a collection of two natural image datasets for pretraining, followed by fine-tuning the target dataset, enabling the model to adjust weights for the unique characteristics of the medical images. The study concluded that transfer learning within the same domain using whole-image training performed better

than the other approaches.

Moreover, CNNs have shown promise in automating histopathological image analysis, with potential gains in diagnosis accuracy and efficiency. For instance, Malik et al. [90] conducted a study on colorectal cancer diagnosis by classifying patch images into four categories. They used various AUG techniques to enhance the dataset and employed early stopping methods to prevent overfitting during training. The study included a comparison of different methods: five classical feature extraction techniques followed by an SVM classifier, a CNN using the InceptionV3 pre-trained network with fine-tuning, and a CNN model developed from scratch. The results demonstrated that transfer learning based on the fine-tuning strategy outperformed both the CNN designed from scratch and traditional approaches. Ohata et al. [91] conducted extensive experiments to classify the CRC dataset into eight distinct tissue classes. The study involved two main steps: first, utilising multiple pre-trained networks as feature extractors to analyse the images and extract relevant features. In the second step, these extracted features are fed into five different machine learning classifiers to classify the images based on the features. The results demonstrated that DenseNet169 with SVM significantly enhanced the performance. Kumar et al. [92] presented a comparative analysis of lung and colon cancer classification using two different approaches to feature extraction: handcrafted techniques and transfer learning approaches using pre-trained networks. The results demonstrated that features extracted using DenseNet-121 significantly outperformed those from other pre-trained models and handcrafted methods.

Furthermore, brain cancer has attracted significant attention from researchers in various studies. Due to its complexity and impact on health, many works have focused on developing effective diagnostic and classification methods for brain cancer. Amin et al. [93] utilised various preprocessing and segmentation techniques to isolate the tumour lesion before feeding it into a CNN. They leveraged two pre-trained networks, AlexNet and GoogleNet, as feature extractors. The extracted features were combined and used as input for seven different machine-learning classifiers. Similar work was introduced in [94], where the authors extracted features from GoogleNet using two classifiers, SVM and K-nearest neighbour (KNN). Kokkalla et al. [95] combined three dense layers with ResNet v2 to classify brain tumours into three classes. The results were compared with different pre-trained networks. The model was trained with three different training sizes and a small number of epochs.

Table 3.1: Overview of transfer learning techniques in different medical image datasets.

Paper	**Classification task**	**Dataset modality**	**Transfer learning strategy**	**Accuracy**	**Limitations**
Sethy et al. [80]	COVID-19, pneumonia, normal	x-ray	feature extraction with ResNet-50, SVM	95.33%	Relies on X-rays, unsuitable for critical patients
Apostolopoulos et al. [22]	COVID-19 detection	x-ray	deep-tuning	96.78%	limited data availability
Rahman et al. [81]	binary, multiclass lung cancer	x-ray	deep-tuning	98.00%	Limited data availability
Mabrouk et al. [82]	pneumonia detection	x-ray	ensemble learning	93.91%	determining hyper-parameters high variance and bias in EL
Wang et al. [83]	four classes of lung cancer	CT	Freezing some layers	85.71%	high complexity
Samala et al. [84]	benign and malignant of breast cancer	x-ray	Multi-task transfer learning	0.82 $\pm$ 0.02 (AUC)	computational constraints, limited dataset size
Saber et al. [88]	benign, malignant, normal	x-ray	freezing some layers	98.96%	comparing with one classifier
Alkhaleefah et al. [86]	benign and malignant	x-ray	deep-tuning	90.4%	AUG based on the views of radiologists
Ayana et al. in [87]	benign and malignant	ultrasound	multi-task transfer learning	[99 $\pm$ 0.612%, 98.7 $\pm$ 1.1%]	fixed hyperparameters for all pre-trained models

Continued on next page

Table 3.1: Overview of transfer learning techniques in different medical image datasets. (Continued)

Alzubaidi et al. [89]	four classes of breast cancer	histopathology	fine-tuning from labelled dataset	[90.50%, 96.10%]	lacking other evaluation metrics
Malik et al. [90]	hyperplastic polyp, adenoma, cancer	CRC tissue slides	Freezing some layers	94.50%	limited variability of tissue samples
Ohata et al. [91]	eight classes of CRC cancer	histopathology	feature extraction with multi-classifier	92.08%	high complexity, computational resources
Kumar et al. [92]	lung and colon cancer tissue	histopathology	feature extraction with multi-classifier	98.60%	absence of stain normalization
Amin et al. [93]	three class of brain tumour	[MRI, CT]	feature extraction with multi-classifier	[98.91, 98.01] BRATS 2015	high complexity, time consuming
Deepak et al. [94]	three class of brain tumour	MRI	feature extraction with multi-classifier	98%	limited training duration, overfitting
Kokkalla et al. [95]	three class of brain tumour	MRI	deep tuning	99.66%	limited training duration, overfitting

3.3 Class Decomposition Approach in Medical Images

Class decomposition has been widely used as a preprocessing step in various machine learning algorithms to enhance performance, particularly in cases where there is significant class overlap in the dataset [96, 97]. Initial studies focused on real-world datasets, which provided an essential foundation for developing the class decomposition technique. This concept was later extended and adapted to the medical imaging domain to address challenges such as overlapping distributions and class imbalance.

In the real-world domain, Vilalta et al. [9] proposed a method to reduce classifier bias by incorporating the Expectation Maximisation (EM) clustering algorithm as a pre-processing step before classification. EM decomposes each original class into

smaller sub-classes based on probability density functions, resulting in a refined set of classes. These sub-classes are then individually trained using two different classifiers: Naive Bayes and SVM. Naive Bayes. Their experiments revealed that using too many clusters could lead to poor performance. Elyan et al. [98] used the *k*-means clustering algorithm to decompose each class into more homogeneous sub-groups, with the number of clusters ranging from 2 to 8. Then, Random forests were employed to classify the sub-classes independently and get the final classification determined by majority voting. The results demonstrated that increasing data diversity through clustering allows the Random Forest classifier to better learn and differentiate between these sub-classes, thereby enhancing performance. In [7], the authors proposed the use of the One-vs-One (OVO) strategy to decompose multi-class problems into multiple binary classification tasks for 34 real-world datasets. Additionally, they introduced a new evaluation framework that simulates overlapping scenarios by generating synthetic samples near class boundaries. Through comprehensive experiments, the study demonstrated that OVO-based classifiers are more robust to class overlap compared to traditional approaches.

In the medical image datasets, [99] introduced a method that uses multi-level discrete wavelet transform (DWT) to improve lung nodule classification in CT images. DWT applies fixed wavelet kernels to decompose images into components at different resolution levels, capturing both coarse and fine details. This multi-resolution analysis enhances the ability to detect subtle differences between lung nodules, making it easier to distinguish malignant from benign cases, even when they appear visually similar. Additionally, the extracted features help separate classes more effectively and reduce confusion caused by overlapping visual patterns, leading to improved classification performance. In addition, Polaka et al. [100] applied the class decomposition only to the positive classes in various disease datasets. They hypothesised that diseases could present in several forms, making them easier to classify using different algorithms. The study utilised agglomerative hierarchical clustering and *k*-means clustering to generate sub-classes and then examined how the choice of clustering algorithm affected the performance of classifiers like SVM, Random Forests, and C4.5. Vuttipittayamongkol et al. [101] proposed an overlap-based undersampling method, called URNS, to improve the classification of five imbalanced medical datasets. The method aims to reduce class overlap by identifying and removing the majority class (negative) instances that are too close to minority class (positive) samples in the feature space. URNS uses the k-Nearest Neighbours algorithm to recursively explore the local neighbourhoods of minority instances, detecting and eliminating overlapping majority samples. This process is performed twice, with the output of the first round feeding into the second, ensuring effective refinement of the overlapping region.

Polat et al. [102] used similarity-based attribute weighting combined with clustering algorithms to address the issues of feature overlap and class imbalance. They used three clustering methods (*K*-means, Fuzzy C-means, and Mean Shift) to group

similar data points and calculate the distance from each point to its cluster centre. These distances are then used to assign higher weights to points farther from the centre, highlighting their greater importance for classification. Finally, the weighted features are then passed to different machine learning classifiers. Shimizu et al. [103] proposed a method for classifying four types of skin lesions, which involved three key steps: border detection, feature extraction, and classification. They developed a general border detection algorithm that segmented the lesion into subregions (central, peripheral, whole tumour, and normal skin), treating each subregion as a separate subclass for more detailed feature extraction. For classification, they introduced a layered model based on a task decomposition strategy and compared its performance to two traditional machine learning classifiers. Gultekin et al. [104] proposed a two-tier tissue decomposition model for histopathological image classification. They decomposed each image into multitype objects based on texture, shape, and size information. In the first tier, a texture-based segmentation is applied to identify irregularly shaped local tissue regions. In the second tier, these categorised objects were utilised for SVM as a classifier. In [105], the authors introduced a CAD system for automated classification of brain abnormalities using SVM as a classifier. The approach begins with decomposing the images using two advanced techniques: BEMD and VMD. BEMD separates an image into layers based on patterns found in local intensity variations, helping to isolate fine textures and structural details. VMD, on the other hand, divides the image into a fixed number of frequency bands using an optimisation approach, ensuring that each component is distinct and non-overlapping. This decomposition process enhances the visibility of important features, which are then used to train the classifier more effectively.

Alwuthaynani et al. in [106] proposed a Class Decomposition Transfer Learning (CDTL) model to enhance the classification of Alzheimer's disease using 2D structural MRI images. The method leverages VGG19 and AlexNet as pre-trained networks. In the class decomposition strategy, they used clustering methods to divide the imbalanced classes into more uniform sub-classes. Then, an entropy-based technique is applied to select the most informative image slices to focus on the most valuable parts of the MRI scans. Their study highlights the potential of class decomposition in mitigating data irregularities and improving prediction reliability in Alzheimer's detection. Dif et al. [107] introduced a novel class decomposition approach for histopathological image classification by generating synthetic labels through clustering. They employed two clustering algorithms, *K*-means and MOC-Stream, to group image patches based on morphological and textural similarities. These labels are then used to train an InceptionV3 model, which is fine-tuned for transfer learning on various histopathological datasets. Results show that MOC-Stream clustering consistently outperforms models trained on original labels and ImageNet features, demonstrating the strength of this decomposition strategy in improving generalisation and transfer performance.

We previously introduced *DeTraC* in [8] as the first attempt to employ class decomposition within the CNN framework for medical image classification. In the *DeTraC* model, the downstream dataset was decomposed by the *k*-means cluster algorithm, so each class was divided into smaller classes based on feature similarity. The result of this process is a new dataset where each cluster is considered a class of its own. Then, a transfer learning strategy from different ImageNet pre-trained networks was utilised to evaluate the performance before and after applying the decomposing method. After training on the decomposed dataset, each cluster is returned to its parent class to get the final output. *DeTraC* was evaluated on three different medical image datasets: CXR images, histological CRC, and digital mammogram datasets. The results demonstrated its ability to effectively address irregularities in data distribution within the classes compared to traditional CNN models without the class decomposition approach. The same method was applied in [108] to enhance the detection of COVID-19 in CXR images. We used five different pre-trained networks as the backbone of the initial weights of the transfer learning technique. *DeTraC* showed the capability to detect small cases of COVID-19 images, ultimately leading to higher accuracy in diagnoses. Moreover, in [109], the authors integrated the *DeTraC* method with a novel segmentation approach, TCBOGK, which uses pixel similarity and Gaussian functions to enhance COVID-19 detection in CXR images. The results also proved that *DeTraC* improved the classification performance compared to models without it.

3.4 Self-Supervised Learning in Medical Imaging

Despite the fact that transfer learning has shown success in many medical imaging applications, it does come with certain limitations [110]. One of the main limitations is that the features learnt from natural image datasets may not be fully relevant to medical imaging data. Medical images differ from natural images in several fundamental ways. For example, in medical imaging, the focus is often on identifying small, specific regions that indicate tumours or abnormalities, in contrast to natural images, which are easily recognisable objects. Moreover, medical images often have lower contrast to emphasise fine details like tissue boundaries or minor abnormalities, which can be more challenging to distinguish than the high contrast in natural images. Furthermore, medical image datasets tend to have fewer samples in some diseases, which are often time-consuming or expensive to annotate. As a result, applying transfer learning directly from models pre-trained on natural images may not yield optimal solutions for medical image tasks [111].

Recently, researchers have increasingly turned to self-supervised learning (SSL) techniques as an alternative to transfer learning to address the challenges of medical imaging. SSL enables models to learn useful features from unlabelled data within

the same domain as downstream tasks, allowing the model to capture domain-specific features that are highly relevant to medical image datasets [112]. For example, in [113], SSL was employed to address the limitation of COVID-19 samples and improve the performance of the CNN model. The first stage involves utilising several pre-trained models from ImageNet as the initial model for self-supervised learning on a large set of unlabelled CXR images. Then, the knowledge is fine-tuned using a smaller set of labelled CXR images to effectively detect COVID-19 cases. Additionally, the authors employed an explainable component to enhance the interpretability of the results and provide explanations for the model's decisions. In [18] the authors employed SSL to restore medical image contexts that had been disrupted. The results showed an improvement in classification accuracy for image classification on 2D fetal ultrasound images compared to traditional training networks.

Gazda et al. [114] proposed an SSL model using contrastive learning on unlabelled CXR images. The model generates pseudo-labels through data augmentation, forming positive and negative pairs, and optimises a contrastive loss with a ResNet-50 backbone to learn distinctive features. After pre-training, it serves as a feature extractor for downstream pneumonia and COVID-19 classification tasks. Sowrirajan et al. [115] proposed the (MoCo-CXR) model to enhance feature representations in CXR images through self-supervised contrastive learning. In the pre-training phase, the model learns to increase the similarity between augmented views of the same image (positive pairs) while reducing the similarity between different images (negative pairs). After this, the model is fine-tuned to small amounts of labelled data for detecting pleural effusion and tuberculosis from CXR images. Cho et al. in [116] introduced the (CheSS) model, which employs self-supervised contrastive learning to extract feature representation from 4.8 million CXR images. Then the (MoCo v2) [117] framework was used for training the unlabelled images and transformed the learnt weights into the downstream task to classify multi-class disease classification in the CheXpert dataset. The authors also incorporate the AUG techniques on the downstream task to support the training process.

Ciga et al. [118] used a contrastive SSL with several unlabelled datasets to improve the performance of histopathology segmentation and classification tasks. They proved through extensive experiments that the success of contrastive learning depends on the ability to learn better features from the diversity of unlabelled training sets. This diversity can enhance the learning process by enabling the model to extract more meaningful features, leading to improved performance in another task. Koohbanani et al. [119] introduced (Self-Path) model to enhance the classification performance of a small amount of pathology images. Self-Path is designed to enable the pretext model to learn from multiple aspects of histology images, including different scales, spatial relationships between patterns, and semantic characteristics.

For brain tumour classification, SSL has been applied to improve the performance of detecting tumours and make the learning process highly effective for classification tasks with limited annotations. For instance, Chen et al. [18] developed an SSL model based on the context restoration strategy, where the CNN model learns meaningful features from unlabelled medical images by rearranging and restoring image patches. This technique enables the model to capture semantic features that can be useful for various tasks. Nguyen et al. [120] incorporate both spatial awareness and semantic features for effective learning. where the pretext model trains to classify whether an input image is normal or contains corrupted patches and to predict the origin of these corrupted patches relative to their position, allowing the model to learn not just the visual characteristics of individual slices but also the relationships between neighbouring slices. Wang et al. [121] introduced the MI-SelfL model, which incorporates two pretext tasks: multi-input correspondence and geometric transformation. In the first task, parts of certain images are replaced with regions from other images in the same batch. In the second task, images are randomly rotated and flipped. The model is designed to recognise replaced regions by detecting significant differences from the original images, while images modified only by geometric transformations are expected to preserve similar features. This combination allows the model to learn both semantic and spatial features, which enhances its capacity to recognise certain variations and extract rich information to be used in the downstream task. Mishra et al. in [122] introduced (SSCLNet) to enable the pretext model to learn the latent space using contrastive discriminative methods, where similar images from the augmented data are treated as a positive pair and dissimilar images as negative samples. Through this process, the model becomes more effective at identifying patterns in the data, improving its ability to extract meaningful features and, consequently, enhancing performance in downstream tasks. The authors investigated their method using various ratios of labelled data and experimented with different augmentation techniques and ResNet architectures.

The first attempt to adopt a clustering algorithm for self-supervised learning on large-scale datasets (ImageNet and YFCC100M) was made by Caron et al. in [123]. The authors introduced the DeepCluster method, which used k-means clustering to group unlabelled images into clusters based on their feature representations. They treated these clusters as pseudo-labels to pre-train a DCNN and learn new representations. This process is performed iteratively, improving both feature extraction and clustering accuracy and leading to stronger model performance over time.

3.5 Training CNN with Curriculum Learning Strategy

Another notable strategy that has recently demonstrated improved performance in the medical imaging domain is curriculum learning, which involves gradually introducing training data in a meaningful order based on task difficulty or relevance. This approach helps models learn from easier to more complex cases, improving

stability and generalisation to new, unseen data. In [124], Lotter et al. presented a multi-scale CNN combined with a curriculum learning strategy for improved mammogram classification. Initially, the model is trained on simpler tasks by identifying localised areas of lesions within the segmented mask, allowing it to focus on specific lesion characteristics. Then, these learnt features are used to classify entire mammogram images, where the model applies its understanding of localised features to broader and more complex contexts. Jesson et al. [125] introduced a CASED model for detecting pulmonary nodules. The authors adapted the curriculum learning strategy based on the size and localisation of the lung nodules by gradually increasing the complexity of the training examples, starting with simple concepts before moving on to more complex ones. Their model initially started learning by focusing on the immediate surroundings of the nodules to extract the specific characteristics and essential features of the nodules without being distracted by other elements in the image.

Luo et al. [126] introduced a CNN model with the curriculum learning strategy to improve the classification of a digital mammogram dataset into three classes. The method involves dividing the classification problem into two tasks: an easier binary classification task (Malignant and Negative classes) and a difficult classification task (the original three classes). The training scheduler started by focusing on the easier binary task, allowing the model to build foundational knowledge before transferring it to handle the more challenging three-class problem. The results showed that their method outperformed others in enhancing classification accuracy. Park et al. in [127] applied the curriculum learning strategy based on the classification task from easy to hard. They hypothesised that training on entire CXR images could lead to the model converging on poor local minima due to the complexity introduced by overlapping patterns of organs and tissues, and make the classification problem more difficult. So, their initial step involved extracting patch images around regions of interest (ROI) to focus on thoracic abnormalities, allowing the model to learn detailed features specific to the abnormal lesions. These learnt features were then fine-tuned using entire CXR images within a ResNet-50 architecture to classify each image into five different categories.

Yang et al.[128] introduced (Su-MICL) to classify histopathology based on the severity of the conditions, progressing from easy to hard. Initially, the model is trained with images at the most severe level (easy to learn) using all patch images to acquire the basic features associated with severe conditions, thus providing a detailed understanding of these conditions. Subsequently, the model is retrained with less severe images to refine its accuracy. To manage the increased difficulty, Su-MICL employs a selective priority approach to choose the most informative patches for retraining the model. Tang et al. in [129] introduced AGCL for classifying the CXR dataset into multiple disease labels. The authors used the difficulty of diseases from hard to easy to guide the curriculum learning strategy. Initially, the model focused on high-severity diseases, followed by moderate and mild ones, utilising prior

knowledge to guide the training. Additionally, the model's classification probability scores guided the training process, allowing the model to concentrate on the most confident predictions before advancing to the next level of difficulty.

Another scenario is presented in [130] for classifying the histopathology colorectal polyp dataset. For the curriculum schedule, they used the percentage of agreement among multiple human annotators as a measure of difficulty. Images with a high majority voting agreement were considered easy, while those with a low agreement were defined as difficult. The training set was then divided into four levels of difficulty, with each level being fed separately into a pre-trained ResNet18 network. The learnt features from each level were fine-tuned for the next level. The outcomes outperformed the results from both the baseline and anti-curriculum strategies. Similarly, Jimenez-Sanchez et al. [57] presented two curriculum techniques based on the class difficulty for classifying proximal femur fracture images. The first strategy assumed that there were notable differences between categories and assigned ease weights based on the rank of each class. where easier classes are prioritised early in training, while harder classes are introduced later. The second strategy used the level of annotators' agreement to determine the sampling probability. where samples with higher agreement (indicating lower difficulty) are presented earlier in training, while those with lower agreement (indicating higher difficulty) are introduced later. The results proved that the curriculum learning strategy outperformed the ResNet-50 as a baseline network. Extended to their work, they also introduced in [131] a combination of prior knowledge with the uncertainty of the model to classify the same dataset. The curriculum training schedule is guided by two factors: prior knowledge of the dataset, as determined by clinical expertise, and the uncertainty of the model's predictions during training. Based on these predictions, the model identifies harder images and feeds them later in the training process.

Moreover, combining SSL with curriculum learning has achieved significant improvement in many works, leading to faster convergence, better generalisation ability, and alleviating overfitting to in-domain data. For example, Srinidhi et al. [132] introduced HaDCL to enhance histology image classification performance. They first used two different SSL techniques to train unlabelled sets using ResNet-18, then fine-tuned these learnt representations on the downstream task. HaDCL consists of two stages. First, the model starts with easier examples and gradually progresses to harder ones. Second, the model has fine-tuned the previous knowledge by focusing specifically on the very hard examples. The difficulty of the samples in each mini-batch is ranked based on their loss values, allowing the model to adapt and handle progressively harder examples as training continues. Liu et al. [133] introduced the ACPL method, a semi-supervised learning algorithm that improves classification on CXR datasets by selecting the most informative unlabelled samples for pseudo-labelling, starting from the difficult samples and gradually introducing the easier ones. The cross-distribution sample informativeness (CDSI) guides the

selection of unlabelled samples. The method also employs a mechanism called informative mixup to combine predictions from an ensemble of classifiers and KNN as a classifier, reducing confirmation bias and improving prediction accuracy. Burduja et al. [134] introduced a model that gradually deblurs input images using a Gaussian filter, starting with blurred images, which are easier to align, and progressively transitioning to clear, original images. Alsharid et al. [135] proposed a dual-curriculum approach that integrates both image and text data, allowing for training in a structured manner from simpler to more complex examples. They utilised different distance measures for constructing the curriculum, finding that the Wasserstein distance is most effective for image data while tf-idf works best for text data. Experimental results indicated that their method significantly improved the performance compared to traditional stochastic mini-batch training methods.

The curriculum learning strategy has effectively addressed the issue of data imbalance, leading to enhanced performance in different image analysis tasks. Wang et al. [136] introduced a DCL model as a first attempt to tackle the challenges of imbalanced data in the human attribute dataset. The idea is to enhance model performance by adjusting sampling strategies and loss weights by transitioning from imbalanced to balanced data distributions during the training process. The sampling scheduler starts with fewer samples from minority classes and progresses to more abundant samples from majority classes, allowing the model to develop a robust representation of the minority class samples before focusing on distinguishing among all classes. The loss scheduler controls how much importance is given to the learning features by assigning large weights at the beginning and gradually decreasing their impact over time. Li et al. [137] designed the CLDL model to address the challenges of imbalanced label distributions in medical image segmentation by decomposing the segmentation task into multiple label distribution estimation tasks. The method begins by establishing a region label distribution to minimise disparities in region distributions. The segmentation task is then decomposed into multiple sub-tasks of varying difficulty using a task-oriented curriculum learning strategy. Building on this, the model incorporates prior information from simpler tasks to reinforce feature learning across various stages of the curriculum. Zhao et al. [138] presented the SEDC model to address data imbalances in glaucoma diagnosis. The curriculum learning strategy starts with non-glaucoma classes, utilising the learnt features to better understand the smaller samples of glaucoma images. During the training process, the model adjusts weights assigned to each sample based on their difficulties to ensure that the model does not bias toward the majority class while ignoring the minority class, where samples from the minority class are given more weight. This iterative process gradually refines the decision boundary and enhances overall classification performance.

Table 3.2: Overview of different curriculum learning methods.

Paper	Tasks	Dataset	Criterion	Training schedule	Order	Limitations
Lotter et al. [124]	classification	digital mammograms	label	segmentation mask then whole-image	easy-to-hard	difficulty detecting small lesions
Jesson et al. [125]	detection	lung	size of nodules	nodule size	easy-to-hard	depends on quality of annotated data
Luo et al. [126]	classification	digital mammograms	output task	number of classes	easy-to-hard	limited evaluation metrics, restricting generalisation.
Park et al. [127]	classification	lung	size of ROI	learnt weights	easy-to-hard	limited lesion variety
Yang et al.[128]	classification	histopathology	severity of conditions	patch selection	easy-to-hard	depends on severity labels, lack of context information
Tang et al. [129]	classification	lung	classification probabilities	iterative attention-guided	easy-to-hard	high computational complexity, limited training duration
Wei et al. [130]	classification	colorectal polyp	percentage of annotators agreement	Four levels of difficulty	easy-to-hard	limited annotator diversity, using small and single dataset, generalisation uncertain
Jimenez-Sanchez et al.[57]	Classification	femur fractures	expert annotators	samples-based probabilistic	easy-to-hard	Performance influenced by imbalanced classes

Continued on next page

Table 3.2: Overview of different curriculum learning methods. (Continued)

Jimenez-Sanchez et al. [131]	Classification	femur fractures	clinical expertise	Prior knowledge and prediction uncertainty	easy-to-hard	computationally expensive, sensitive to noisy labels or low resolution
Srinidhi et al. [132]	classification	histopathology	loss values	difficulty of samples	easy-to-hard	computational cost of tracking sample hardness
Liu et al. [133]	classification	lung and skin lesion	difficulty of samples by CDSI	iterative process	hard-to-easy	quality of initial labeled data
Burduja et al. [134]	image registration	liver tumours	Gaussian filter	high blurred images to original classes	easy-to-hard	sensitive to dropout schedule
Alsharid et al. [135]	image captioning	fetal ultrasound images	entropy	training based-batches	easy-to-hard	computational cost due to trainable parameters
Wang et al. [136]	classification	a human facial attribute	loss function	sample selections, weights	easy-to-hard	dependence on dataset characteristics
Li et al. [137]	Segmentation	brain tumour	label distribution	tasks by difficulty	easy-to-hard	Relies on accurate segmentation
Zhao et al. [138]	classification	brain tumour	difficulty of samples	weights	easy-to-hard	computational cost

3.6 Discussion

The related work in this chapter explores different methods to improve classification performance and address the challenges in training DCNNs for medical image classification. As stated in Section 3.2, transfer learning has been widely used and performs well in many tasks. However, its effectiveness is limited in the medical

imaging domain. This is due to major differences between medical and natural images, such as intensity, colour, and texture. Some studies [89, 83, 87] addressed this limitation by pre-training models on labelled data sets from the same medical domain before fine-tuning them for specific tasks. This approach helps to reduce the domain gap and improve performance. However, it heavily depends on the availability of large and diverse domain-specific datasets. In practice, such datasets are difficult to obtain, particularly for rare diseases where data is scarce and annotations are limited.

Moreover, medical image datasets often suffer from a limited number of samples for certain diseases, making it challenging to train DCNN models effectively. To address this, many studies have employed AUG techniques to artificially expand the training set [81, 86, 88, 90]. While augmentation can enhance performance by increasing data diversity, it has limitations. Augmented images are still derived from a limited dataset and do not introduce entirely new or diverse cases. As a result, they replicate existing patterns without reflecting real-world variability. Furthermore, when class distributions overlap, augmentation may worsen class confusion by reinforcing similarities between classes rather than helping the model differentiate them. To address the issue of overlapping distributions, class decomposition has proven to be an effective strategy. As discussed in Section 3.3, previous works have explored various approaches to class decomposition. For example, Vilalta et al. [9] decomposed classes based on probability functions but observed that creating too many clusters could negatively impact prediction accuracy. In contrast, Elyan et al. [98] proved that increasing diversity within the data improved both learning and performance. However, most of these methods were developed using different traditional machine learning models, limiting their applicability to deep learning and medical imaging challenges. To extend these efforts to the deep learning domain, we previously introduced *DeTraC*, which integrates, for the first time, the class decomposition method within CNNs for medical image classification [8].

Another major challenge in training medical imaging models is the limited number of annotated samples, especially for rare conditions. As discussed in Section 3.4, several previous studies have adopted SSL as an alternative to traditional transfer learning, aiming to leverage unlabelled data from the same domain as the downstream task. However, the performance of these SSL approaches often depends heavily on the quality and diversity of data augmentations, which may introduce bias or fail to reflect real-world clinical variations. Additionally, techniques based on contrastive learning or image restoration may not effectively capture the complex spatial and semantic structures of medical images, particularly when disease indicators are subtle or appear in small regions [114].

Curriculum learning has also emerged as a promising strategy and has attracted growing interest in the medical imaging domain. As demonstrated in Table 3.5, recent works have applied curriculum learning in different tasks to improve model

performance. However, many of these studies face notable limitations. For example, some studies rely heavily on the quality of annotated data or clinical expertise, which can introduce bias and limit generalisability [131]. Others are constrained by limited lesion variety or depend on severity labels that are not always available or consistent across datasets [132]. These limitations highlight the need for more robust and adaptive curriculum strategies capable of handling complex datasets and limited annotations that our proposed methods are specifically designed to address.

Building upon these limitations, we employed the *DeTraC* framework to develop deep CNN models that combine class decomposition with SSL and curriculum learning strategies. This integrated approach aims to address critical challenges in medical imaging, including the scarcity of annotated samples and the difficulties posed by irregular class distributions. By leveraging the strengths of each component, our proposed method enhances learning efficiency and generalisation in complex medical classification tasks.

3.7 Summary

In this chapter, we reviewed various methods to improve the classification of medical image datasets and discussed the challenges that arise during the training process in Section 3.6. A major challenge identified was the complexity of working with overlapping class distributions. Data decomposition approaches inspired us to develop more generalised and effective systems capable of addressing the difficulties associated with training medical image datasets, making them more suitable for clinical applications. Furthermore, the chapter highlighted the challenges of the scarcity of large, diverse, and well-annotated datasets, particularly for rare diseases, which complicate the development of accurate models and limit the model's performance. Self-supervised learning has been identified as a promising approach to achieving these goals. Self-supervised learning uses unlabelled data to learn meaningful representations, reducing the reliance on large labelled datasets and improving feature transferability. On the other hand, curriculum learning improves the model's ability to efficiently learn meaningful patterns by structuring the training process, guiding the model from simpler to more complex tasks.

These strategies represent significant progress toward the development of effective solutions for medical image classification. In the next chapter, we present our first contribution, *4S-DT*, and its developed version, *XDecompo*, including the detailed methodology and a comprehensive experimental study.

Chapter 4

Self-Supervised Learning and Class Decomposition Approach for Classification and Explanation

In the previous chapter, we reviewed the literature work in the field of medical image classification, highlighting the previous methodologies used to address the challenges in training medical image datasets. In this chapter, we present our first contribution to this thesis, *4S-DT* and its enhanced version, *XDecompo*. We then provide detailed information on the datasets used in our experimental work. In addition, we introduce a post-hoc explainable AI method to provide insights into the features learnt by the model. Finally, we compare our approach with other state-of-the-art methods. Findings reported in this chapter have been published in [139, 140].

4.1 Overview

In this chapter, we introduce *4S-DT* to enhance the detection of COVID-19 in chest x-ray (CXR) images and address the issue of limited dataset samples. The model was evaluated on two CXR datasets with a small number of COVID-19 cases, referred to as dataset-A and dataset-B. Unlike the parametric nature of *4S-DT*, *XDecompo* benefits from a non-parametric approach to enhance its generalisation capabilities. In the experimental work, *4S-DT* achieved a high accuracy of 97.54% and 99.80% for detecting COVID-19 cases in dataset-A and dataset-B, respectively. Additionally, *XDecompo* achieved accuracies of 96.16% and 94.30% for colorectal cancer and brain tumour images, respectively, outperforming *4S-DT* and other training strategies.

The chapter is organised as follows: Section 4.2 provides an overview of the contribution of this chapter. Section 4.3 introduces the *4S-DT* model and outlines its fundamental elements. In Section 4.4, we present a detailed description of our developed approach, *XDecompo*. Section 4.5 covers our experimental setup and findings. Section 4.6 discusses explainable AI techniques and feature visualisation that we used in our work. Finally, Section 4.7 provides a summary of this chapter and outlines the motivation for the upcoming contribution.

4.2 Introduction

The availability of annotated medical image datasets remains a significant challenge for researchers aiming to achieve high accuracy. While transfer learning can alleviate this issue by transferring knowledge from general image recognition tasks to medical image classification, it often fails to provide a robust solution when there are irregularities in the distribution of the dataset [141]. SSL provides a promising approach by leveraging unsupervised learning tasks to enhance supervised learning objectives, making it particularly valuable when labelled data is limited or certain classes are difficult to obtain. Typically, SSL involves three key steps: pseudo-labelling, pretext task, and downstream task. a) Pseudo-labelling, or self-labelling, generates labels from the data's structure, allowing the model to learn from huge amounts of unlabelled data. b) The pretext task: is conducted in a self-supervised manner and encourages the model to learn meaningful features (such as relationships between patterns, object colours, and textures) by using a pre-trained model as a backbone for initial weights. These learnt features are then fine-tuned for different tasks with small annotated examples. c) The downstream task: uses a smaller labelled dataset and benefits from the features extracted from the pretext model, leading to improved performance and generalisation on an unseen dataset.

Based on these principles, we introduce "Self-Supervised Super Sample Decomposition for Transfer Learning With Application to COVID-19 Detection", known as *(4S-DT)*, to enhance COVID-19 detection from CXR images. *4S-DT* employs the *K*-means clustering algorithm to generate sub-classes for the downstream task, which can be affected by its sensitivity to initial centroid placement and the presence of outliers. To address this, we developed *XDecompo* to automatically determine cluster structures, enhance feature transferability, and improve decomposition quality. *XDecompo* employs the affinity propagation (AP) method to guide the class decomposition approach in the downstream task. This allows the model to effectively define the class boundaries and generate more precise clusters without requiring predefined parameters. In addition, *XDecompo* is supported by a post-hoc explainable component, enabling a deeper understanding of the model's decision by highlighting the important features in a heatmap. First, we used a self-supervised sample decomposition method with a convolutional autoencoder (CAE) to extract features from a large number of samples. The extracted features are then fed into the Density-Based Spatial Clustering of Applications with Noise (DBSCAN) cluster algorithm to generate the pseudo-labels. We used the pre-trained AlexNet network to train the pretext model and classify the pseudo-labels. For downstream decomposition, we employed the AP clustering algorithm to learn salient features and discover the number of clusters without user intervention. Finally, we provided evidence of the effectiveness of our method through an explainable component to give a clear view

of how the model has made decisions to improve the learning process and performance. The performance of our model was evaluated on two different medical image datasets: colorectal cancer histology and brain tumour images. These datasets were selected due to the presence of irregularity issues within classes in the downstream datasets and the large number of unlabelled related images.

The contributions of this chapter are summarised as follows:

- introduced the *4S-DT* model to improve the detection of small annotated COVID-19 cases from CXR images;
- developed a model named *XDecompo*, which automatically learns the boundaries of the class in the downstream datasets using an AP based on the class decomposition mechanism;
- investigate the effectiveness of *XDecompo* in enhancing feature transferability and addressing irregular data distribution issues on two different medical image datasets;
- demonstrate the robustness and effectiveness of *XDecompo* through the use of post-hoc explainable AI methods, such as the Grad-CAM function, to provide insights into the features learnt by the model and enhance its transparency; and
- perform comprehensive quantitative and qualitative experimental evaluations to compare *XDecompo* with *4S-DT* and other related methods, highlighting its performance advantages in the field.

4.3 4S-DT Model

In this section, we describe in detail the *4S-DT* model, which was proposed to overcome the limitations and costs of data annotations for COVID-19 cases by learning visual features from a large set of unlabelled CXR images. As shown in Fig. 4.1, the model starts by extracting feature representations from the unlabelled CXR images using a stacked autoencoder model (SAE) [142]. Second, the features are fed into the DBSCAN clustering algorithm to generate pseudo-labels. Third, the pre-trained ResNet-18 network is used for training the pretext task and classifying the pseudo-labels [74]. Finally, the learnt features from the pretext model are fine-tuned to a downstream task using a parametric clustering algorithm (k-means) to detect small cases of COVID-19 from CXR images. Algorithm 1 provides a detailed description of the process of *4S-DT*.

4.3.1 Sample Decomposition on Unlabelled CXR Images

In SSL, unlabelled data are used for feature extraction by generating pseudo-labels for each sample, allowing models to capture useful representations without the need

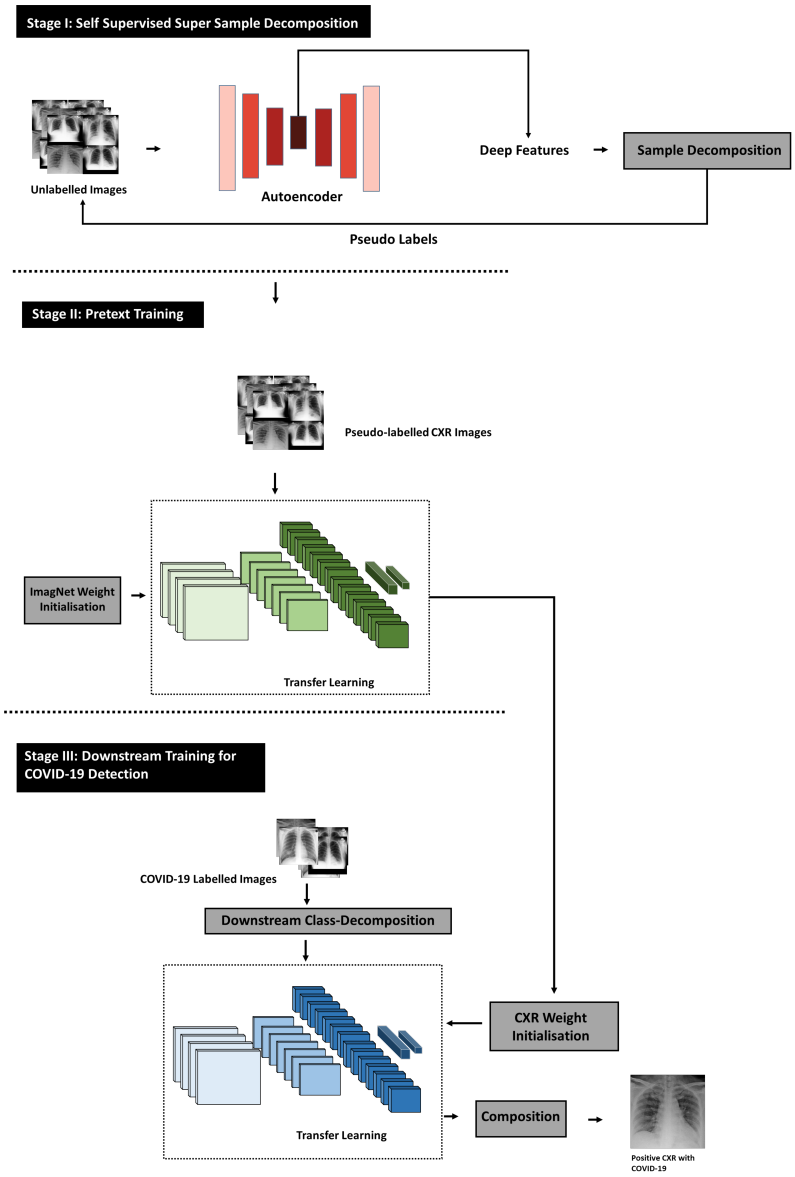

FIGURE 4.1: The framework of the *4S-DT* model. First, SAE is used to extract feature representations from unlabelled CXR images, followed by clustering with the DBSCAN algorithm to generate pseudo-labels. Then, the ResNet-18 pre-trained network is employed to train the pretext task and classify these pseudo-labels. The features learnt from the pretext model are then fine-tuned for the downstream task, which is decomposed using class decomposition guided by the *k*-means clustering method. Finally, error correction is applied to obtain the final prediction.

for manual labelling. The *4S-DT* model starts with using SAE to extract feature representation from an enormous number of CXR images. SAE is an unsupervised learning technique that consists of multiple encoding and decoding layers, where the output of one layer is fed as the input to the next. The encoder layers compress the input data into a lower-dimensional representation, which the decoder then reconstructs to match the original input as much as possible. This process allows the

network to capture and learn hierarchical features within the unlabelled samples. Let the representation vector from SAE denoted as h^d, then the reconstructed input image $\hat{x}$ from the encoder layer can be defined as:
Encoding process

$$h^d = \phi(W^{(1)}x + b^{(1)}), \tag{4.1}$$

Decoding process

$$\hat{x} = \phi(W^{(2)}h^d + b^{(2)}), \tag{4.2}$$

where $W^{(1)}$ and $W^{(2)}$ are the weight matrices for the encoding and decoding layers, respectively, $b^{(1)}$ and $b^{(2)}$ are the bias vectors, and ϕ is non-linear activation function. Once the latent space vector h^d is obtained, it is passed to the DBSCAN cluster algorithm to generate a number of classes C (pseudo-labelled). DBSCAN [143] does not require the number of clusters to be predefined, offering greater flexibility when working with datasets that have unknown or undefined cluster structures. This adaptability is particularly crucial in medical imaging, where data distributions can be irregular and challenging to model. Additionally, DBSCAN can classify noisy points as outliers, which helps ensure that clusters remain clean and more accurate, further improving the quality of the clustering results.

The pseudo-labelled can be defined as $X' = \{(x^i, y^c)|c \in C\}$, where X' is the new dataset after applying the sample decomposition method. The baseline ResNet-18 was adapted for training the pretext model to extract meaningful and informative features from the pseudo-labels that can be applied to downstream tasks in which labelled data is scarce or expensive to obtain. By solving this task, the model learns to recognise patterns, structures, and relationships within the general CXR images, which are beneficial for other tasks.

4.3.2 Class Decomposition with k-means Clustering

Class decomposition uses clustering algorithms to break down complex classes into more homogeneous subgroups. This method enables the model to focus on learning more precise patterns within each subgroup, leading to better generalisation and classification performance. *4S-DT* model uses k-means cluster algorithm to apply the decomposition process. k-means is a popular clustering algorithm due to its computational efficiency and simplicity, making it ideal for large datasets. However, it requires pre-defining the number of clusters (k) beforehand, which can be challenging without prior knowledge of the data structure. Practically, k-means works by identifying similar features among observations and grouping them into clusters. Initially, centroids are randomly selected, and each data point is assigned to the cluster whose centroid is closest. The mean of all points in each cluster is then used to recalculate the new centroids. This process is repeated until the centroids stop changing. The objective function for k-means clustering based on the Sum of Squared Euclidean Distances *(SED)* is expressed as follows:

$$SED = \sum_{j=1}^{k} \sum_{i=1}^{n} \| x_i^{(j)} - c_j \|^2, \tag{4.3}$$

Finally, a class relabelling is used to remap the classification back to the original problem using a simple error correction criterion, see Fig. 1.3 in Chapter 1.

Algorithm 1: 4S-DT *Model*

Input: A large set of unlabelled CXR images, labelled dataset. **Output:** prediction output.
Sample Decomposition:
    Use SAE on the unlabelled images to extract feature representations.
    Apply DBSCAN to generate pseudo-labels.
    Use a pre-trained network to classify the pseudo-labels.
Downstream Decomposition:
    Use K-means clustering to decompose the classes of the labelled dataset.
    Assign new labels to the new dataset.
A coarse transfer learning:
    Adapt the final classification layer of the pretext CNN model to the decomposed classes.
    Fine-tune the learnt weights from the pretext model to the new dataset.
Model Evaluation:
    Evaluate the model on the test set.
    Use error-correction criteria to obtain the final output.

4.3.3 Dataset used in *4S-DT*

4S-DT was introduced to improve the detection of COVID-19 cases from CXR. In the experimental work, we used two different types of data: unlabelled CXR im- ages to extract generic features and a labelled dataset, which contains a small number of COVID-19 samples; see Table 1.1 in Chapter 1. For the labelled dataset, we collected 50,000 CXR images from various sources [24, 25, 26, 27], to ensure diversity in the images and the extracted meaningful features. For the labelled dataset, due to the limited availability of large publicly accessible COVID-19 datasets at that time, we used two different CXR labelled datasets referred to as dataset-A and dataset-B. Dataset-A collected from [23], contains 105 COVID-19 images, while dataset-B available at (`https://www.kaggle.com/prashant268/chest-xray-covid19-pneumonia`), includes 576 COVID-19 images. It should be noted that the CXR images in datasets

A and B are updated over time. Fig. 4.2 shows examples from the dataset used to evaluate *4S-DT*.

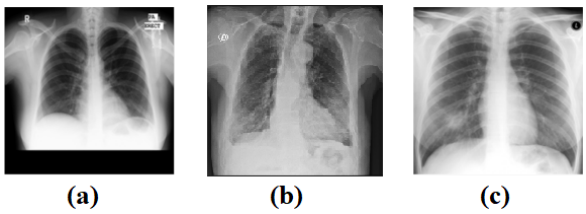

FIGURE 4.2: Examples of the downstream dataset used in the experimental *4S-DT* model, where (a) Normal, (b) COVID-19, and (c) SARS.

4.3.4 Experimental Analysis of *4S-DT*

We investigated the performance of the proposed *4S-DT* framework using three different baseline networks: ResNet-18, GoogleNet, and VGG-19. Due to the limited number of COVID-19 images, all models were trained in deep-tuning mode. In the pretext stage, an SAE model was employed to extract deep features from the unlabelled CXR images, using 80 neurons in the first hidden layer and 50 neurons in the second. The extracted features were then clustered using the DBSCAN algorithm to generate pseudo-labels. ResNet-18 was used for training the pretext task with a mini-batch size of 256 over 200 epochs and a weight decay of 0.0001 to mitigate overfitting. The learnt representations were subsequently transferred to train on the small labelled datasets (Dataset-A and Dataset-B). During this downstream training phase, the learning rate was set to 0.0001, the mini-batch size was 128, the number of epochs was 256, and the weight decay was 0.001. Furthermore, the learning rate was scheduled to drop by a factor of 0.95 every five epochs.

4.3.5 Performance Measures

In the experimental work, all models were evaluated using confusion matrices, which included overall accuracy (ACC), precision (PR), recall (RE), and F1-score (F1) metrics for a multi-class confusion matrix [144]. These metrics provide more insight into the performance of a model by quantifying its quality through various measures and understanding the model's strengths and weaknesses in classification tasks. Accuracy gives an overall measure of how the model correctly predicted the samples. Precision measures the model's effectiveness in correctly identifying negative cases, such as classifying normal images correctly among all actual normal images. Recall, also known as sensitivity or true positive rate, focuses on the model's ability to correctly identify positive cases, such as detecting cancer images among all actual cancer images. Finally, F1-score can be defined as a weighted average of precision and recall, offering a balanced measure of a model's accuracy in identifying positive instances. The confusion matrices are defined as follows:

$$\text{Accuracy}(ACC) = \frac{TP + TN}{TP + TN + FP + FN}, \tag{4.4}$$

$$\text{Precision}(PR) = \frac{TP}{TP + FP}, \tag{4.5}$$

$$\text{Recall}(RE) = \frac{TP}{TP + FN}, \tag{4.6}$$

$$\text{F1-score}(F1) = 2 \times \frac{PR \times RE}{PR + RE} \tag{4.7}$$

Where TP and TN represent the true positive and true negative for a specific class C_i, respectively, while FP and FN refer to the incorrect predictions made by the model for other classes and are defined as:

$$TP_i = \sum_{i=1}^{n} x_{ii} \tag{4.8}$$

$$TN_i = \sum_{j=1}^{c} \sum_{k=1}^{c} x_{jk}, j \neq i, k \neq i \tag{4.9}$$

$$FP_i = \sum_{j=1}^{c} x_{ji}, j \neq i \tag{4.10}$$

$$FN_i = \sum_{j=1}^{c} x_{ij}, j \neq i, \tag{4.11}$$

where x_{ii} is an element in the diagonal of the matrix. In addition, we used the Receiver Operating Characteristic (ROC) curve to provide a visual representation of the model's performance. The ROC curve plots the true positive rate (TPR) or sensitivity against the false positive rate (FPR). We also report the Area Under the ROC Curve (AUC) to summarise the model's performance. A higher AUC value (approaching 1) indicates a highly efficient model, while a lower AUC value (approaching 0) indicates poor performance.

4.3.6 Performance of *4S-DT* Model

To evaluate the performance of *4S-DT*, we used three different baseline networks: ResNet-18, GoogleNet, and VGG19. In addition, *4S-DT* was compared with two different training strategies: traditional transfer learning using the baseline networks and the *DeTraC* model, where the pre-trained network weights are fine-tuned for the downstream task using a class decomposition approach. As shown in Table 4.1, for dataset-A, *4S-DT* achieved the highest performance only when using ResNet-18, with an accuracy of 97.54%, precision of 97.15%, recall of 97.88%, and F1-score of 97.51%. For dataset-B, *4S-DT* outperformed all other training strategies across all baseline networks. The best results were obtained using VGG19, achieving 99.80%, 100%, 99.70%, and 99.84% for ACC, PR, RE, and F1-score, respectively, see Table

4.2. Fig. 4.3 and Fig. 4.4 show the confusion matrix and the ROC curve for *4S-DT*, respectively.

On the other hand, *DeTraC* achieved better performance than traditional transfer learning across different pre-trained networks on both Dataset-A and Dataset-B. This improvement is mainly due to its use of class decomposition, which simplifies complex class structures and reduces overlap between similar categories. The same process is adopted in the *4S-DT* model for the downstream task. However, instead of transferring knowledge from a pre-trained model on a large dataset like ImageNet, which contains images from various domains, *4S-DT* utilises SSL with sample decomposition. This process allows knowledge transfer from features learnt within the same domain, enabling the model to focus on more relevant features, ultimately enhancing performance on downstream tasks.

To further evaluate the effectiveness of *4S-DT*, we applied statistical significance testing using the Wilcoxon signed-rank test with continuity correction [145]. The statistical comparison between *4S-DT* and *DeTraC* produced a p-value of 0.0024, highlighting the substantial performance gain achieved by incorporating SSL and sample decomposition. The statistical comparison between *4S-DT* and *DeTraC* produced a p-value of 0.0024, highlighting the substantial performance gain achieved by incorporating SSL and sample decomposition. Furthermore, comparing the *4S-DT* model to traditional transfer learning resulted in a p-value of 0.0025, confirming that our model significantly outperforms standard transfer learning approaches.

TABLE 4.1: *4S-DT*: Comparison of the performance of *4S-DT* and other models on the COVID-19 (Dataset-A) based on deep tuning mode.

Baseline	Traditional learning				DeTraC				*4S-DT*			
Network	ACC (%)	PR (%)	RE (%)	F1 (%)	ACC (%)	PR (%)	RE (%)	F1 (%)	ACC (%)	PR (%)	RE (%)	F1 (%)
ResNet-18	92.50	94.30	65.01	76.96	95.12	91.87	97.91	94.79	**97.54**	**97.15**	**97.88**	**97.51**
GoogleNet	93.68	91.52	92.59	92.05	94.71	95.76	97.80	96.76	94.15	93.08	97.07	95.03
VGG19	94.59	93.08	91.64	92.35	97.35	96.34	98.23	97.27	95.28	97.15	93.66	95.37

Based on the obtained results, we concluded that the *4S-DT* model significantly improves classification performance and effectively detects a small number of COVID-19 cases by utilising SSL with sample decomposition before transferring knowledge into another small dataset. In addition, *4S-DT* has the ability to handle irregularities within the classes of the downstream dataset by employing the class decomposition method in the downstream dataset. This is achieved by understanding and clarifying the boundaries between classes, leading to more accurate and reliable outcomes. However, using a parameter-based clustering algorithm to detect

TABLE 4.2: *4S-DT*: Comparison of the performance of *4S-DT* and other models on the COVID-19 (Dataset-B) based on deep tuning mode.

Baseline	Traditional learning				DeTraC				*4S-DT*			
Network	ACC (%)	PR (%)	RE (%)	F1 (%)	ACC (%)	PR (%)	RE (%)	F1 (%)	ACC (%)	PR (%)	RE (%)	F1 (%)
ResNet-18	94.74	97.84	93.30	95.52	97.50	98.20	95.50	96.83	**99.60**	**99.90**	**96.50**	**98.17**
GoogleNet	94.43	91.15	88.17	89.63	97.10	99.48	97.41	98.43	**99.20**	**99.70**	**93.90**	**96.71**
VGG19	95.28	94.23	92.74	93.47	98.20	99.40	96.50	97.92	**99.80**	**100**	**99.70**	**99.84**

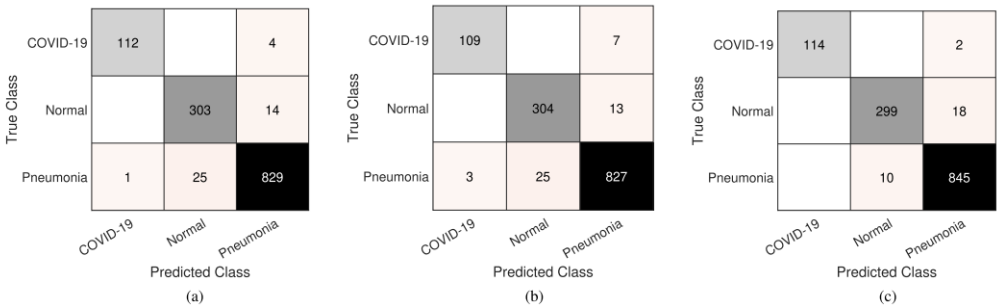

FIGURE 4.3: Confusion matrix of *4S-DT* on COVID-19 dataset-B using different pre-trained networks: (a) ResNet-18, (b) GoogleNet, and (c) VGG19.

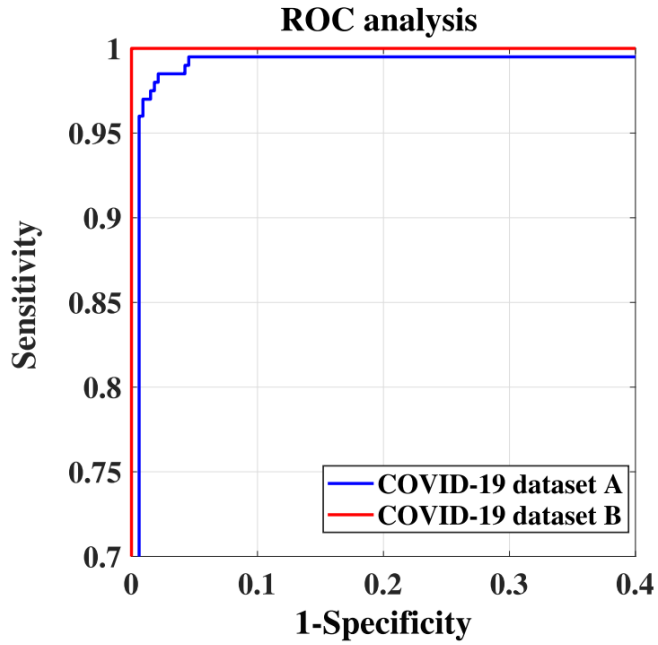

FIGURE 4.4: ROC curve obtained during the training of *4S-DT* on the COVID-19 dataset-A and COVID-19 dataset-B.

the number of sub-classes in the downstream dataset can directly impact the quality of these sub-classes. This, in turn, affects the quality of the features transferability. In the next section, we introduce our first contribution method, *XDecompo*, which is designed to overcome the limitations associated with parameter-based clustering algorithms, thus providing a more robust solution for handling sub-class identification and data irregularities.

4.4 XDecompo Model

This section provides a detailed explanation of the developed *XDecompo* model, including its framework architecture and the key modifications made to the *4S-DT* model. *XDecompo* is designed to address the limitation of *4S-DT* and enhance decomposition quality through an automated clustering algorithm. Compared to the *4S-DT* model, *XDecompo* provides a better generalisation and feature visualisation due to the non-parametric nature of its class decomposition approach.

As shown in Fig. 4.5, *XDecompo* consists of four main stages. First, we used a CAE to extract feature representations from the unlabelled images, which were then clustered using the DBSCAN algorithm to generate pseudo-labels. Next, the pre-trained ResNet-50 network is employed to train the pretext task and classify the pseudo-labels. Third, the learnt features from the pretext model are fine-tuned for the downstream task, where class decomposition is guided by the AP clustering method. In addition, *XDecompo* incorporates the Grad-CAM algorithm as a post hoc explainable method to highlight the contribution of each pixel in the input images toward the model's final prediction, providing insights into the feature robustness and transferability. Algorithm 2 provides a detailed description of the process of *XDecompo*.

4.4.1 Feature Extraction with CAE

In this stage, we extracted deep local features from a large number of unlabelled images using CAE. The choice of CAE in *XDecompo* comes from the ability of convolutional layers to capture local spatial features by scanning the entire image with convolutional filters, which is particularly important for medical images [146]. In contrast, SAE, which uses fully connected layers, focuses on learning global patterns and may not preserve spatial relationships as effectively. This makes CAEs more suitable for tasks where spatial structures are vital for accurate feature extraction, such as in medical imaging.

CAE consists of two blocks: the encoder and the decoder layers. The encoder block is made up of multiple convolutional layers with a non-linear activation function and a pooling layer that downsamples the input image. The decoder block is responsible for reconstructing the input image, bringing it as close as possible to the original. The 2D convolution operation can be defined as:

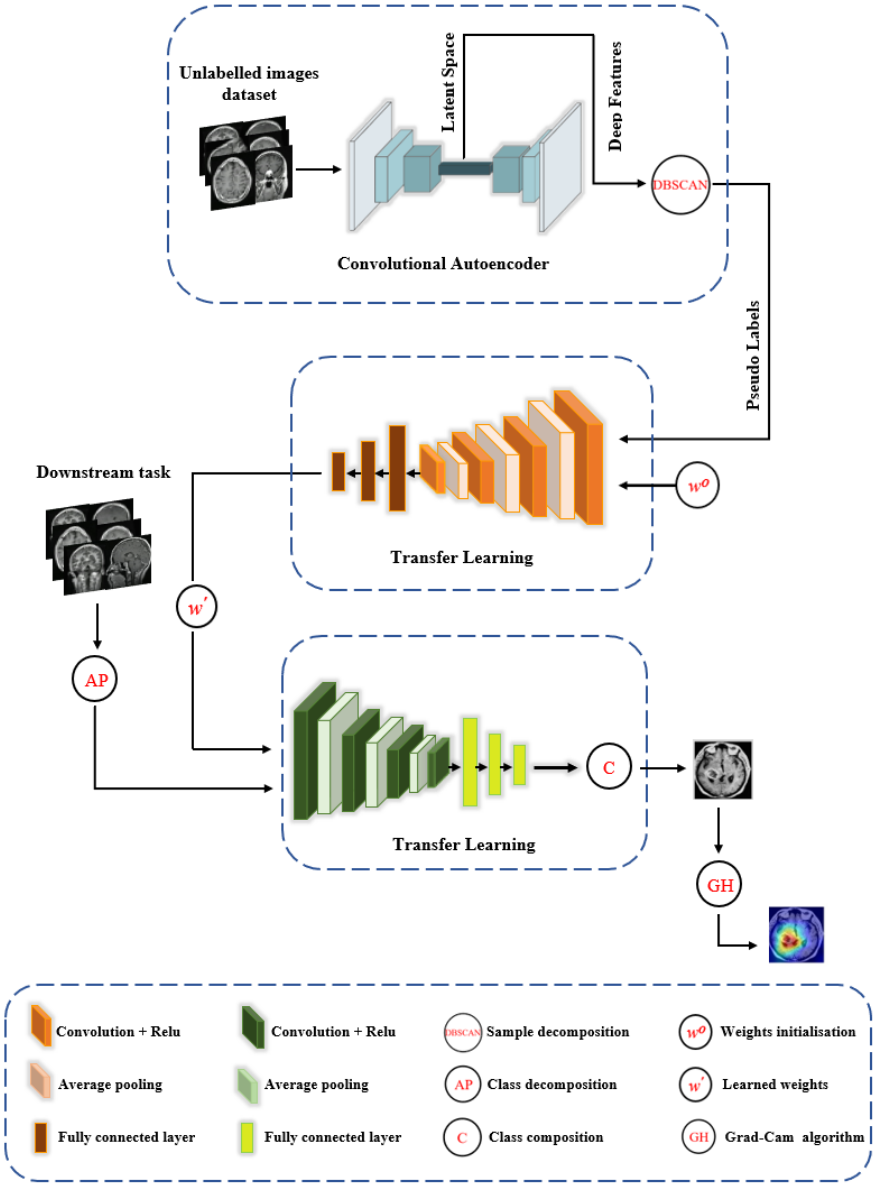

FIGURE 4.5: The framework of the *XDecompo* model. First, we use a CAE to extract feature representations from unlabelled images, followed by clustering with the DBSCAN algorithm to generate pseudo-labels. Next, a pre-trained ResNet-50 network is employed to train the pretext task and classify these pseudo-labels. The features learnt from the pretext model are then fine-tuned for the downstream task, which is decomposed using class decomposition guided by the AP clustering method. Finally, error correction is applied to obtain the final prediction. Additionally, *XDecompo* integrates a post-hoc explanation tool that highlights the contribution of each pixel in the input images to the model's final prediction.

$$A(i,j) = \sum_{u=-2f-1}^{2f+1} \sum_{v=-2f-1}^{2f+1} x(i-u, j-v)\, w(u,v) + b_{ij}, \tag{4.12}$$

where $A(i,j)$ is the output activation map in position (i,j), x is the input image,

and w is the weights of a square convolution filter with dimension $(2f+1, 2f+1)$. For d depth, the produced activation maps from the input x can be defined as:

$$A^d = \sigma\left(x \times W^d + b^d\right), \tag{4.13}$$

where **œ** is an activation function and b^d is the bias for d-th activation maps. The reconstructed image $\hat{x}$ is obtained by:

$$\hat{x} = \sigma\left(\sum_{d \in H} \hat{A}^d \times \hat{W}^d + b\right), \tag{4.14}$$

where H refers to a set of activation maps and $\hat{W}$ represents the inversion process applied to both dimensions of the weights. The DBSCAN clustering was then employed to generate pseudo-labels for the data. DBSCAN is an unsupervised clustering algorithm that groups data points into a single cluster by looking at the local density of the data points, so it is considered robust to real-life data which may contain noisy points and outliers [143].

DBSCAN is sensitive to the value of two parameters that can significantly impact the outcomes: Epsilon *(Eps)*, which represents the radius of the neighbourhoods around a data point x, and *MinPts* refers to the minimum number of data points/observations (neighbours) within that radius. Generally, *MinPts* can be computed from the dimensions (D) of the dataset as MinPts >= D+1, and the Eps-neighbourhood can be defined as:

$$N_{Eps}(x_j) = \{x_i \in X | dis(x_i, x_j) \leq Eps\}. \tag{4.15}$$

This process results in c clusters, where each cluster is formed by maximising the density reachability relationships among images. The resulting c cluster labels are then assigned to the n' unlabelled images, which will serve as pseudo-labels for the pretext training task in the self-supervised learning mechanism. The pseudo-labelled image dataset for the pretext task can be formally represented as:

$$X' = \{(x^i, y^c) | x^i \in X, y^c \in C, c \in \{1, 2, ..., c\}\} \tag{4.16}$$

Where x^i refers to an image from the unlabelled dataset and y^c represents its corresponding pseudo-label (cluster assignment). These labels are then used to train the model in a self-supervised manner, allowing it to learn meaningful feature representations for downstream tasks.

4.4.2 Pretext Training

The next stage involves training the pretext model using the baseline ResNet-50 network to classify the pseudo-labelled images. The ResNet-50 network, known for

its deep architecture with residual connections, provides an effective mechanism for training very deep networks. These residual connections help mitigate the vanishing gradient problem, ensuring that gradients can flow through the network more efficiently during the back-propagation process. During training, the model learns meaningful patterns and representations from the pseudo-labelled images, which are crucial for fine-tuning the model for the downstream task. This process improves the model's ability to generalise by allowing it to capture the relationships and complex features relevant to the specific context of the task. As a result, training the downstream task becomes more efficient in making accurate predictions, leveraging the knowledge gained during pretext training to adapt effectively to new data and tasks.

4.4.3 Class Decomposition with AP

The class decomposition method can be understood as breaking down the original classes of a dataset into smaller sub-classes for better model performance. Let the dataset (D) contain pairs of data points (x, y), where y is the corresponding label, and C is the number of classes. The dataset can be represented as:

$$D = \{(x_j, y_i) | \forall j \in [1, N], y_i \in \{1, 2, ..., C\}\} \tag{4.17}$$

After decomposition, each original class y_i is split into k_i sub-classes and can be defined as $D(y_i) = \{y_{i1}, y_{i2}, ..., y_{ik_i}\}$. As a result, the new dataset D' containing these sub-classes, is defined as:

$$D' = \{(x_j, y_{ij}) | \forall j \in [1, N], y_{ij} \in D(y_i)\} \tag{4.18}$$

where N is the total number of samples in the dataset, and y_{ij} represents the sub-classes of the parent class y_i.

XDecompo utilises the *AP* [147] clustering method to execute the decomposition process for the downstream task. The *AP* algorithm, an unsupervised technique, works by passing messages between data points and does not require the predefinition of the number of clusters. It determines how well the (j-th) point serves as an exemplar for the (i-th) point by alternating between two message-passing updates, named the responsibility and availability matrices, and can be defined as:

Responsibility matrix:

$$\rho(i, k) = sE(i, k) - max\left\{\alpha\left(i, k'\right) + sE\left(i, k'\right) \forall k' \neq k\right\}, \tag{4.19}$$

Availability matrix:

$$a(i, k) = min\left(0, \rho(k, k) + \sum_{i' \notin \{i,k\}} max\left(0, \rho(i', k)\right)\right) i \neq k, \tag{4.20}$$

where the responsibility matrix measures how well the point x_k serves as an exemplar for x_i compared to other candidate exemplars for x_i. On the other hand, the availability matrix considers the appropriateness of x_i choosing x_k as its exemplar based on how many other data points also favour x_k as their exemplar.

Error correction prediction:

Once the model is trained on the decomposed dataset D', the model's predictions at the sub-class level are mapped back to their respective original classes to obtain the final classification results, see Fig. 1.3 in Chapter 1. The output predictions are merged back into the original class y_i, yielding the final classification result. Given a model prediction $\hat{y}_{ij}$ for a data point x_j, we define the recombination function E(.) as:

$$E(\hat{y}_{ij}) = y_i \tag{4.21}$$

We used the cosine similarity measure for AP to learn the boundary between certain features within each class. Cosine similarity is a structural similarity measure based on the idea that two vectors (X_i, X_j) are supposed to be similar if they have many neighbours in common, where a similarity of 0 indicates that the vector orientation is completely different, while a similarity of 1 indicates that the vector orientation is the same. A similarity of 0 means a completely different vector orientation, and a similarity of 1 means that the vector orientation is the same.

4.4.4 Explainable Techniques in Machine Learning

In healthcare, the black-box nature of deep learning models presents a significant challenge to understanding and trusting their decision-making processes [148]. As deep learning techniques become more widely used in clinical tasks, the demand for interpretability methods develops, ensuring that users understand the underlying mechanics driving the model's predictions. Explainable Artificial Intelligence (XAI) improves the openness and reliability of these models by providing insights into how and why specific decisions are made [149, 150, 151]. In healthcare, XAI techniques enable professionals to provide qualitative explanations, such as visualisations of important elements, which can aid in the justification and validation of model outputs [152]. This not only increases trust in AI-driven choices but also promotes the wider adoption of deep learning solutions in medical practice [153, 154]. For example, Maqsood et al. [155] used an XAI function to visualise the final predictions for brain tumour detection. Esmaeili et al. [156] employed explainable AI to detect and interpret early-stage brain tumours in CMR images, providing insights into the performance of three DL models and aiding in the selection of optimal training strategies. Wang et al. [157] introduced COVID-Net to detect COVID-19 in CXR images, employing an explainability method to ensure transparent decision-making

and provide insights into COVID-19 features to assist physicians. Similarly, Bhandari et al. [158] proposed a DCNN for detecting lung diseases, including COVID-19, pneumonia, and tuberculosis, using CXR images. The model's predictions were interpreted by different XAI algorithms, ensuring transparency and providing valuable insights to radiologists. Sabol et al. [159] developed an XAI-based system, CFCMC, for classifying eight tissue types from histopathological cancer image samples. The model is designed to assist medical experts rather than fully automate the diagnostic process.

The visualisation of explainable AI can be achieved using various attention mechanisms, including trainable, post-hoc, soft, and hard attention methods [160, 161]. Trainable attention is incorporated during the model's learning phase, helping the network concentrate on important regions of the image. In contrast, post-hoc attention is applied after training with fixed weights to generate heatmaps such as occlusion [149], saliency [162], CAM [163], or Grad-CAM [164] maps. In this work, we employed the Grad-CAM algorithm to identify specific patterns in the input images that guide the predictions made by the *XDecompo* model using an activation heatmap [165]. The core concept of Grad-CAM is that the weights of a convolutional layer are determined by computing the gradient of the classification score ∂x^c for a given class c with respect to the activation map ∂A^d of the d-th feature map. The importance of each feature map's neurons is then derived by performing a global average pooling of the gradients at position (i, j) and defined as:

$$\varphi_d^c = \frac{1}{m} \sum_i \sum_j \frac{\partial x^c}{\partial A_{ij}^d}, \tag{4.22}$$

where m represents the total number of pixels in A^d. To generate the final Grad-CAM heatmap, the ReLU activation function is applied to the sum of the products between φ_d^c and the corresponding feature map A^d as defined in the following equation:

$$H_{Grad-CAM}^c = ReLU\left(\sum_d \varphi_d^c A^d\right) \tag{4.23}$$

Algorithm 2: XDecompo *Model*

Input: a large set of unlabelled images, a labelled dataset.
Output: prediction output.
Feature Extraction:
    Use CAE on the unlabelled images to extract feature representations.
    Apply DBSCAN to generate pseudo-labels.
Pretext Training:
    Use a pre-trained network to classify the pseudo-labels.
Downstream Decomposition:
    Use AP to decompose the classes of the labelled dataset.
    Assign new labels to the new dataset.
    Fine-tune the learnt weights from the pretext model to the new dataset.
    Evaluate the predicted value on the test set.
    Refine the final classification using error-correction criteria.
Explainable AI:
    Apply an XAI technique to interpret the classification decisions.

4.5 Experimental Setup and Results

This section discusses the datasets used in our experiments and the evaluation metrics employed to assess classification performance. In addition, we discuss the results of *XDecompo* model and compare its performance with the *4S-DT*, *DeTraC* models, and other related work in the field. Finally, we demonstrate the model's ability to highlight significant features through heatmaps, providing a visual representation of the learnt patterns.

4.5.1 Datasets Collection

In this experimental study, two medical image datasets were used: colorectal cancer data (CRC) and brain tumour datasets. *XDecompo* leverages two types of data: unlabelled data for generating pseudo-labels and extracting rich information through training a pretext model, and labelled data for training and evaluating the downstream model.

As mentioned in Chapter 1, Section 1.7, the dataset "NCT-CRC-HE-100K" was selected as the unlabelled data, containing 100,000 samples. The "CRC-VAL-HE-7K" dataset was used as the labelled dataset, containing 7,180 image patches across nine unbalanced classes, with all patches sized at 224×224 pixels at 0.5 microns per pixel. For our experiment, we used three classes: Adipose (ADI), stroma (STR),

and tumour epithelium (TUM), containing 1,338, 421, and 1,233 samples, respectively. Fig. 4.6 represents examples of the three classes we used from the CRC dataset. Similarly, as summarised in Section 1.7, the brain tumour dataset includes both labelled and unlabelled samples. The unlabelled samples were collected from a publicly available source (https://www.kaggle.com/datasets/navoneel/brain-mri-images-for-brain-tumor-detection). Several data augmentation processes, including reflection, shifting, wrapping, and rotation at various angles, were applied to increase the sample size, resulting in 45,960 brain tumour images. The labelled dataset was collected from [30], dividing into three classes: 1,426 glioma, 708 meningioma, and 930 pituitary tumour images, all sized at 400 × 400 pixels, see Fig. 4.7. Each dataset was randomly divided into 60% for training, 20% for validation, and 20% for testing. *XDecompo* was evaluated using 268 ADI, 84 STR, and 247 TUM samples from the CRC dataset and 1,426 glioma, 708 meningioma, and 930 pituitary tumour samples from the brain tumour dataset.

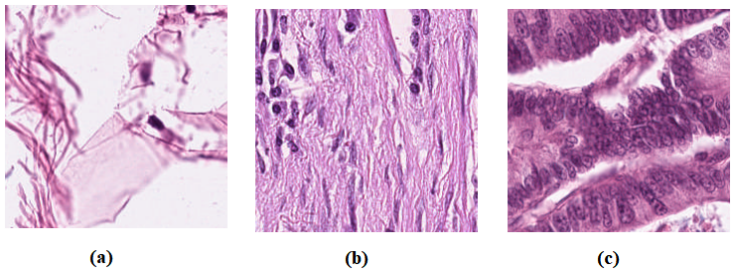

FIGURE 4.6: Example patch images from the CRC-VAL-HE-7K colorectal cancer dataset used in our experiment: (a) ADI, (b) STR, and (c) TUM.

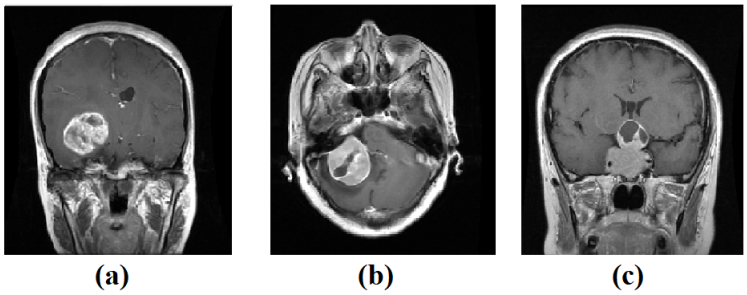

FIGURE 4.7: Example images from the brain tumour test set: a) glioma, b) meningioma, c) pituitary tumour.

4.5.2 Hyperparameter Settings

To extract deep features from the unlabelled dataset and generate pseudo-labels for *XDecompo*, we built a CAE model comprising two convolutional layers with a kernel size of 3 pixels and ReLU activation. For the histological dataset, the first layer included 64 filters, while the second layer used 32. In contrast, for the brain tumour

image dataset, the first and second layers were configured with 32 and 16 filters, respectively, see Fig. 4.8. Our study also compares *XDecompo* with the *4S-DT* model, which employs an SAE with 600 neurons in the first hidden layer, 400 in the second, and 200 for latent space representation; see Fig. 4.9. As shown in Fig. 4.9 and Fig. 4.8, SAE struggles to preserve fine details and spatial structures. On the other hand, CAE highlights its ability to reconstruct the input image with significantly better preservation of spatial structures and finer details, bringing it much closer to the original image. This ability to maintain important features makes CAE more suitable for medical image analysis.

Then, the latent features extracted from autoencoder models are fed into the DBSCAN clustering algorithm to generate pseudo-labels. Both the SAE and CAE models were trained using learning rates of 0.001 and 0.0001, respectively, with a mini-batch size of 128, a minimum of 100 epochs, and a learning rate decay of 0.9 every 10 epochs. This process generated 4 and 2 classes for the CRC and brain tumour datasets, respectively. While the SAE model produced 8 classes for the CRC dataset and 6 for the brain tumour dataset.

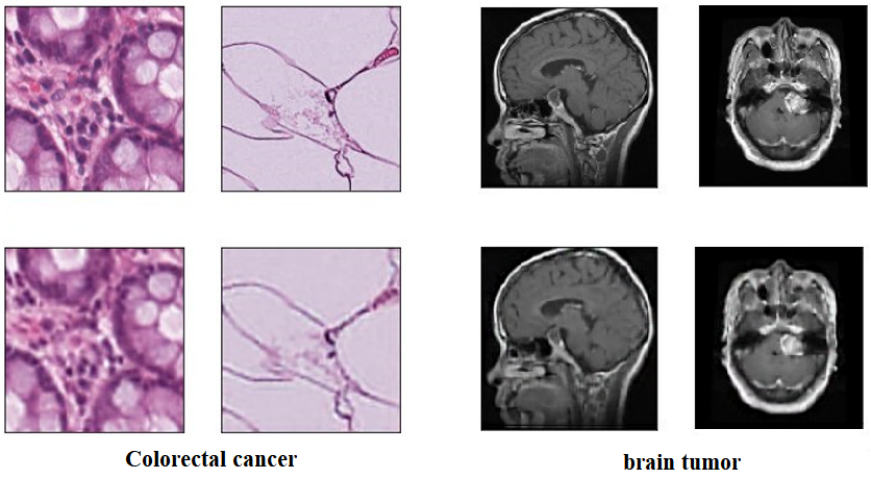

FIGURE 4.8: The CAE for unlabelled images; first row: the original images of the dataset, second row: the reconstructed images.

For each labelled dataset, we used the AlexNet pre-trained network as a feature extractor to extract the discriminative features between classes. We set the learning rate to 0.0001, which decreased by a factor of 0.9 every 3 epochs for 100 training epochs with a batch size of 128. Then these features were fed to clustering algorithms to create sub-classes. For *XDecompo*, the AP technique was used with the cosine similarity metric. The damping factor was set at 0.9 for the CRC dataset and 0.85 for the brain tumour dataset, with a maximum of 1,000 iterations and a convergence parameter of 50. In contrast, for the *4S-DT* model, we used the k-means clustering algorithm with k set to 2. Tables 4.3 and 4.4 illustrate the outcomes of this process for CRC and brain tumour datasets, respectively. As shown in the tables, in the

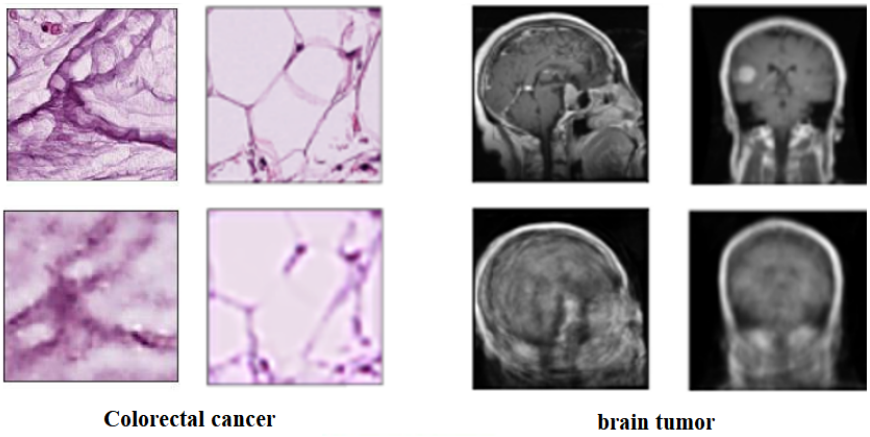

FIGURE 4.9: The SAE for unlabelled images; first row: the original images of the dataset, second row: the reconstructed images.

AP clustering method, each downstream dataset was divided without predefined parameters, leading to a random division of the original classes.

For training the downstream datasets, we adapted transfer learning based on fine-tuning mode, so the model started training from the last fully connected layer (FC) until the block named Conv5-x, which contains three residual layers.

Based on trial and error experiments, the CRC dataset was trained with a learning rate of 0.0001 for the CNN layers, with a learning rate decay of 0.95 every 5 epochs and a mini-batch size of 50. For the brain tumour dataset, a similar learning rate of 0.0001 was used, with a decay schedule of 0.95 every 4 epochs. To reduce overfitting, we applied L2 regularisation with a value of 0.001. The learning rate for the final fully connected layer was set to 0.01, as this layer focuses on the classification task rather than learning general features, like the earlier layers. Additionally, the output layer was modified to match the number of classes in each dataset.

TABLE 4.3: The number of instances before and after applying the class decomposition on the CRC data set using k-means and AP clustering algorithms.

Method	Original dataset	ADI		STR		TUM	
	# instances	1070		337		986	
k-means	Decomposed dataset	ADI_1	ADI_2	STR_1	STR_2	TUM_1	TUM_2
	# instances	666	404	171	166	406	580

AP	Decomposed dataset	ADI_1	ADI_2	ADI_3	ADI_4	STR_1	STR_2	TUM_1	TUM_2	TUM_3
	# instances	377	270	222	201	171	166	381	371	234

4.5.3 Performance Measures

We adopt accuracy, precision, recall, and F1-score, which were defined in Section 4.3.5. In addition, we plot the ROC curve to provide a visual representation of the model's performance.

TABLE 4.4: The number of instances before and after applying the class decomposition on the brain tumour dataset using *k*-means and AP clustering algorithms.

Method	Original dataset	glioma		meningioma		pituitary tumour	
	# instances	1140		565		744	
k-means	Decomposed dataset	GLI_1	GLI_2	MEN_1	MEN_2	PIT_1	PIT_2
	# instances	577	563	298	267	426	318

AP	Decomposed dataset	GLI-1	GLI-2	GLI-3	MEN-1	MEN-2	PIT-1	PIT-2	PIT-3
	# instances	455	529	156	290	275	214	322	208

4.5.4 Performance of *XDecompo* Model

We evaluated *XDecompo* on two datasets: CRC and brain tumour, using a fine-tuning strategy. The model was trained based on fine-tuning four layers, and results are summarised in the last column in Table 4.5 and Table 4.6 for the CRC and brain tumour datasets, respectively. For CRC dataset, *XDecompo* achieved a significant ACC of 96.16%, with a PR of 97.82% and RE of 90.87%, 94.22% for F1-score for classifying 599 test images. Similarly, Table 4.6 reports the performance on the brain tumour dataset, where *XDecompo* also achieved a higher accuracy with 94.30%, 97.04% for PR, 93.27% for RE, and 95.12% for F1-score on 615 brain tumours images.

4.5.5 Ablation Study

We conducted an ablation study comparing *XDecompo* with *4S-DT*, and *DeTraC*, where the pre-trained network weights are fine-tuned as a backbone for training the downstream task, in which class decomposition is applied. The models were also compared with traditional transfer learning using ResNet-50. As shown in Table 4.5, *XDeCompo* outperformed *4S-DT*, which achieved 92.65%, 95.83%, 82.54%, and 88.69% for ACC, PR, RE, and F1-score, respectively. Likewise, for the brain tumour, *4S-DT* achieved 92.84% for ACC, 96.40% for PR, 92.05% for RE, and 94.17% for F1-score, which are lower than the results obtained by *XDecompo*.

On the other hand, the performance of *DeTraC* is lower than both *4S-DT* and *XDecompo*, achieving accuracy of 91.31% and 89.91% on the CRC and brain tumour datasets, respectively. In addition, transfer learning with ResNet-50 achieved the lowest performance on the CRC and brain tumour datasets with accuracies of 90.31% and 87.96%, respectively.

These findings demonstrate that *XDecompo* can improve the feature transferability, which plays a critical role in improving model generalisation compared to other models. Furthermore, the comparison results demonstrate that *4S-DT* outperforms *DeTraC* in some cases but does not reach the same performance levels as *XDecompo*. Transfer learning with ResNet-50 consistently showed the lowest performance across both datasets, indicating that with small datasets, traditional transfer

learning may lead to poor generalisation, especially when the dataset distribution is irregular.

The confusion matrices for each model are represented in Fig. 4.10 and Fig. 4.11 for the CRC and brain tumour datasets, respectively. In addition, Fig. 4.12 and Fig. 4.13 show the AUC for each class across all models, where *XDecompo* achieves the highest AUC values for each class on the test sets.

We also measured statistical significance using the Wilcoxon signed-rank test with continuity correction [145] to validate the results. At a 0.05 significance level, *XDecompo* achieved statistically significant improvements on the brain tumour dataset compared to *4S-DT* (p = 0.00167), DeTraC (p = 0.0012), and traditional transfer learning (p = 0.00097). Similarly, for the CRC dataset, *XDecompo* outperformed *4S-DT* (p = 0.00124), *DeTraC* (p = 0.0015), and traditional transfer learning (p = 0.00038). These results demonstrate that *XDecompo* with a non-parametric clustering method leads to better feature transferability and improved model performance.

TABLE 4.5: *XDecompo*: Classification performance of each model on the test set of the CRC dataset.

Layer	Traditional training (ResNet-50)				DeTraC				*4S-DT*				*XDecompo*			
Name	ACC (%)	PR (%)	RE (%)	F1 (%)	ACC (%)	PR (%)	RE (%)	F1 (%)	ACC (%)	PR (%)	RE (%)	F1 (%)	ACC (%)	PR (%)	RE (%)	F1 (%)
FC	90.81	94.79	78.17	85.68	91.48	95.17	80.34	87.13	92.82	95.92	82.93	88.95	**95.49**	**97.44**	**89.28**	**93.18**
Conv5-3	90.65	94.69	77.67	85.33	92.15	95.54	81.34	87.87	92.32	95.64	81.74	88.14	**94.82**	**97.06**	**87.97**	**92.29**
Conv5-2	90.65	94.69	77.67	85.33	91.65	95.26	80.51	87.27	92.98	96.02	83.33	89.23	**94.99**	**97.15**	**88.36**	**92.55**
Conv5-1	90.31	94.50	76.98	84.85	91.31	95.07	79.63	86.67	92.65	95.83	82.54	88.69	**96.16**	**97.82**	**90.87**	**94.22**

TABLE 4.6: *XDecompo*: Overall classification performance of each model on a testing set of the brain tumour dataset.

Layer	Traditional training (ResNet-50)				DeTraC				*4S-DT*				*XDecompo*			
Name	ACC (%)	PR (%)	RE (%)	F1 (%)	ACC (%)	PR (%)	RE (%)	F1 (%)	ACC (%)	PR (%)	RE (%)	F1 (%)	ACC (%)	PR (%)	RE (%)	F1 (%)
FC	87.31	93.74	86.16	89.79	89.91	94.97	88.84	91.80	91.70	95.82	90.95	93.32	**92.52**	**96.15**	**91.62**	**93.83**
Conv5-3	86.29	93.75	87.01	90.25	90.24	95.36	90.21	92.71	92.52	96.32	91.99	94.11	**92.19**	**96.03**	**91.71**	**93.82**
Conv5-2	86.67	93.62	86.71	90.03	89.26	95.01	89.66	92.26	93.00	96.43	92.12	94.23	**92.84**	**96.15**	**91.52**	**93.78**
Conv5-1	87.96	93.84	86.57	90.06	89.91	95.19	90.14	92.60	92.84	96.40	92.05	94.17	**94.30**	**97.04**	**93.27**	**95.12**

4.5.6 Comparison with State-of-the-art Methods

To demonstrate the effectiveness of our model, we compared *XDecompo* with other state-of-the-art DCNN models that have used the same datasets but with different experimental settings. Tables 4.7 and 4.8 show the comparison of *XDecompo* with

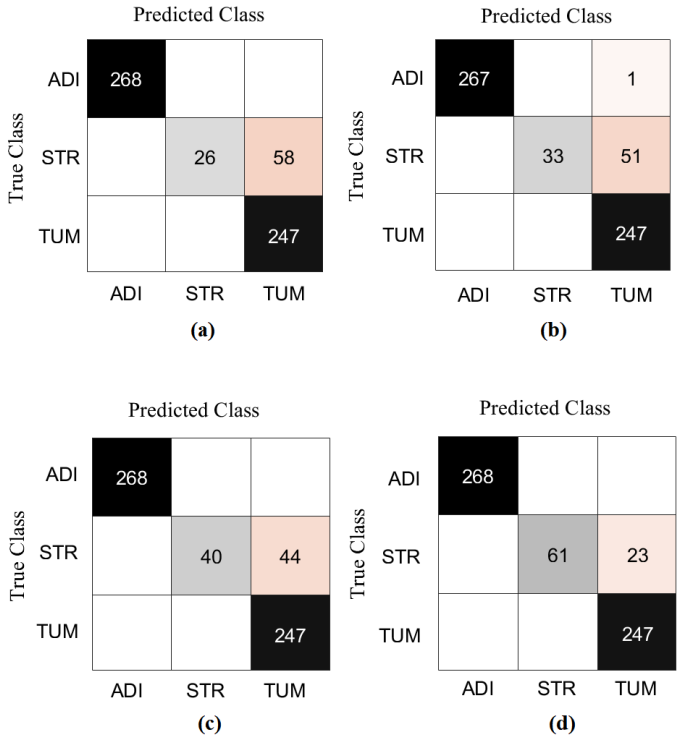

FIGURE 4.10: The confusion matrix results of the CRC dataset obtained by: a) ResNet-50 pre-trained network, b) DeTraC, c) *4S-DT*, and d) *XDecompo*.

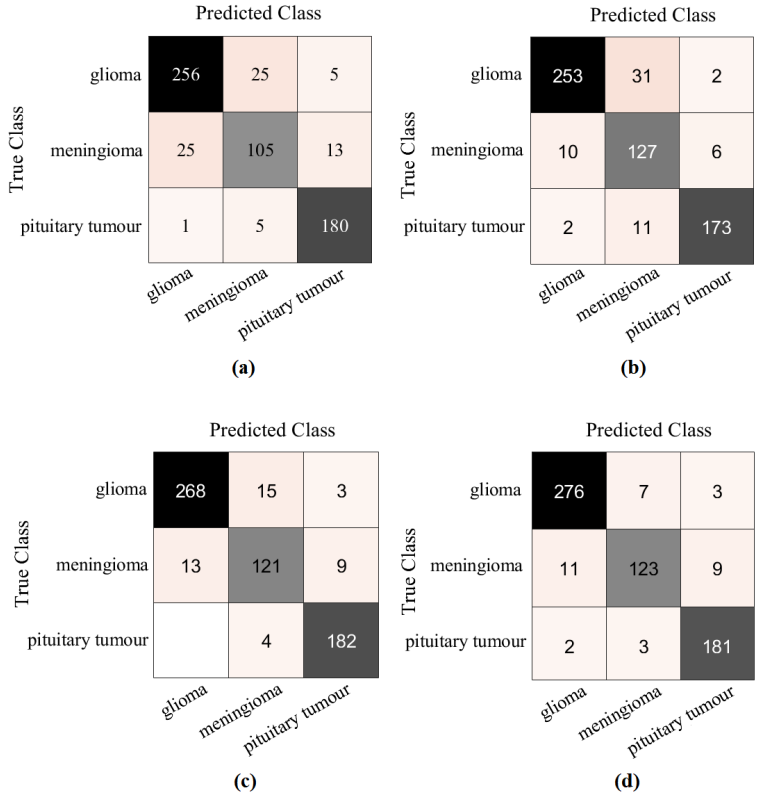

FIGURE 4.11: The confusion matrix results of the brain tumour dataset obtained by: a) ResNet-50 pre-trained network, b) DeTraC, c) *4S-DT*, and d) *XDecompo*.

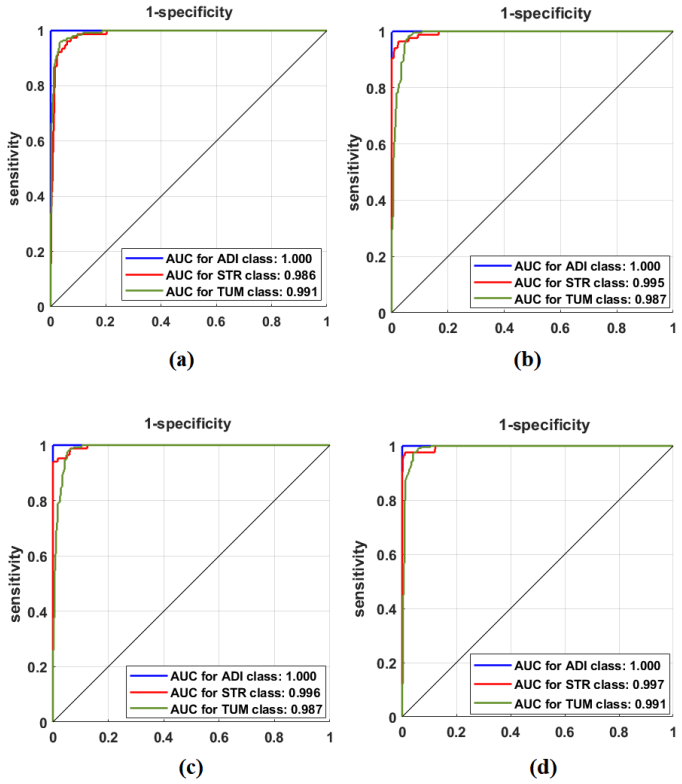

FIGURE 4.12: ROC analysis of the CRC test set obtained by: a) ResNet-50 pre-trained network, b) DeTraC, c) *4S-DT*, and d) *XDecompo*.

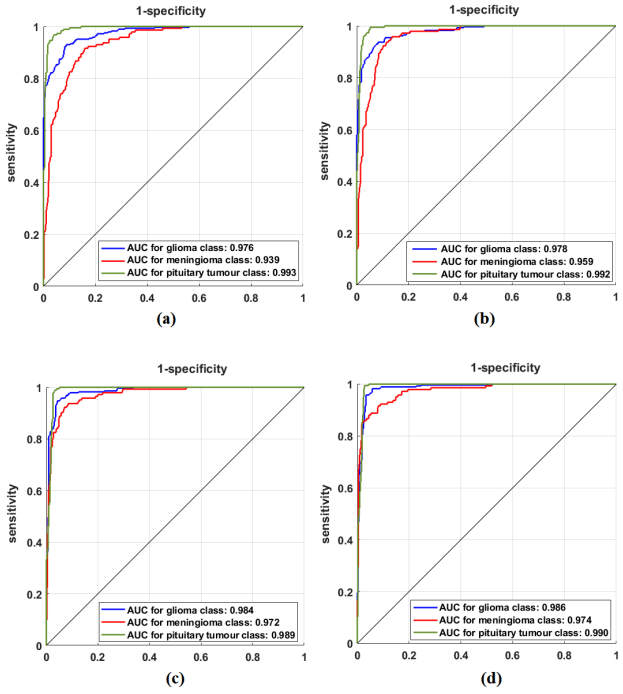

FIGURE 4.13: ROC analysis of the brain tumour test set obtained by: a) ResNet-50 pre-trained network, b) DeTraC, c) *4S-DT*, and d) *XDecompo*.

other methods on the CRC and brain tumour datasets, respectively. In terms of accuracy, *XDecompo* outperformed the other models, achieving superior results after fine-tuning only the weights of the last four layers. For CRC images, Peng et al. [166] employed a K-nearest neighbour-based method for histopathology image classification and retrieval. However, their reliance on manual expert validation makes their method time-consuming and limits its generalisation to unseen datasets. Similarly, Ghosh et al. [167] proposed an ensemble method that combines two large labelled datasets to improve model performance on unseen data, which may not be practical for datasets with smaller samples. Li et al. [168] introduced the DeepDisMISL, combining patches from two CRC datasets to improve output prediction. However, this method may not generalise well to datasets with different characteristics due to selection bias. In contrast, our work seeks to mitigate such biases by enabling the model to learn meaningful features from unlabelled data, allowing for more flexible adaptation to diverse datasets. Moreover, Kather et al. [169] used a transfer learning strategy with different pre-trained networks on training a large labelled dataset, then evaluated it on another small dataset, relying entirely on a supervised learning task, which limits its generalisability, particularly for medical imaging tasks where annotations are costly and scarce.

Regarding brain tumour datasets, Abiwinanda et al. [170] and Afshar et al. [171] designed their custom CNN models from scratch, but their approaches depend heavily on trial-and-error-based hyperparameter tuning, huge annotated samples, and resource consumption. Cheng et al. [172] applied data augmentation techniques on tumour regions and used different statistical feature extraction methods, followed by SVM for classification. While these techniques can improve performance, they are limited by relying on the performance of handcrafted features, which may not fully capture the complex patterns in medical images as effectively as CNNs. In addition, [173] used CNNs as feature extractors with different classifiers, which may restrict the model's ability to leverage backpropagation for faster convergence and improved performance. Tazin et al. [174] compared the performance of three pre-trained models, utilising various data augmentation techniques alongside transfer learning to enhance accuracy. However, they did not evaluate their method without relying on such preprocessing, leaving uncertainty about whether the improvements stem from the model architecture itself or the preprocessing techniques. The results are summarised in Tables 4.7 and 4.8, demonstrating that *XDecompo* outperforms previous methods by addressing their limitations in reducing dependence on large labelled datasets and improving the generalisation.

TABLE 4.7: Comparing the performance of several approaches and *XDecompo* for classifying the CRC dataset.

Ref.	Method	ACC (%)
[166]	Multitask ResNet-18	95.0
[166]	CNN-ResNet-50	93.60
[167]	Ensemble DNN	92.83
[168]	CNN-Xception	94.4
[169]	CNN-VGG19	94.3
[140]	***XDecompo***	**96.16**

TABLE 4.8: Comparing the performance of several approaches and *XDecompo* for classifying the brain tumour dataset.

Ref.	Method	ACC (%)
[170]	7-layered CNN	84.19
[172]	BoW + SVM	91.28
[173]	CNN + KELM	93.68
[174]	CNN-transfer learning	92.00
[171]	CapsNet	90.89
[140]	***XDecompo***	**94.30**

4.6 Visualizing Learnt Features

This section focuses on visualising the features learnt by *XDecompo* and other training strategies, interpreting what they represent, and making the model's decisions more understandable. To get insights into the decision-making process, we used the GRAD-CAM algorithm as a post hoc explainable AI to highlight the salient features influencing the model's predictions and generate the final heatmap for each class of the downstream datasets. The heatmaps overlay on the input images, highlighting the areas that most influence the model's prediction, making it easier to understand why a certain prediction was made. In the heatmap, the red colour points to the highest relevance that contributes significantly to the model's prediction, while the yellow colour refers to a low level of importance and is less activated than the red colour. The blue area means there is no contribution to the model's prediction. To assist interpretation, black, red, and white arrows are used to point to the red, yellow, and blue regions, respectively. For the CRC dataset, the heatmaps for each class of the test set are illustrated in Fig. 4.14, Fig. 4.15, and Fig. 4.16. As demonstrated, *XDecompo* achieves superior localisation of relevant regions compared to the *4S-DT* model, ResNet-50 pre-trained network, and *DeTraC* model. For example, in Fig. 4.14, *XDecompo* accurately detects all areas of adipose tissue (black arrows), whereas other models miss some regions (white arrows) and focus on misleading areas (red arrows). Likewise, in Fig. 4.15 and 4.16, *XDecompo* more effectively identifies the relevant regions within the images compared to other models.

For the brain tumour dataset, Fig. 4.17, 4.18, and 4.19 display the heatmaps for

glioma, meningioma, and pituitary tumours classes, respectively. As shown, *XDecompo* outperforms other models in accurately detecting the tumour (black arrows), particularly in the meningioma class, while avoiding misleading regions (red arrows).

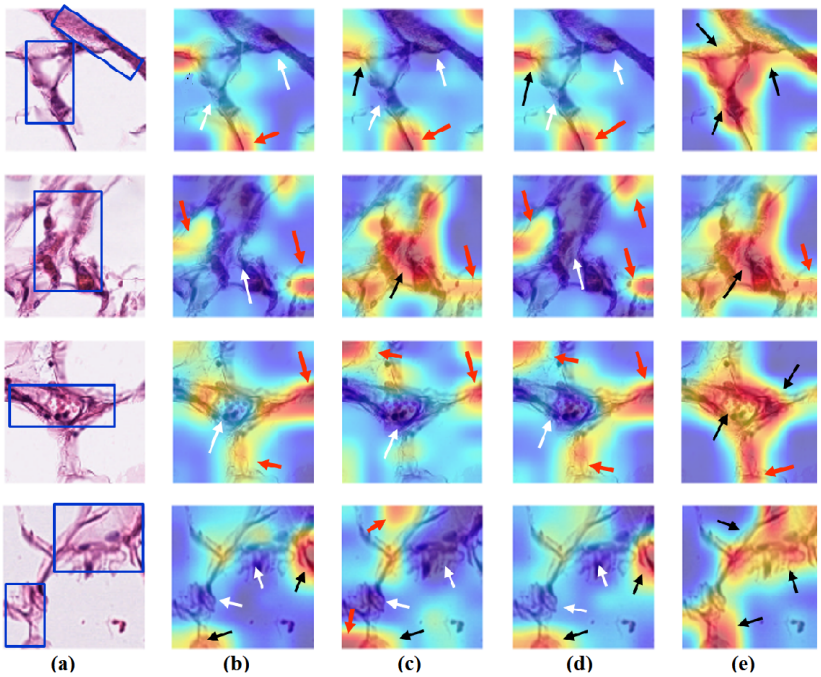

FIGURE 4.14: Visualisation of deep features for class ADI of CRC test set images obtained by each model: a) original image, b) ResNet-50 pre-trained network, c) DeTraC, d) *4S-DT*, and e) *XDecompo*.

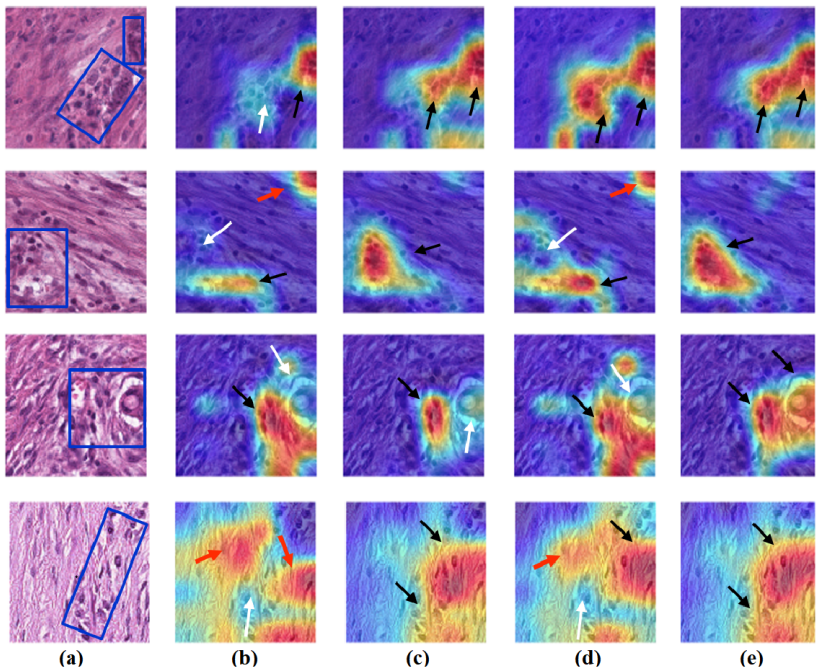

FIGURE 4.15: Visualisation of deep features for class STR of CRC test set images obtained by each model: a) original image, b) ResNet-50 pre-trained network, c) DeTraC, d) *4S-DT*, and e) *XDecompo*.

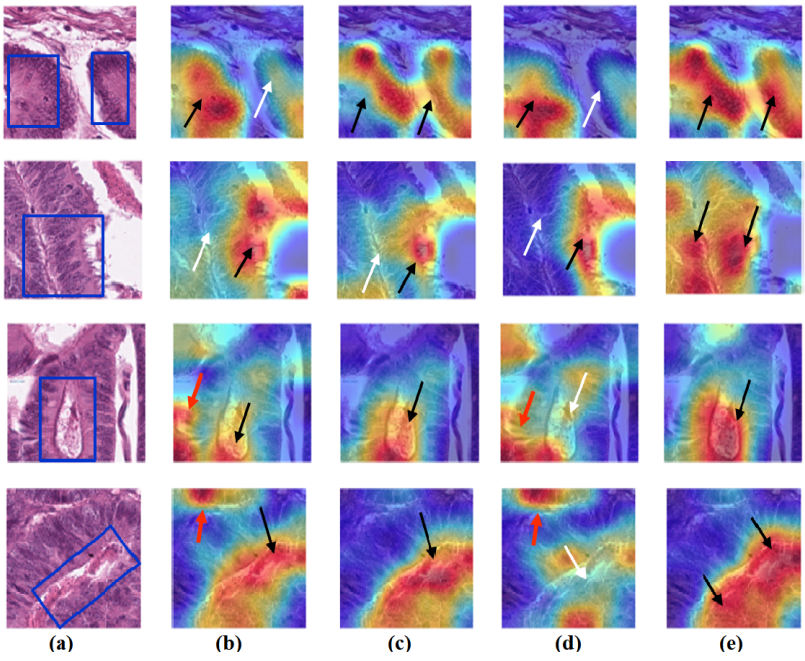

FIGURE 4.16: Visualisation of deep features for class TUM of CRC test set images obtained by each model: a) original image, b) ResNet-50 pre-trained network, c) DeTraC, d) *4S-DT*, and e) *XDecompo*.

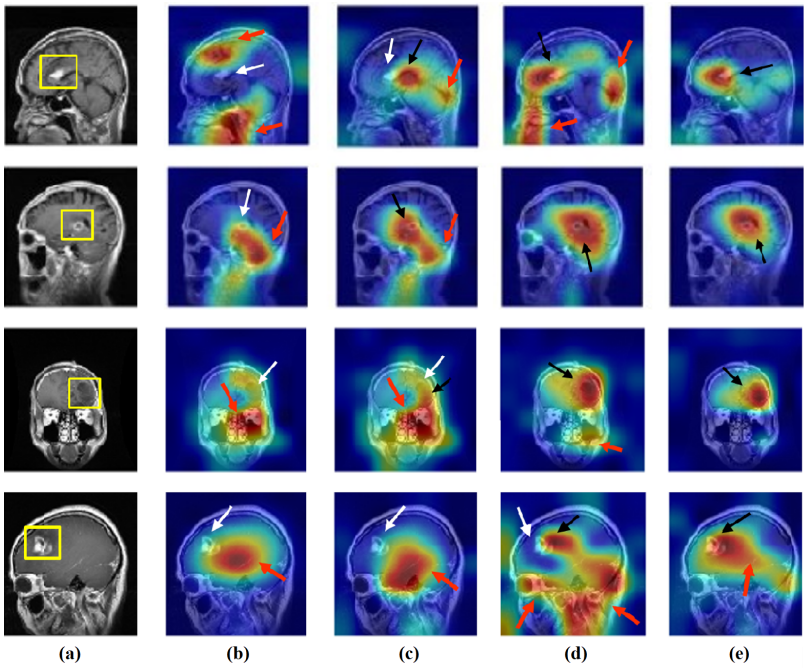

FIGURE 4.17: Visualisation of deep features for class glioma of brain tumour test set images obtained by each model: a) original image, b) ResNet-50 pre-trained network, c) DeTraC, d) *4S-DT*, and e) *XDecompo*.

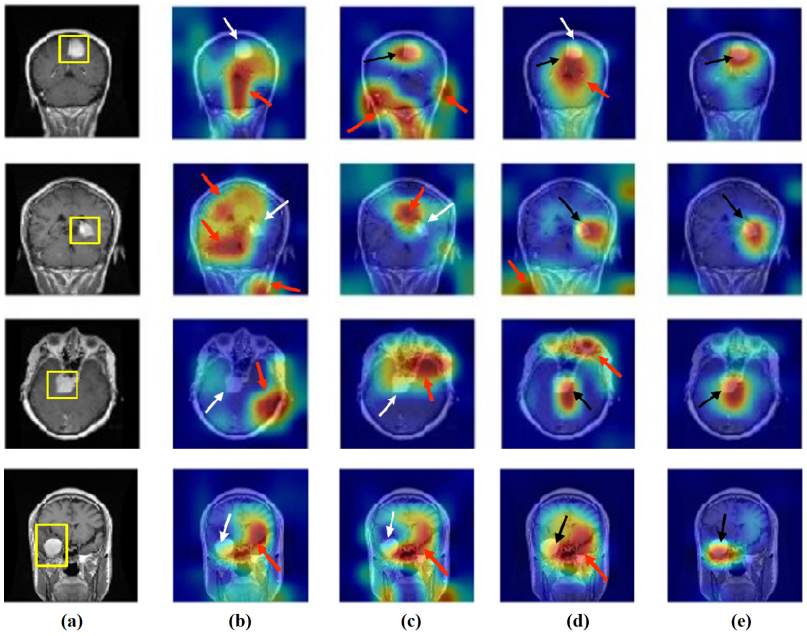

FIGURE 4.18: Visualisation of deep features for class meningioma of brain tumour test set images obtained by each model: a) original image, b) ResNet-50 pre-trained network, c) DeTraC, d) *4S-DT*, and e) *XDecompo*.

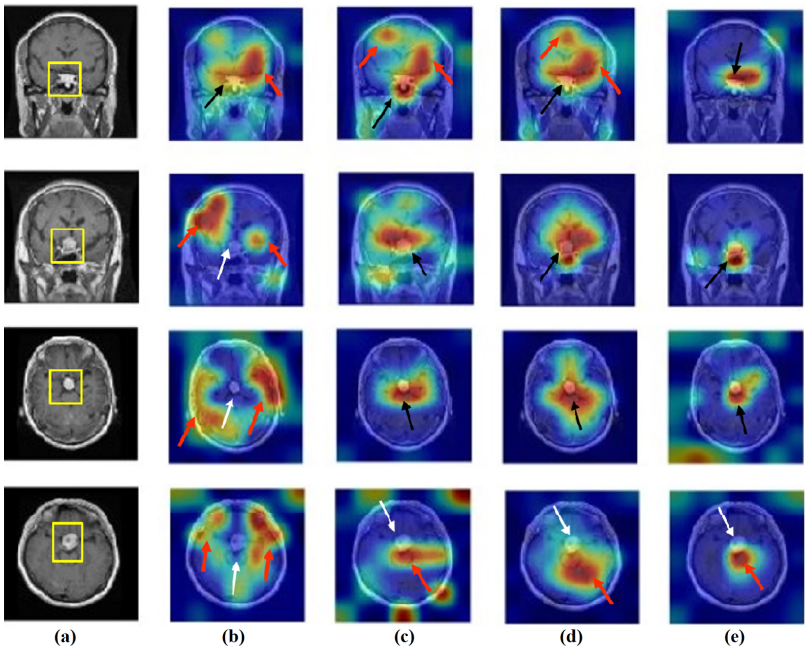

FIGURE 4.19: Visualisation of deep features for class pituitary tumours of brain tumour test set images obtained by each model: a) original image, b) Traditional transfer learning, c) DeTraC, d) *4S-DT*, and e) *XDecompo*.

4.7 Summary

This chapter introduced our first contribution to this thesis: *4S-DT* and its developed version, *XDecompo*, to enhance the feature transferability and improve the classification performance of medical image datasets, particularly those with limited sample sizes. First, we introduced the *4S-DT* model to improve the detection of COVID-19 cases by using the *k*-means clustering algorithm to generate sub-classes for the downstream task. While *k*-means is computationally efficient and well-suited for large datasets, it is highly sensitive to initial centroid placement and outliers, which can lead to inaccurate clustering. To address these limitations, *XDecompo* introduces a non-parametric clustering algorithm that automatically determines the optimal number of sub-classes in the downstream dataset. In *XDecompo*, the decomposition process is guided by the AP clustering algorithm, which automatically identifies the number of clusters without user intervention. This automated approach is more effective for complex and heterogeneous datasets, as it helps better define class boundaries. Consequently, the model generalises more effectively and performs well on test sets. Moreover, *XDecompo* includes an explainable component that highlights vital areas contributing to the model's predictions, helping in understanding and validating its outputs. For evaluation of *XDecompo*, we used two different medical image datasets: the CRC and the brain tumour datasets, which suffer from irregular distribution within the classes. The results obtained from the model were compared with the results of *4S-DT*, *DeTraC* models, and the traditional transfer learning technique. The comparison showed that *XDecompo* outperformed other training strategies and previous works in the field. In terms of explainable AI, *XDecompo* also demonstrated its ability to provide clear visual explanations by highlighting the most critical regions in tumour images, thus enhancing the interpretability and trustworthiness of the classification results. In conclusion, the results indicate that utilising an AP-based approach for the decomposition of downstream datasets significantly enhances the model's ability to capture the inherent structure of the data. This improvement leads to more effective feature learning and superior classification performance. The combination of automated clustering and explainable AI components demonstrates the potential of *XDecompo* as a powerful tool for medical image analysis.

The next chapter will explore curriculum learning as another strategy for improving the classification of medical image datasets, particularly those with irregular class distributions. Curriculum learning, inspired by the way humans learn, involves training models with increasingly complex examples. This approach can help models gradually adapt to difficult samples, enhancing their robustness and performance. By investigating curriculum learning, we aim to address the challenges of irregular distribution, providing an additional method to improve the classification accuracy and generalisation of medical image analysis.

Chapter 5

Curriculum Learning with Class Decomposition for Classification

In the previous chapter, we presented a detailed explanation of the *4S-DT* and *XDecompo* models, which form the first contribution of this thesis. These models are introduced to enhance the classification performance of medical image datasets and address the issue of irregular class distribution, particularly when certain classes have limited sample sizes. Unlike the parametric nature of *4S-DT*, *XDecompo* benefits from a non-parametric approach to enhance its generalisation capabilities and generate more precise clusters. *XDecompo* demonstrated superior feature transferability compared to *4S-DT*, significantly improving classification performance on two distinct datasets: brain tumours and colorectal cancer (CRC).

In this chapter, we introduce a novel model called *CLOG-CD*, designed to improve generalisation and the training process by gradually increasing class complexity in a structured way. This allows the model to learn more relevant features and reduce class overlap by simplifying complex structures into smaller, more homogeneous groups. Findings reported in this chapter are accepted in IEEE Transactions on Emerging Topics in Computing, 2025.

5.1 Overview

In this chapter, we introduce a novel convolutional neural network (CNN) called *CLOG-CD*: Curriculum Learning based on Oscillating Granularity of Class Decomposed Medical Image Classification. *CLOG-CD* combines an anti-curriculum learning strategy with class decomposition to progressively learn and transfer discriminative features across different levels of class granularity, enhancing both feature transferability and classification performance. In addition, it mitigates the issues of irregular distribution within classes. *CLOG-CD* was evaluated on four different medical image datasets using two baseline networks. It achieved an accuracy of 96.08% for the chest x-ray (CXR) dataset, 96.91% for the brain tumour dataset, 79.76% for the digital knee x-ray, and 99.17% for the CRC dataset with ResNet-50. In addition, with DenseNet-121, *CLOG-CD* achieved 94.86%, 94.63%, 76.19%, and 99.45% for CXR, brain tumour, digital knee x-ray, and CRC datasets, respectively.

This chapter is organised as follows: Section 5.2 provides an introduction to the background and motivation behind the *CLOG-CD* model. In Section 5.3, we provide an in-depth explanation of the *CLOG-CD* model. Section 5.4 outlines the experimental setup, including the datasets used, performance metrics, and a comprehensive analysis of the results obtained by applying *CLOG-CD* to medical image datasets. Section 5.5 presents the ablation study. Section 5.6 provides a discussion of the results. Finally, Section 5.7 concludes the chapter with a summary of our findings and suggestions for future research directions, highlighting the model's impact on classification accuracy and training efficiency.

5.2 Introduction

Medical datasets often exhibit irregularities in data distribution and significant overlap between classes, posing substantial challenges to conventional classification methods. State-of-the-art models still struggle to learn precise class boundaries, leading to reduced performance and reliability. Therefore, there is a pressing need for innovative training approaches that can adapt to these complexities and enhance the robustness of classification systems. Curriculum learning (CL) provides advantages over traditional deep learning strategies by structuring the training process in a meaningful sequence, from easy-to-hard (traditional CL) or hard-to-easy tasks (anti-CL). This ordered approach helps the model learn faster and more effectively, leading to better generalisation. CL techniques have been introduced in various areas such as natural language processing, reinforcement learning, and different computer vision tasks. Bengio et al. [21] introduced this educational technique to the machine and deep learning fields as a way to improve model training by mimicking students' learning processes, beginning with concepts and moving on to more complicated ones. Instead of delivering samples in random sequence as in traditional training systems, Bengio et al. suggested that organising the model's presentation of training instances could be more beneficial, starting from simpler tasks or examples and eventually progressing to more difficult ones. This strategy helps the model to generalise more effectively in short-time training, overcome getting stuck in local minima, and improve the performance of different computer vision tasks.

In this chapter, we introduce a novel CNN based on anti-CL combined with the class decomposition approach to structure the learning process in a way that makes training more effective. First, deep local features are extracted from each dataset using the encoder layer of a convolutional autoencoder (CAE). Then, these features are clustered using the k-means algorithm, where each original class is divided into smaller groups, each assigned a new label that corresponds to the original class. Finally, we adapted the anti-CL strategy with the class decomposition method for training the downstream task, where the model begins training at the highest level of granularity (with the maximum number of sub-classes) and gradually transitions to lower levels. Here, the class decomposition method helps the model first learn

specific features by simplifying the dataset's complex structure and defining clear boundaries between classes. This makes the learning process easier for the model to understand relationships between examples and reduces the impact of overlapping class distributions.

The contributions of this chapter are summarised below:

- introducing the class decomposition process as an effective strategy for enhancing CL in classification tasks;
- combining the class decomposition method with CL allows for handling the irregularities and complexities within the dataset by simplifying the local structure within the classes;
- adopting the anti-CL strategy by initiating model training with the maximum number of decomposed classes allows the classification task to be easier, enabling the model to effectively learn the most relevant features through homogeneous sub-classes;
- utilising anti-CL with different oscillations in class decomposition granularity to enable the model to learn more meaningful features across different levels of specificity within sub-classes; and
- conducting extensive experiments on four different medical image datasets using two baseline models, achieving better performance compared to state-of-the-art methods.

5.3 CLOG-CD Model

In this section, we describe our CL based on the Oscillating Granularity of Class Decomposed (*CLOG-CD*) Medical Image Classification model in detail. As demonstrated in Fig. 5.1, *CLOG-CD* consists of three stages: a) First, deep local features were extracted from each dataset using the encoder layer in CAE. The feature representations from the latent space are then clustered using the k-means cluster algorithm with k=5, forming the granularity levels for dataset decomposition. b) The model is trained using different speed functions (i.e. 1, 2, and 4) to control the transition between granularity levels, each speed represents the pacing at which the model moves between different granularity levels. Slower speeds allow for more levels of learning, while faster speeds encourage quicker transitions between levels. c) In addition, we evaluated different training strategies to compare with the performance of the *CLOG-CD* model: the ascending-descending order (ASG) and the descending-ascending order with one single iteration (DEG). *CLOG-CD* was evaluated over many iterations in each direction, starting from descending-ascending order and returning towards the ascending direction. The sequence of granularity-decomposed datasets is queued for training, starting first with the initial

weights from a pre-trained network. After each level achieves convergence, the learnt weights are transferred to the subsequent granularity level. Once the process reaches the final level (whether high or low granularity), the model's performance is then evaluated on the test sets. As discussed in Chapter 4 Label 4.4.3, an error correction equation was calculated to refine the decomposed clusters back to their original form, allowing for producing the final prediction. Fig. 5.1 illustrates the stages of *CLOG-CD* over multiple iterations (I) based on three different oscillation steps of granularity decomposition and other training strategies, including the ascending-descending strategy (ASG) and descending-ascending with one single iteration (DEG).

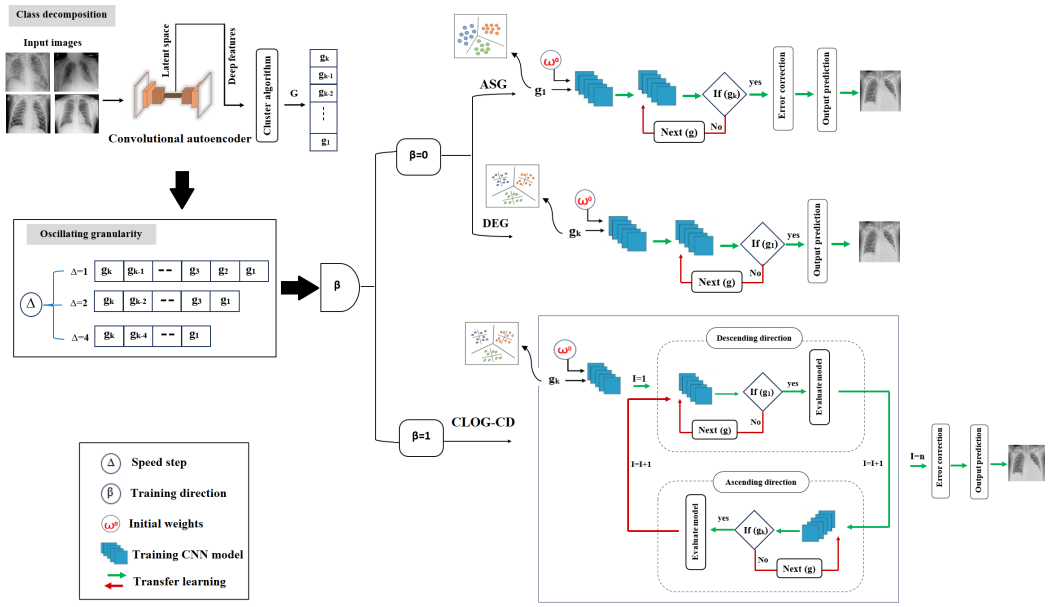

FIGURE 5.1: Architecture of the *CLOG-CD* model. Our model starts with extracting deep local features from the medical image dataset using the encoder layer of CAE. The feature representations from the latent space are then clustered using the k-means algorithm, forming different levels of decomposition granularity. Next, the model is trained using different pacing speeds (1, 2, and 4), which control how the model transitions between these granularity levels. The model is evaluated using simple CL and anti-CL strategies based on a single iteration and using both directions several times. Finally, an error correction equation is applied to refine the decomposed clusters back to their original state, leading to the final prediction.

5.3.1 *CLOG-CD* Feature Extraction

In the CLOG-CD model, feature extraction is a major step that involves obtaining deep local features from the input medical images. We used a CAE, which compresses the high-dimensional input data into a more compact latent space through its encoder layer. The encoder captures critical patterns and salient features from the

images, transforming them into feature vectors that highlight significant information. These vectors form the basis for the granularity of the decomposition process later. The feature vectors are then passed to a k-means clustering algorithm, which organizes the latent space into clusters that represent different granularity levels for dataset decomposition. The mathematical formulation of the encoder's operation is covered in detail in Chapter 4, Section:4.4.1.

5.3.2 Granularity of Class Decomposition

The concept of granularity of decomposition revolves around breaking down each class in a dataset progressively into multiple sub-classes based on the k parameter, where each level of decomposition corresponds to a specific degree of granularity. The purpose of this is to provide the model with different levels of class complexity, allowing the model to learn meaningful features and distinguish features between those sub-classes from diversity decomposition levels during the training process. Where, at the highest level of granularity, the model faces a more difficult classification task with the maximum number of sub-classes. While at the lowest level, the original class structure is retained.

In addition, employing the class decomposition approach simplifies this challenge by breaking down the local structure of these complex classes into smaller and more homogeneous sub-classes, enabling the model to initially learn the most relevant features between data points and making the classification task easier, before gradually integrating this knowledge to fewer sub-classes (low granularity). Therefore, this process allows the model to handle irregularities and complexities within the dataset more effectively.

To construct the granularity of class decomposition for each dataset, let $\mathbf{G}$ denotes the granularity vector from the latent space, which we aim to break down into k new datasets, each representing a specific granularity level, arranged sequentially in decreasing order. Thus, k is expressed as $k = \{k, k-1, k-2, ...\}$. For instance, as depicted in the Fig. 5.2, when $k = 4$, the resulting datasets are $4, 3, 2, 1$, where $k = 1$ reflects the original classes, and $k = 4$ implies that each class has been subdivided into four sub-classes. Consequently, the granularity decomposition with k levels, ordered by descending complexity, is represented as $\mathbf{G} = \{g_k, g_{k-1}, \cdots, g_1\}$. Here, g_1 refers to the original dataset, g_i represents a dataset formed by dividing each class into i sub-classes, and g_k is the dataset derived from the highest level of decomposition with k sub-classes. This process of decomposition can be viewed as a hierarchical structure, where different levels of granularity represent different new datasets.

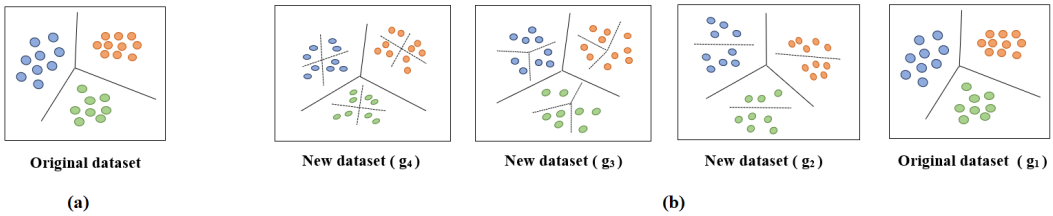

FIGURE 5.2: Illustration of the granularity decomposition concept in the *CLOG-CD* model. It shows how the original dataset is progressively decomposed into more granular clusters: a) the original class, and b) the newly generated datasets after applying the granularity of class decomposition (e.g. k=4). During the training process, the model transitions through these granularity levels, starting at the higher granularity levels and moving gradually towards the lower granularity levels.

5.3.3 Curriculum Learning Strategy and Oscillation

The CL strategy changes the learning process from feeding the entire dataset to the ML model at once into a progressive approach by introducing data gradually, starting with simpler examples and advancing to more complex ones. *CLOG-CD* adopted the anti-CL strategy, where the model learns from the highest granularity level and then moves toward lower ones. This structure promotes better learning by allowing the model to build a foundational understanding with the help of the decomposition mechanism before handling more challenging tasks with fewer classes. In addition, *CLOG-CD* adaptive different oscillation steps to the training process that introduce a dynamic variation in the learning sequence. By incorporating different speed functions between granularity levels, the model has the ability to control how quickly or slowly the fine-tuned techniques move between different granularity levels during training. Where a slower speed allows the model to capture more detailed information across various levels of granularity, whereas a faster speed encourages quicker learning and skipping some intermediate levels.

The CL strategy involves two main critical factors that shape its implementation: 1) The score function refers to the training scheduler or a method to rank training examples based on how difficult they are to learn. It can be determined in various ways: manually through an expert in the domain [130], automatically using predefined tasks by analysing features such as uncertainty or model loss, or dependent measures based on the difficulty of the domain, such as the length of the text or the size of the image [175]. It also includes the order of the training examples based on the difficulty they are to learn, which can follow a traditional ascending order (easy-to-hard), a random order, or a reverse order from a descending-ascending strategy (anti-CL). 2) The pacing function, also called the speed function, defines the plan when and how quickly to introduce more challenging examples to the model during training. In other words, it controls how quickly the transitions between levels occur. It can follow a fixed schedule, predefined before training, or adjust dynamically based on the model's performance.

The objective of the *CLOG-CD* model is to enhance the training of the predictive function $f_\theta : X \rightarrow Y$, where θ represents the parameters being optimised during the learning process. This is done by progressively learning a sequence of models, denoted as $(f_{\theta_1}, ..., f_{\theta_n})$, each gaining knowledge from prior stages. The scoring function $S(x_i, y_i)$, which organises the data starting from the complexity task, can be defined as $S(x_i, y_i) > S(x_j, y_j), \forall\, S : X \rightarrow R$. Where the data point $S(x_i, y_i)$ is considered more difficult than $S(x_j, y_j)$. To facilitate training, the model employs mini-batches (MB) using stochastic gradient descent (SGD). These mini-batches are represented as $MB = \{B_1, B_2, ...B_M\}$, where MB $\subseteq$ X and M is the number of mini-batches. For each mini-batch B_i and a subset $X_i^{'}$, the pacing function $P_\theta(i)$ can be defined as: $P_\theta(i) = |X_i^{'}|$. Where $X_i^{'} = \{X_1^{'}, X_2^{'}, ..., X_M^{'}\}$ represents the samples within the mini-batch B_i, sorted by the complexity given by the scoring function. This approach ensures that the model begins with more complex data and gradually moves to simpler examples, allowing for more adaptive learning across different levels of data difficulty.

In this study, we evaluated the *CLOG-CD* model using three different oscillation step sizes denoted as $\triangle$, where $\triangle = (1, 2, 4)$, to examine how varying pacing strategies influence the model's learning performance at different levels (l) of granularity **G**. Each step size defines the speed at which the model transitions between different decomposition levels, providing valuable insights into how the transfer of prior knowledge affects classification accuracy and the model's capacity to generalise on unseen test data. In more detail, when the speed is set to $\triangle = 1$, the model progresses through each granularity level sequentially, starting training at the highest granularity level g_5, and gradually reducing complexity by transitioning to g_4, g_3, g_2, and finally g_1. While at the speed $\triangle = 2$, the model skips some levels, moving directly from g_5 to g_3, then moves to g_1. This faster pacing could risk losing beneficial features and details, but still allows the model to gain insights from both highly detailed and more general representations of the data. Finally, with $\triangle = 4$, the model rapidly transitions from g_5 to g_1. This fast transition reduces the time and focuses on training from the most detailed and the most general levels without spending time on intermediate granularity.

Moreover, we evaluated the effectiveness of the *CLOG-CD* model using a traditional CL strategy, where training begins at the lowest granularity level g_1 and progresses towards the most challenging level g_k. We donated β as a directional parameter of the training process, $\beta = \{0, 1\}$, where $\beta = 0$ corresponds to the descending direction (anti-CL strategy) and $\beta = 1$ refers to the ascending direction (traditional CL). Algorithm 3 provides a detailed description of the process of *CLOG-CD* based on different oscillating steps of granularity decomposition levels.

Algorithm 3: CLOG-CD *Model*

Input: Unlabelled samples, labelled dataset, $\triangle$: oscillation step, β: training direction, k: cluster component, I: number of iterations.
Output: G: new datasets generated by using the class decomposition method, prediction output.
Granularity of Class Decomposition:
    Use CAE for training unlabelled samples.
    Extract features from the latent representation.
    Use k-means to generate **G** in descending order.
    $\mathbf{G} = \{g_k, g_{k-1}, \cdots, g_1\}$
Training *CLOG-CD* on one direction:
    Training with ($\triangle = 1$, $\beta = 0$).
if *process* = *ASG* **then**
    Arrange **G** in ascending order.
for *i in G* **do**
    if *i = 1* **then**
        $w' \leftarrow$ Initial training (pre-trained network, $\mathbf{G}[i]$).
    W: the best learned weights.
    Training the model (W, $\mathbf{G}[i]$).
Evaluate the performance.
Training *CLOG-CD* based on both directions:
    Training with ($\triangle = [1, 2, 4]$, $\beta = 1$).
    $w' \leftarrow$ initial training (pre-trained network, g_k)
$I = 0$
while $I < n$ **do**
    $I = I + 1$
    for *i in G* **do**
        W: Transfer w' only at I=1.
        G $\leftarrow$ Descending order.
        Training the model (W, **G**[i]).
        W: the best learned weights.
    Evaluate on the test set.
    $I = I + 1$
    for *i in G* **do**
        G $\leftarrow$ Ascending order.
        Training the model (W, **G**[i]).
        W: the best learned weights.
    Evaluate on the test set.
Select the best performance among I_n.

5.4 Experimental Study

In this section, we present a comprehensive experimental study and describe the datasets used to evaluate our model. We conducted experiments with varying oscillation speeds: $\triangle = 1$, $\triangle = 2$, and $\triangle = 4$, to assess how the model's progression through different granularity levels influenced its generalisation ability. These processes were named "*CLOG-CD* $(\triangle = 1)$", "*CLOG-CD* $(\triangle = 2)$", and "*CLOG-CD* $(\triangle = 4)$". The model starts training at the highest granularity level, descending toward the lowest level, and then returning to the highest. This was repeated in both directions for 20 iterations with ResNet-50 and 10 with DenseNet-121. Additionally, this section includes comparisons with other training strategies: (1) transfer learning with fine-tuning, (2) an ascending-descending CL strategy "*CLOG-CD(ASG)*", and (3) an anti-CL strategy "*CLOG-CD(DEG)*". Finally, we evaluated the model's effectiveness before and after applying data augmentation techniques to demonstrate its ability to generalise well from the original dataset features and structures.

5.4.1 Datasets Used

In this study, we used four different medical image datasets to evaluate *CLOG-CD* both before and after applying data augmentation techniques. We used the labelled datasets described in Chapter 1, Section 1.7. Each dataset was randomly divided into three sets: 70% for the training set, 20% for the validation set, and 10% for the test set, which was used for evaluating the performance. For brain tumour dataset, 615 images were reserved as a test set, and 2,449 was increased to 34,286 after applying several augmentation processes. Similarly, for the CRC dataset, we have used the whole 7,180 images of the dataset, which are divided into nine classes, see Fig. 5.3. 723 images were dedicated as a test set, and the rest of the data was augmented to 48,630. The third dataset we used is the CXR dataset, containing 21,165 images divided into classes (3,616 COVID-19, 6,012 Lung-Opacity, 10,192 Normal, and 1,345 Viral Pneumonia), see Fig. 5.4. 2,119 images were dedicated to the test set, and the rest of the images were increased to 57,156 images. Finally, the digital knee dataset includes five classes with 1,650 MRI images. 168 images were reserved for the testing set, and the rest of the images were expanded to 50,748 through different augmentation processes. Fig. 5.5 shows examples of the dataset used.

5.4.2 Hyperparameter Settings

For feature extractions from the downstream dataset, we designed a CAE model with two convolutional layers, and a kernel size was set to 3×3, utilising the ReLU activation function. For the CRC and knee x-ray datasets, the first layer contained 32 filters, while the second had 16. For the CXR and brain tumour datasets, the first and second layers were configured with 16 and 8 filters, respectively. The models were trained using the Adam optimiser with a learning rate of 0.001, across 50 epochs,

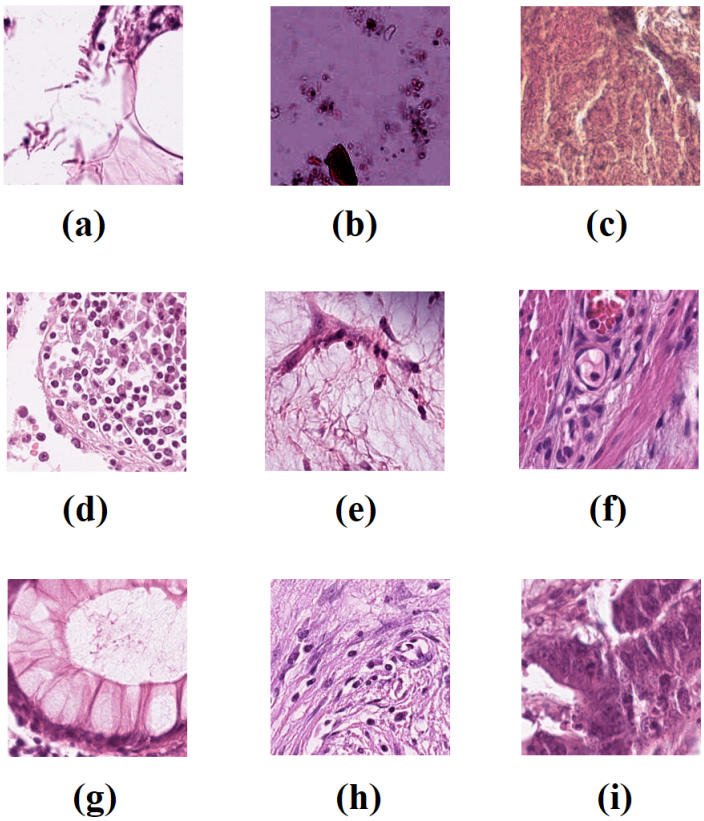

FIGURE 5.3: Example patch images from the CRC-VAL-HE-7K colorectal cancer test set used in our experiment: a) ADI, b) BACK, c) DEB, d) LYM, e) MUC, f) MUS, g) NORM, h) STR, i) TUM.

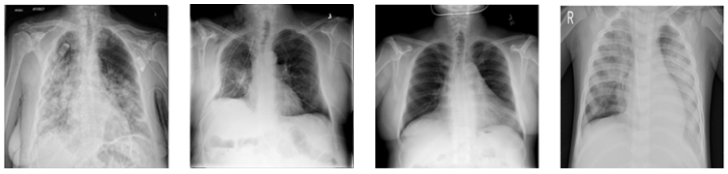

FIGURE 5.4: Example images from CXR test set: a) COVID-19, b) Lung-Opacity, c) Normal, d) Viral Pneumonia.

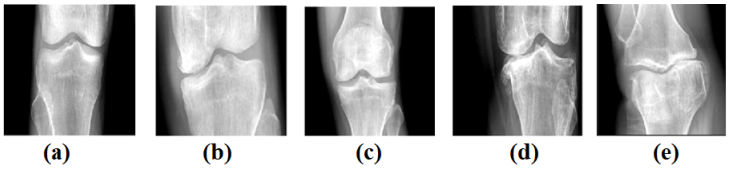

FIGURE 5.5: Example images from the digital knee x-ray images test set: a) Normal, b) Doubtful, c) Mild, d) Moderate, e) Severe.

and a mini-batch size of 50. The feature representations from the latent space were then input into the k-means clustering algorithm to create decomposition granularity clusters with $k = 5$. This process generated four new datasets with sub-classes corresponding to the original class labels, in addition to the original dataset.

For the training *CLOG-CD* model, we used ResNet-50 and DenseNet-121 architectures as the backbones for initial training weights. The choice of these networks is due to their effectiveness in complex image classification tasks. Where ResNet-50 incorporates skip connection layers to prevent vanishing gradients, allowing for deeper networks to be trained effectively. Similarly, DenseNet-121 employs dense connections that enhance gradient flow and reduce the number of parameters, making feature extraction more efficient.

The input layer of these pre-trained networks accepts image size 224×224 pixels, so we decided to resize all images of the datasets to 224×224 pixels to be compatible with the pre-trained networks, and we used a bi-linear interpolation technique for the resizing process, which is commonly used in image processing to maintain image quality, critical features and minimise artifacts.

The models were trained based on the deep-tuning strategy for 50 epochs with a mini-batch size of 50, with a cross-entropy loss function and mini-batch stochastic gradient descent (mSGD) as an optimizer. To prevent overfitting, we used the regularisation technique L2 with a value of 0.001 for CXR, brain tumour, and knee x-ray, and 0.0001 for the CRC dataset. The parameter settings for training each dataset are reported in Table 5.1. The learning rate for the last fully connected layer was set to 0.01, and the output layer was modified to match the number of classes in each new dataset.

TABLE 5.1: *CLOG-CD*: Experimental hyperparameter settings for each dataset.

	ResNet-50		DenseNet-121	
Dataset	Learning rate	Learning rate-decay	Learning rate	Learning rate-decay
CXR	0.001	0.85 every 10 epochs	0.001	0.80 every 15 epochs
Brain tumour	0.0001	0.9 every 10 epochs	0.001	0.80 every 10 epochs
Knee x-ray	0.001	0.90 every 15 epochs	0.0001	0.85 every 10 epochs
CRC	0.0001	0.95 every 15 epochs	0.001	0.90 every 15 epochs

5.4.3 Performance Evaluation

We adopt accuracy, precision, recall, and F1-score, which were defined in Section 4.3.5. In addition, we computed the 95% confidence interval (CI) using the t-test over I iterations for each dataset to provide a robust evaluation of our model's performance, where $I = 20$ for ResNet-50 and $I = 10$ for DenseNet-121 [176]. For each iteration, we calculated the accuracy of the model and then computed the CI around the mean accuracy to account for variability in performance. The t-score is used in hypothesis testing to assess the reliability of sample-based estimates, ensuring a statistically sound evaluation of our model's performance over multiple iterations. The

confidence interval based on the t-test is calculated using the following formula:

$$CI = \bar{x} \pm t.\frac{s}{\sqrt{n}} \tag{5.1}$$

where $\bar{x}$ is the sample mean, t is the critical value from the t distribution, s is the sample standard deviation, and n is the sample size.

5.4.4 Performance of *CLOG-CD*

We evaluated the performance of *CLOG-CD* based on three different oscillation steps using two different ImageNet pre-trained networks (ResNet-50 and DenseNet-121) for classifying the test set of each dataset. Table 5.2 and Table 5.3 summarise the classification performance of *CLOG-CD* using ResNet-50 and DenseNet-121 networks, respectively. It can be noticed that *CLOG-CD* ($\triangle = 1$) has the highest overall classification on the test sets for each dataset. In addition, we can see that, the performance of *CLOG-CD* ($\triangle = 1$) without using augmentation has also achieved the highest performance on all the datasets compared to other values of speed step.

The confidence intervals of *CLOG-CD* based on different oscillation steps are presented in Table 5.4 and Table 5.5. The results indicate that *CLOG-CD* ($\triangle = 1$) provides a more consistent confidence interval compared to other strategies.

In addition, we conducted a statistical significance analysis using the Wilcoxon signed-rank test at 0.05 to evaluate the impact of different oscillation steps in the *CLOG-CD* model. The analysis was conducted on all four datasets using two baseline networks: ResNet-50 and DenseNet-121. With ResNet-50, the p-values on the CXR dataset were 0.0412 ($\triangle = 1$ vs. $\triangle = 2$) and 0.0047 ($\triangle = 1$ vs. $\triangle = 4$). For the brain tumour dataset, the corresponding p-values were 0.0160 and 0.0240; for the digital knee x-ray dataset, 0.00028 and 0.00035; and for the CRC dataset, 0.0220 and 0.0096. Similarly, using DenseNet-121, the p-values on the CXR dataset were 0.0322 ($\triangle = 1$ vs. $\triangle = 2$) and 0.0039 ($\triangle = 1$ vs. $\triangle = 4$). For the brain tumour dataset, the p-values were 0.0019 and 0.00097; for the digital knee x-ray dataset, 0.0322 and 0.0029; and for the CRC dataset, 0.00091 and 0.00097, respectively. These findings confirm that adopting a single speed ($\triangle = 1$) in *CLOG-CD* consistently leads to statistically significant improvements over the other oscillation steps.

TABLE 5.2: classification performance of *CLOG-CD* based on different oscillating steps using the baseline (ResNet-50) for all the datasets.

Dataset	*CLOG-CD* ($\triangle = 1$)				*CLOG-CD* ($\triangle = 2$)				*CLOG-CD* ($\triangle = 4$)			
	ACC (%)	PR (%)	RE (%)	F1 (%)	ACC (%)	PR (%)	RE (%)	F1 (%)	ACC (%)	PR (%)	RE (%)	F1 (%)
CXR	**96.08**	**97.16**	**96.71**	**96.94**	95.66	96.63	96.30	96.46	94.86	95.72	95.16	95.44
brain tumour	**96.91**	**96.86**	**96.32**	**96.59**	96.75	96.36	96.44	96.40	95.12	94.69	94.75	94.73
digital knee x-ray	**79.76**	**81.60**	**78.80**	**80.18**	76.19	77.31	75.65	76.47	72.02	74.51	69.05	71.68
CRC dataset	**99.17**	**99.12**	**98.99**	**99.06**	98.34	98.34	98.06	98.20	98.34	98.11	98.05	98.08
	Without data augmentation techniques											
CXR	**93.58**	**94.55**	**94.38**	**94.47**	88.01	88.60	88.48	88.54	87.54	87.52	87.61	87.57
brain tumour	**90.73**	**89.40**	**89.84**	**89.62**	87.97	86.45	86.11	86.28	84.88	82.89	83.20	83.04
digital knee x-ray	**67.85**	**69.38**	**64.40**	**66.80**	65.47	62.59	57.23	59.79	63.69	61.79	56.32	58.93
CRC dataset	**88.52**	**88.51**	**88.28**	**88.39**	87.28	79.50	82.80	81.16	83.26	80.73	81.88	81.30

TABLE 5.3: classification performance of *CLOG-CD* based on different oscillating steps using the baseline (DenseNet-121) for all the datasets.

Dataset	*CLOG-CD* ($\triangle = 1$)				*CLOG-CD* ($\triangle = 2$)				*CLOG-CD*($\triangle = 4$)			
	ACC (%)	PR (%)	RE (%)	F1 (%)	ACC (%)	PR (%)	RE (%)	F1 (%)	ACC (%)	PR (%)	RE (%)	F1 (%)
CXR	**94.86**	**96.10**	**95.43**	**95.76**	93.44	94.94	91.33	93.10	91.27	94.00	90.09	92.00
brain tumour	**94.63**	**93.91**	**93.99**	**93.95**	92.85	92.45	91.65	92.05	91.87	90.91	90.55	90.73
digital knee x-ray	**76.19**	**75.59**	**76.79**	**76.19**	73.21	73.01	73.10	73.05	72.02	74.81	71.78	73.23
CRC dataset	**99.45**	**99.57**	**99.40**	**99.49**	98.34	98.22	97.76	97.99	97.51	97.66	96.29	96.97
	Without data augmentation techniques											
CXR	**89.29**	**90.47**	**87.44**	**88.93**	84.57	83.98	84.31	84.15	82.30	78.05	77.90	77.98
brain tumour	**91.87**	**90.67**	**90.53**	**90.60**	86.99	85.26	85.40	85.32	85.37	83.47	83.13	83.30
digital knee x-ray	**67.26**	**67.15**	**64.14**	**65.61**	60.11	61.19	58.06	59.59	55.95	57.05	53.09	55.00
CRC dataset	**92.25**	**91.19**	**89.60**	**90.39**	89.76	87.45	84.82	86.11	89.63	86.95	86.55	86.51

TABLE 5.4: Confidence intervals at 95% for *CLOG-CD* based on different oscillating steps with baseline ResNet-50

Dataset	*CLOG-CD* ($\triangle = 1$)	*CLOG-CD* ($\triangle = 2$)	*CLOG-CD*($\triangle = 4$)
CXR	(94.42% and 95.31%)	(94.08% and 94.85%)	(93.92% and 94.41%)
brain tumour	(94.09% and 95.69%)	(91.00% and 94.16%)	(90.19% and 93.08%)
digital knee x-ray	(74.42% and 77.90%)	(69.66% and 73.61%)	(64.78% and 67.99%)
CRC	(84.65% and 95.34%)	(83.74% and 93.22%)	(79.69% and 91.41%)

TABLE 5.5: Confidence intervals at 95% for *CLOG-CD* based on different oscillating steps with denseNet-121.

Dataset	*CLOG-CD* ($\triangle = 1$)	*CLOG-CD* ($\triangle = 2$)	*CLOG-CD*($\triangle = 4$)
CXR	(88.08 and 92.56)	(86.79 and 90.83)	(80.29 and 87.06)
brain tumour	(84.07 and 91.35)	(76.47 and 90.30)	(73.04 and 88.72)
digital knee x-ray	(69.08 and 75.21)	(68.71 and 72.23)	(67.44 and 70.79)
CRC	(96.77% and 99.77%)	(94.17 and 98.01)	(87.08 and 96.49)

5.5 Ablation Study

The ablation studies are performed to assess the influence of each component in *CLOG-CD*. We investigated the performance of *CLOG-CD* with two different strategies: (1) traditional CL strategy, we called this process (ASG), where $\triangle = 1$ and $\beta = 0$. In this process, the model starts training at the lowest granularity level (g_1) with original classes, and the convergence weights are fine-tuned progressively to the next higher level (g_4) until reaching the highest level of granularity (g_5) where the maximum number of sub-classes. After one iteration, the overall classification performance is evaluated on the test set. (2) *CLOG-CD* model based on anti-CL strategy, called DEG, where $\triangle = 1$ and $\beta = 1$. The model is trained in one iteration loop, starting at the highest granularity level (g_5) and gradually the learnt weights are transformed to the next level, until reaching the easiest level (g_1). At the end of this process, the overall classification performance was evaluated on the test set of each dataset. Table 5.6 and Table 5.7 summarise the obtained results from traditional transfer learning and the ASG process. As shown, when augmentation was applied, the ASG model outperformed the traditional transfer learning method on the digital knee X-ray dataset. Specifically, with ResNet-50, the ASG model achieved 69.05% for ACC, 67.61% for PR, 69.19% for RE, and 68.39% for the F1-score. When using DenseNet-121, it also yielded better performance, reaching 61.90% for ACC, 60.51% for PR, 57.44% for RE, and 58.94% for the F1-score. Similarly, without using AUG, the ASG process still demonstrated higher accuracy than the traditional transfer learning method on the knee dataset. However, for the other datasets, the ASG approach performed slightly lower than traditional transfer learning strategies. On the other hand, Table 5.8 shows the performance of the DEG process on the test sets for all the datasets. The DEG model achieved a notable improvement in classification accuracy, particularly with ResNet-50 on the CXR and brain tumour datasets. Furthermore, it outperformed both the ASG model and traditional training methods across all datasets when using DenseNet-121.

The confusion matrices for *CLOG-CD* and other training strategies using ResNet-50 are shown in Fig. 5.6, Fig. 5.7, Fig. 5.8, and Fig. 5.9 for the CXR, brain tumour, digital knee X-ray, and CRC datasets, respectively. Similarly, Fig. 5.10, Fig. 5.11, Fig. 5.12, and Fig. 5.13 present the confusion matrices using DenseNet-121 for the same datasets.

Table 5.9 provides a summary of the results obtained from all the evaluated models using both ResNet-50 and DenseNet-121. The results clearly demonstrate that *CLOG-CD* outperforms other training methods in terms of accuracy. This highlights the effectiveness of combining curriculum learning with class decomposition to improve generalisation and handle irregular data distributions. Moreover, Table 5.10 shows the statistical significance results p-values of the *CLOG-CD* model ($\triangle = 1$) compared to other models on all datasets.

TABLE 5.6: Classification performance of the traditional transfer learning on test sets of all image datasets, using ResNet-50 and DenseNet-121 as baseline networks.

Dataset	Traditional training (ResNet-50)				Traditional training (DenseNet-121)			
	ACC (%)	PR (%)	RE (%)	F1 (%)	ACC (%)	PR (%)	RE (%)	F1 (%)
CXR	91.65	93.82	91.99	92.90	88.72	90.84	87.79	89.29
brain tumour	90.89	89.71	89.83	89.77	86.99	86.86	83.32	85.05
digital knee x-ray	63.69	61.36	62.72	62.03	60.12	58.29	62.14	60.15
CRC dataset	97.28	92.66	91.06	91.85	98.20	97.99	97.87	97.93
	Without data augmentation techniques							
CXR	89.76	90.81	89.67	90.24	84.38	85.89	82.66	84.24
brain tumour	66.99	69.35	59.26	63.91	66.34	61.78	56.55	59.05
digital knee x-ray	35.12	33.84	35.07	34.44	39.29	44.09	33.66	38.18
CRC dataset	80.50	78.02	76.37	77.22	78.56	71.64	71.65	70.21

TABLE 5.7: Classification performance of the (ASG) process on test sets of all image datasets, using ResNet-50 and DenseNet-121 as baseline networks.

Dataset	ASG (ResNet-50)				ASG(DenseNet-121)			
	ACC (%)	PR (%)	RE (%)	F1 (%)	ACC (%)	PR (%)	RE (%)	F1 (%)
CXR	89.52	90.53	89.06	89.79	86.74	88.35	83.81	86.02
brain tumour	77.56	75.62	71.19	73.33	72.03	70.59	68.34	69.45
digital knee x-ray	69.05	67.61	69.19	68.39	61.90	60.51	57.44	58.94
CRC dataset	97.37	94.46	95.75	96.60	96.13	94.97	94.23	94.60
	Without data augmentation techniques							
CXR	88.44	90.13	87.05	88.56	82.16	84.88	73.31	78.67
brain tumour	69.11	63.95	62.68	63.31	63.25	56.19	56.91	56.55
digital knee x-ray	58.93	59.22	58.72	58.97	44.64	39.12	35.51	37.23
CRC dataset	72.47	65.83	65.02	65.42	69.16	63.24	60.84	62.02

TABLE 5.8: Classification performance of the (DEG) process on test sets of all image datasets, using ResNet-50 and DenseNet-121 as baseline networks.

Dataset	DEG (ResNet-50)				DEG (DenseNet-121)			
	ACC (%)	PR (%)	RE (%)	F1 (%)	ACC (%)	PR (%)	RE (%)	F1 (%)
CXR	93.16	94.84	94.18	94.51	91.69	88.28	93.76	90.94
brain tumour	92.20	91.43	91.64	91.54	91.71	90.43	91.31	90.87
digital knee x-ray	64.88	67.62	65.66	66.62	70.24	73.08	69.84	71.42
CRC dataset	97.78	93.57	92.54	93.05	98.89	98.83	98.64	98.74
	Without data augmentation techniques							
CXR	89.75	90.81	89.67	90.24	86.46	87.82	85.77	86.78
brain tumour	83.45	81.46	82.25	81.85	81.63	80.17	77.19	78.65
digital knee x-ray	56.54	56.78	50.26	53.31	61.31	61.11	59.60	60.34
CRC dataset	85.20	84.09	82.36	83.22	87.83	83.90	84.35	84.13

TABLE 5.9: Overall performance comparison of all models across the four datasets using ResNet-50 and DenseNet-121 backbones.

Method	dataset using ResNet-50				dataset using DenseNet-121			
	CXR	brain tumour	digital knee x-ray	CRC	CXR	brain tumour	digital knee x-ray	CRC
Traditional learning	91.65	90.89	63.69	97.28	88.72	86.99	60.12	98.20
ASG model	89.52	77.56	69.05	97.37	86.74	72.03	61.90	96.13
DEG model	93.16	92.20	64.88	97.78	91.69	91.71	70.24	98.89
CLOG-CD($\triangle = 1$)	**96.08**	**96.91**	**79.76**	**99.17**	**94.86**	**94.63**	**76.19**	**99.45**

TABLE 5.10: Statistical significance p-values of *CLOG-CD* ($\triangle = 1$) compared with traditional transfer learning, ASG, and DEG models on all datasets using ResNet-50 and DenseNet-121 backbones.

Dataset	*CLOG-CD* vs			DenseNet-121 Backbone		
Name	Traditional learning	ASG	DEG	vs Traditional learning	vs ASG	vs DEG
CXR	0.00093	0.0091	0.00097	0.00127	0.00121	0.0013
brain tumour	0.0185	0.0121	0.0322	0.0103	0.0294	0.0270
Knee x-ray	0.0097	0.0082	0.0093	0.0091	0.0093	0.0092
CRC	0.0019	0.0019	0.0319	0.0029	00164	0.0024

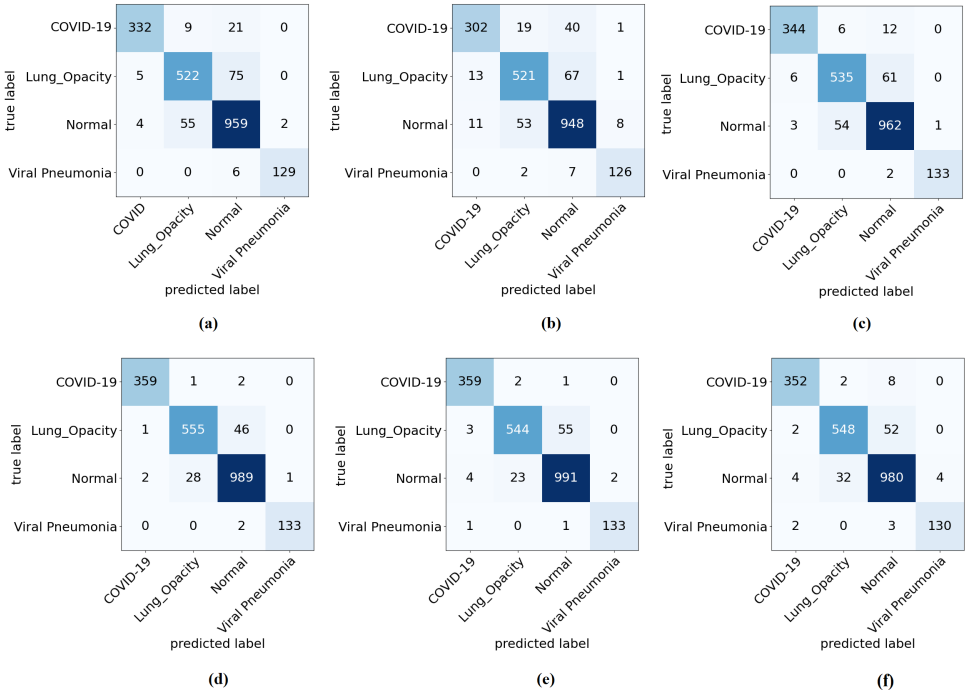

FIGURE 5.6: The confusion matrix results of the CXR dataset obtained by: a) ResNet-50 baseline, b) ASG, C) DEG, d) *CLOG-CD*($\triangle = 1$), e) *CLOG-CD*($\triangle = 2$), and f) *CLOG-CD*($\triangle = 4$).

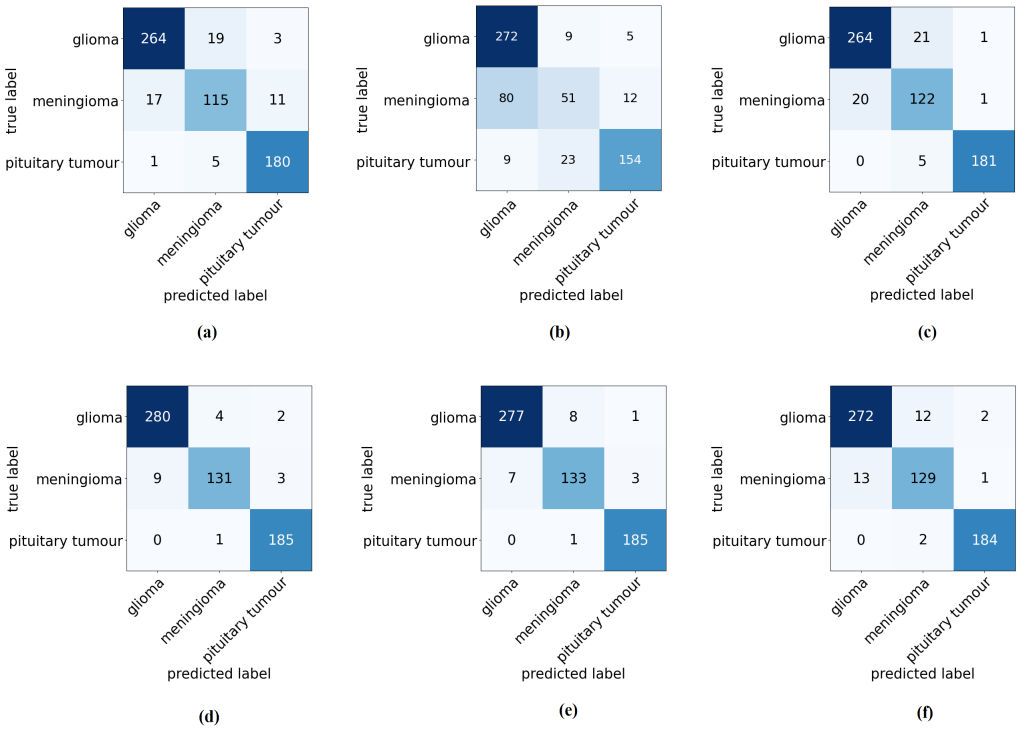

FIGURE 5.7: The confusion matrix results of the brain tumour dataset obtained by: a) ResNet-50 baseline, b) ASG, C) DEG, d) *CLOG-CD*($\triangle = 1$), e) *CLOG-CD*($\triangle = 2$), and f) *CLOG-CD*($\triangle = 4$).

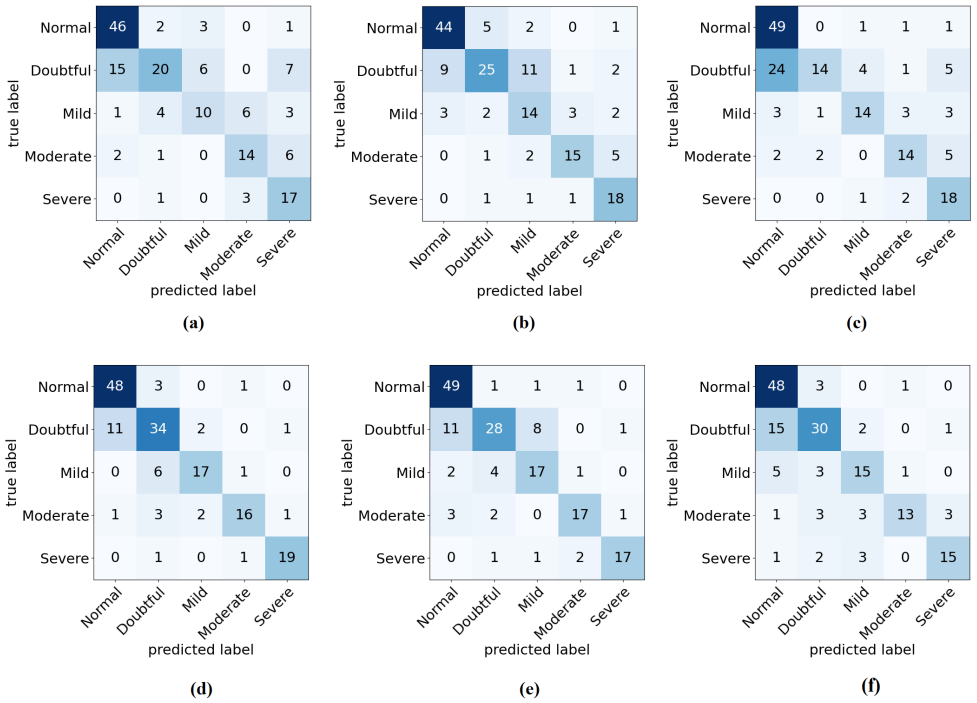

FIGURE 5.8: The confusion matrix results of the digital knee x-ray obtained by: a) ResNet-50 baseline, b) ASG, C) DEG, d) *CLOG-CD*($\triangle = 1$), e) *CLOG-CD*($\triangle = 2$), and f) *CLOG-CD*($\triangle = 4$).

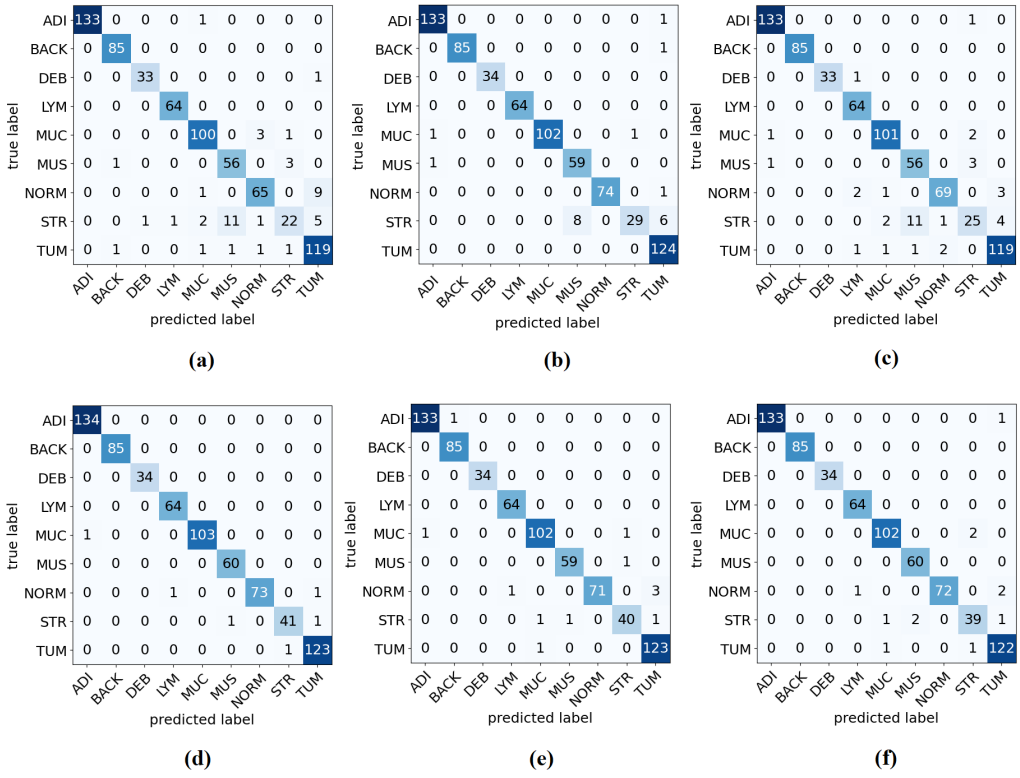

FIGURE 5.9: The confusion matrix results of the CRC dataset obtained by: a) ResNet-50 baseline, b) ASG, C) DEG, d) *CLOG-CD*($\triangle = 1$), e) *CLOG-CD*($\triangle = 2$), and f) *CLOG-CD*($\triangle = 4$).

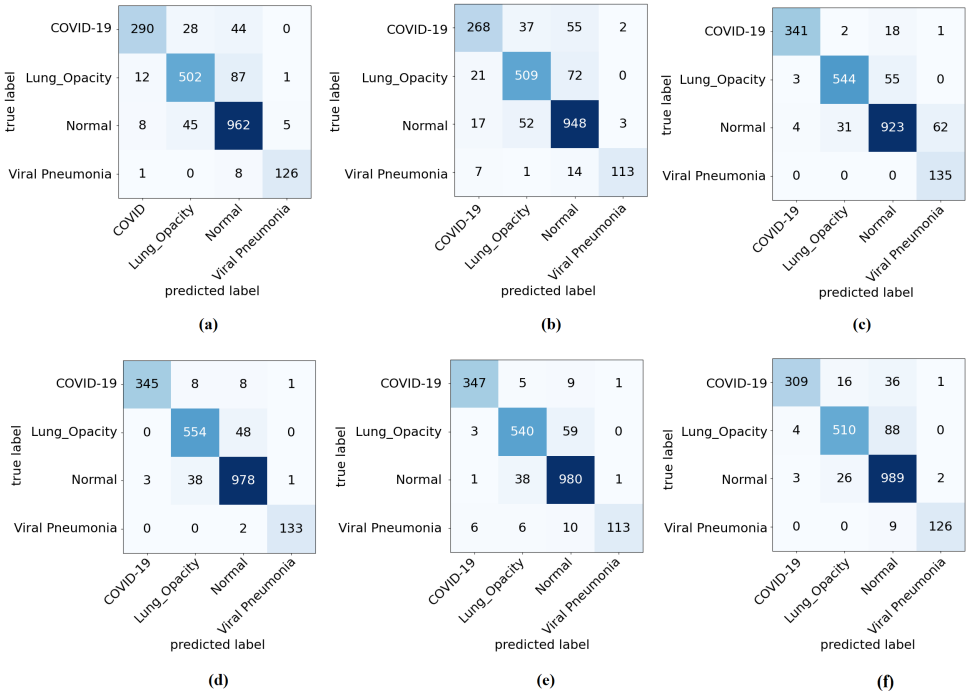

FIGURE 5.10: The confusion matrix results of the CXR dataset obtained by: a) DenseNet-121 baseline, b) ASG, C) DEG, d) *CLOG-CD*($\triangle = 1$), e) *CLOG-CD*($\triangle = 2$), and f) *CLOG-CD*($\triangle = 4$).

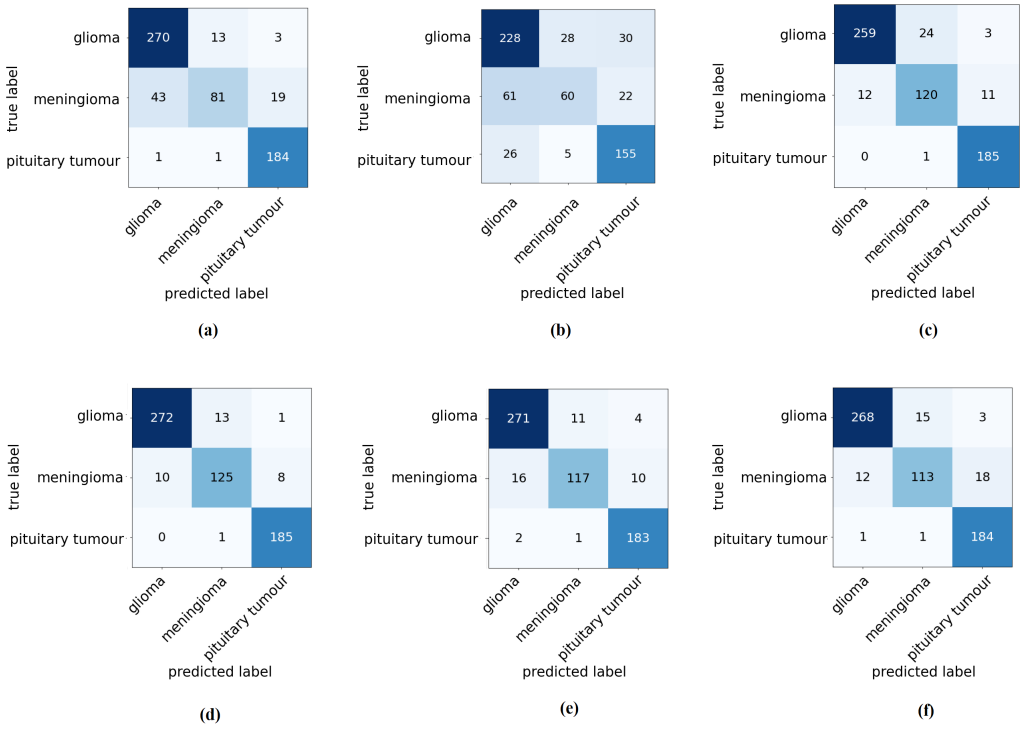

FIGURE 5.11: The confusion matrix results of the brain tumour dataset obtained by: a) DenseNet-121 baseline, b) ASG, C) DEG, d) *CLOG-CD*($\triangle = 1$), e) *CLOG-CD*($\triangle = 2$), and f) *CLOG-CD*($\triangle = 4$).

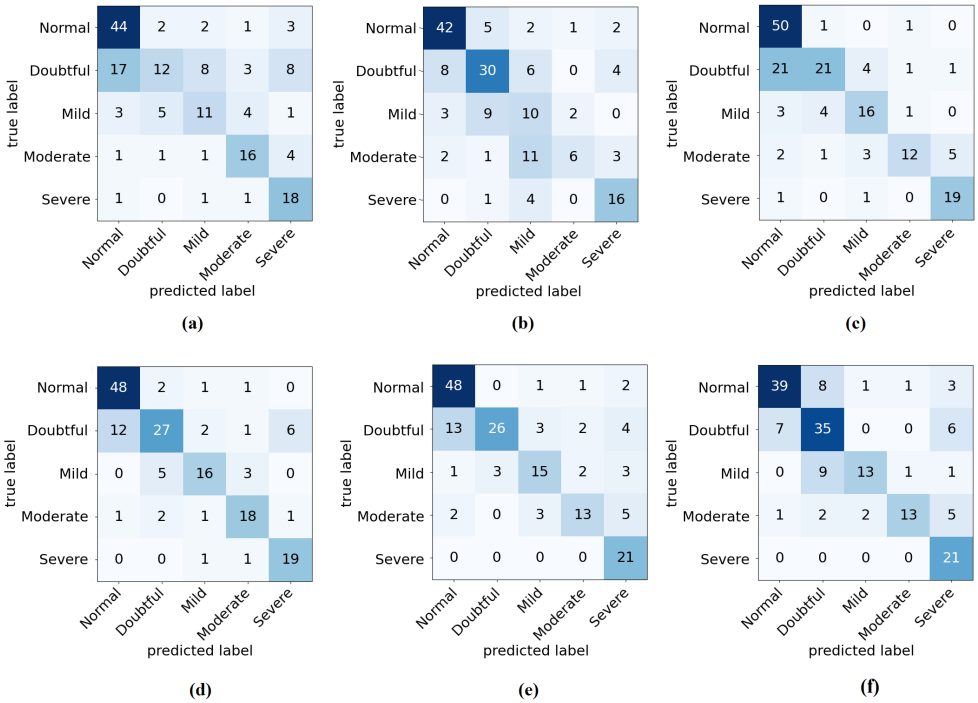

FIGURE 5.12: The confusion matrix results of the digital knee x-ray obtained by: a) DenseNet-121 baseline, b) ASG, C) DEG, d) *CLOG-CD*($\triangle = 1$), e) *CLOG-CD*($\triangle = 2$), and f) *CLOG-CD*($\triangle = 4$).

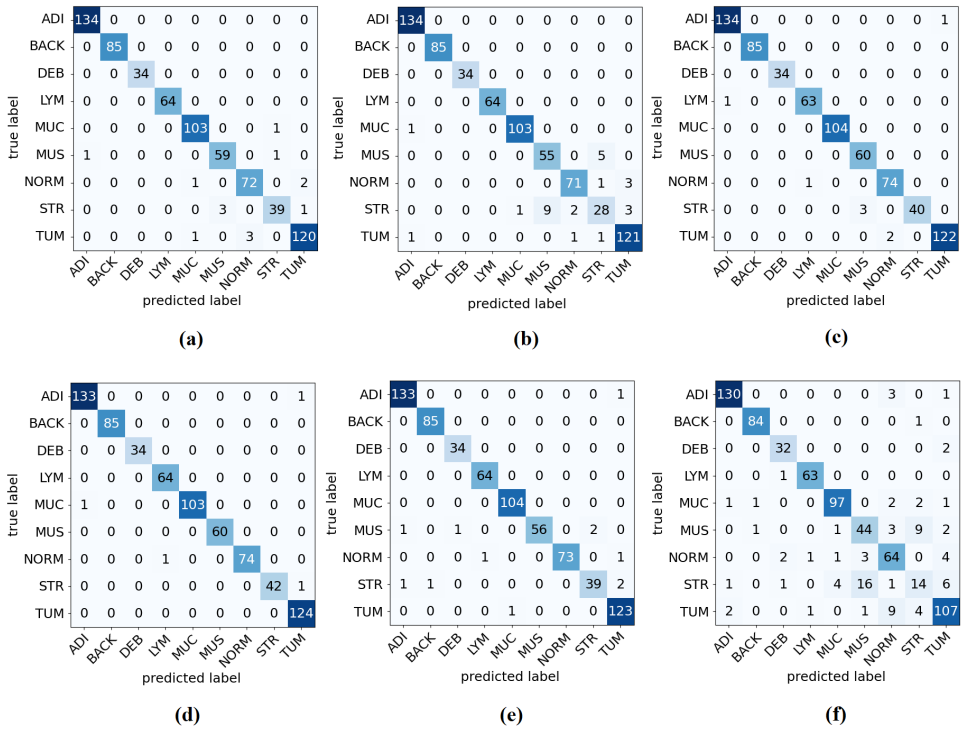

FIGURE 5.13: The confusion matrix results of the CRC dataset obtained by: a) DenseNet-121 baseline, b) ASG, C) DEG, d) *CLOG-CD*($\triangle = 1$), e) *CLOG-CD*($\triangle = 2$), and f) *CLOG-CD*($\triangle = 4$).

5.5.1 Comparison with State-of-the-art Methods

The obtained results in this study were compared with other works that achieved successful results in the field, such as DCLU [177] and curriculum learning with the prior uncertainty method [131]. DCLU introduces a novel CL strategy based on uncertainty estimation to dynamically adapt training to the difficulty of data samples. Jiménez-Sánchez et al. [131] introduce three curriculum strategies: weighting, reordering, and sampling training data, guided by two scoring functions based on domain-specific knowledge and leveraging dynamic uncertainty. The datasets used in the comparison were evaluated without applying the augmentation technique. For the DCLU model, the reported classification accuracies were 88.44%, 90.20%, 49.40%, and 91.84% on the CXR, brain tumour, digital knee x-ray, and CRC datasets, respectively. from Table 5.2, our *CLOG-CD*($\triangle = 1$) model outperformed DCLU on the CXR, brain tumour, and digital knee x-ray datasets. For the CRC dataset, our model with ResNet-50 achieved an accuracy of 88.52%, lower than DCLU's performance. However, when using DenseNet-121, *CLOG-CD* surpassed DCLU with an accuracy of 92.25%, see Table 5.3. In addition, we compared our results with the prior uncertainty method [131]. The model achieved 61.11%, 51.38%, 40.00%, and 65.28% on CXR, brain tumour, digital knee x-ray, and CRC datasets, which are lower than the performance of our *CLOG-CD*($\triangle = 1$) model, see Table 5.2 and Table 5.3.

Moreover, comparisons with prior studies that used the same datasets under different experimental setups highlight the effectiveness of the *CLOG-CD* method. For instance, in CXR classification, our *CLOG-CD* process (with $\triangle = 1$) achieved 96.08% accuracy, outperforming models such as CNN-DenseNet201 (95.11%) [29], CNN-LSTM (94.50%) [178], and *CoroDet* (94.20%) [179]. In brain tumour classification, our model attained a high accuracy of 96.91%, exceeding other techniques like CNN-MobileNetV2 (92.00%) [180], Genetic Algorithm (94.34%) [181], 7-layered CNN (84.19%) [170], and *XDecompo* (94.30%) [140]. On the digital knee x-ray dataset, our model achieved an accuracy of 79.76%, outperforming CNN-ResNet-50 (64.58%) [182], CNN-VGG-19 (69.70%) [183], and CNN-LSTM (75.28%) [184]. Finally, on the CRC dataset, *CLOG-CD*($\triangle = 1$) achieved the highest accuracy of 99.17%, surpassing ICAL [185] with 94.07%, multi-class texture with CL [169] with 94.3%, and the multi-task ResNet-50 model [166] with 95.0%.

5.6 Discussion of Results

This section discusses the outcomes from all the evaluated models, including the effectiveness of the proposed *CLOG-CD* model under different oscillation steps, as well as a comparison with other training strategies. We first evaluated *CLOG-CD* using three different oscillation steps to investigate how varying pacing strategies influence the model's performance across different levels of granularity. Each step size

controls how quickly the model transitions between decomposition levels, providing valuable insights into how the transfer of prior knowledge affects classification accuracy and the model's capacity to generalise on unseen test data.

As shown in Table 5.2 and Table 5.3, the results consistently show that the slower pacing strategy ($\triangle = 1$) outperformed both $\triangle = 2$ and $\triangle = 4$ across all datasets. The results also indicate that $\triangle = 2$ yields better outcomes than $\triangle = 4$. The superior performance of a slower speed step (i.e. 1) comes from the model spending more time to learn relevant features and capture more detailed information at each level of granularity. Where the model starts training at the highest level of granularity and gradually moves to lower levels, refining its understanding step by step. With ($\triangle = 2$), the model skips some levels, moving directly from g_5 to $g3$, and then moves to g_1. This faster pacing could risk losing beneficial features and details, but still allows the model to gain insights from both highly detailed and more general representations of the data. On the other hand, the fastest speed component ($\triangle = 4$) encourages quicker transitions between levels, potentially saving training time but at the cost of skipping important feature refinement stages.

Regarding other training strategies, *CLOG-CD* also outperforms traditional transfer learning, ASG, and DEG models. The ASG model, which starts training at the lowest granularity (original classes) and progresses to higher granularity (maximum number of sub-classes), shows lower performance. This may be due to the model's initial struggle to extract meaningful features from complex data without prior structure. Conversely, the DEG model, which starts from more specific sub-classes and gradually moves to a more complex structure, shows better performance than ASG. This is due to the class decomposition method, which simplifies the complex pattern by dividing each class into more homogeneous sub-classes, allowing the model to first learn specific features before moving to more generic ones. Finally, the traditional training strategy introduces samples in a random order, which might cause noisy or complex samples to be presented early. This can make it difficult for the model to learn complex patterns and, as a result, slow down convergence.

5.7 Summary

In this chapter, we introduced a novel CNN based on the anti-CL strategy and the class decomposition approach, called Curriculum Learning based on Oscillating Granularity of Class Decomposed (*CLOG-CD*) model. This model aims to improve the classification performance of medical image datasets and address irregular class distributions. *CLOG-CD* was designed to simplify the challenges of multi-class classification tasks by applying anti-CL with different levels of granularity. In addition, *CLOG-CD* allows the model to handle irregularities and complexities within the dataset, leading to more robust performance and generalisation.

This approach starts by training the model on the most complex classification task, where each class is broken down into the maximum number of sub-classes.

Once the model achieves stability and convergence at a given granularity level, it gradually transitions to simpler tasks, where the dataset contains fewer sub-classes. Here, the class decomposition method plays a crucial role in simplifying challenging classification tasks by understanding the boundaries between sub-classes, helping the model focus on the most relevant features. Furthermore, the *CLOG-CD* model incorporates an oscillation component that controls the rate of transition between granularity levels, ensuring an adaptive and efficient learning process. The findings proved that using a single-speed transition ($\triangle = 1$) is more robust and capable of improving the classification performance compared to other oscillation steps ($\triangle = 2$ and $\triangle = 4$). where a gradual transition between different levels of granularity allows the model to efficiently learn from each level and generalise better across all levels of granularity, even without introducing augmented data.

CLOG-CD was evaluated on four different medical image datasets using two baseline networks, ResNet-50 and DenseNet-121, demonstrating its superiority over traditional fine-tuning with ImageNet pre-trained networks and other training strategies. *CLOG-CD* has achieved high accuracy with ResNet-50, recording 96.08% for the CXR dataset, 96.91% for the brain tumour dataset, 79.76% for the digital knee x-ray, and 99.17% for the CRC dataset. Similarly, with DenseNet-121, the model achieved 94.86%, 94.63%, 76.19%, and 99.45% for CXR, brain tumour, digital knee x-ray, and CRC datasets, respectively, and outperformed other state-of-the-art models. *CLOG-CD* is considered the first attempt to combine CL strategy with the class decomposition method to enhance feature transferability and increase the generalisability of deep learning models, especially when dealing with complex and irregular image datasets. Consequently, *CLOG-CD* can be integrated with other methods, making it highly efficient.

In the next chapter, we introduce Curriculum Learning and Progressive Self-supervised Training (*CURVETE*) to enhance the feature representations acquired from the pretext task, thereby improving performance on new tasks. *CURVETE* employs a CL strategy during the training of the pretext model using generic unlabelled samples. This approach encourages a more effective training process that facilitates the learning of rich and meaningful representations, leading to faster convergence and improved performance in downstream tasks.

Chapter 6

Curriculum Learning and Progressive Self-supervised Training

In the last chapter, we explained in detail *CLOG-CD* for improving the training process and the model's performance as the second contribution of the thesis. *CLOG-CD* integrates the anti-curriculum learning strategy with the class decomposition method to boost convergence and improve the classification performance of downstream tasks. In addition, it has the ability to handle the overlapping within classes, which is common in the medical image dataset.

In this chapter, we introduce a self-supervised learning (SSL) model that leverages a curriculum learning (CL) strategy to enhance the training of unlabelled samples through sample decomposition. This approach aims to improve the effectiveness of self-supervised learning by extracting meaningful features across a broad range of solutions, thereby enhancing feature transferability to new tasks. Additionally, the model utilises CL guided by class decomposition (CD) in the downstream task to improve classification performance and overcome the impact of irregular class distributions. Findings reported in this chapter are under review in ICONIP 2025.

6.1 Overview

In this chapter, we develop a self-supervised pre-trained model that utilises a CL-based sample decomposition method for training a large set of unlabelled samples. This strategy helps the model identify complex feature representations within the data and enhances the feature transferability of the downstream data. In addition, by adopting different granularities of decomposition, the optimiser has more room to explore a diverse range of potential solutions and recognise meaningful patterns within the data. As a result, the model improves its ability to generalise and increases the classification performance. *CURVETE* has been evaluated on three medical image datasets: brain tumour, digital knee x-ray, and Mini-DDSM, using two different pre-trained networks. It achieved accuracies of 96.60% on the brain tumour

dataset, 75.60% on the digital knee x-ray dataset, and 93.35% on the Mini-DDSM dataset using the baseline ResNet-50. Furthermore, the classification performance with the baseline DenseNet-121 achieved accuracies of 95.77%, 80.36%, and 93.22% on the brain tumour, digital knee x-ray, and Mini-DDSM datasets, respectively, outperforming other training strategies.

The chapter is structured as follows: in Section 6.2, we give an introduction about the background and the developed work. Section 6.3 illustrates, in detail, the framework of our method. Section 6.4 discusses our experimental results and findings. Section 6.5 summarises our work.

6.2 Introduction

In medical image analysis, acquiring well-annotated samples is a major concern due to their limited availability and the high expense of annotation. Although transfer learning offers a promising solution, it frequently faces difficulties when there is a large domain difference between natural scenes in ImageNet and medical images, which leads to limited transferability. For example, in [186], the experimental work demonstrated that using a pre-training network to detect lymph node metastases in pathology images can increase convergence speed but does not enhance the performance of the transferred features. Moreover, Bau et al. [187] proved that the effectiveness of feature representations obtained through transfer learning relies on how well these representations align with the requirements of the downstream task. This is where SSL comes in as a powerful alternative tool. SSL is similar to transfer learning in that both approaches use an auxiliary pretext task to learn representations before applying them to a target task. However, a key distinction is that in SSL, both the pretext data and the downstream tasks come from the same domain. SSL typically follows two stages: (a) learning meaningful data representations by solving a pretext task that generates pseudo-labels from unlabelled data, and (b) fine-tuning these representations on a specific task using a few labelled samples through transfer learning of the learnt weights.

The most important stage in SSL is the pretext task, which serves as the backbone of SSL by enabling the model to learn useful representations from unlabelled data by solving a related task. This task is designed to learn the model to extract meaningful features that can be transferred to downstream tasks where labelled data may be scarce, such as in medical imaging. The quality and effectiveness of the pretext task directly impact the model's ability to generalise and perform well on the target tasks, making it a critical component in the success of SSL approaches.

In this chapter, we introduce "a Curriculum Learning and Progressive Self-supervised Training" (*CURVETE*) to overcome the limitation of samples in medical image datasets and increase the performance of the classification task. *CURVETE* is designed to investigate the power of using the CL strategy with different granularities of decomposition during the training of generic unlabelled samples. In addition,

CURVETE has the ability to handle the challenge of irregular class distribution by combining the CL strategy with the CD approach in the downstream task. The performance of *CURVETE* has been validated using two baseline networks (ResNet-50, DenseNet-121) on three medical image datasets.

The contributions of the chapter can be summarised as below:

- Introducing an SSL model that utilises CL strategy in training the pretext model to enhance the feature transferability and learn more meaningful representations;
- combining anti-CL strategy with the CD method for a better understanding of class boundaries and simplifying complex tasks;
- adopting the granularity of CD can handle irregularities in data distribution, resulting in improved model performance; and
- conducting a comprehensive experimental study using different pre-trained models on publicly available medical imaging datasets.

6.3 *CURVETE* Model

In this section, we explain our developed method *CURVETE* for solving the problem of limited samples, addressing the overlap within classes, and improving the classification performance of the deep learning model. As illustrated in Fig. 6.1, a large number of unlabelled samples are fed into a convolutional autoencoder (CAE) model to extract deep learning representations, followed by a clustering algorithm to create granular sample decompositions (pseudo-labels). In this stage, we trained the pretext model using the anti-CL with the sample decomposition method, where the model starts training at the most difficult level (with maximum sub-classes) and the learnt weights are then progressively transformed towards easier levels (fewer sub-classes). The CL criterion is based on the learnt weights, which are gradually transformed as the model moves from higher to lower levels of granularity, capturing important and meaningful features at each level. Once the easiest level is reached, the process is reversed, and the model returns back to higher granularity levels, ensuring that learnt representations are refined across different complexities. This process is critical for capturing salient features and meaningful information that can later be fine-tuned for a new problem task, as it allows the self-supervised pretext task to discover a wide variety of viable solutions and identify significant patterns in the data. *CURVETE* also incorporates the anti-CL strategy with the CD method to train on smaller subsets of the downstream data, effectively reducing the overlap in class distributions. As explained in Chapter 5, *CLOG-CD* with a single oscillation step between granularity levels has achieved the highest classification performance compared to other strategies. Consequently, *CURVETE* was also designed to utilise anti-CL based on a single oscillation step,

and the operation was repeated in both directions.

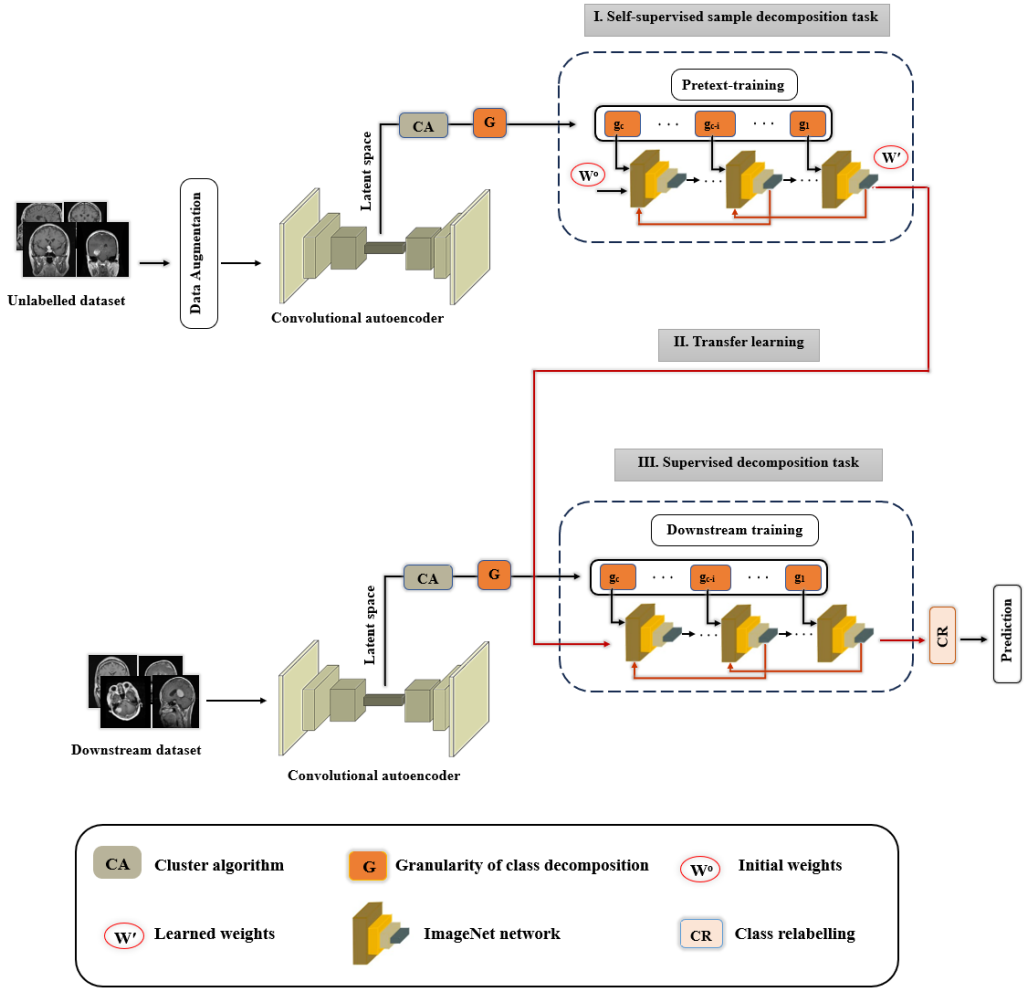

FIGURE 6.1: The workflow of the *CURVETE* model involves three main stages. First, different granular levels of sample decomposition are created using the *k*-means clustering algorithm. Next, the pretext model is trained based on an anti-curriculum learning (anti-CL) strategy, utilising the granularity of the sample decomposition. This training process is repeated multiple times to refine the model's ability to learn rich features and meaningful representations, which are then fine-tuned for a new problem. Finally, the extracted features serve as initial weights for training on a small downstream dataset, where the anti-CL strategy with CD is applied again to further improve classification performance.

6.3.1 Self-Supervised Pretext Task Learning

CURVETE starts first by extracting local feature representations from a large set of unlabelled samples using a CAE model. These features are then fed to a cluster algorithm to generate different granularities of sample decomposition (pseudo-labelled). In *CURVETE*, we used the same scenario in Chapter 5, Section 5.2, where *k*-means

cluster algorithm was used to apply the sample decomposition process and generate sequential levels of granularity for each dataset. Then, we used a pre-trained network as initial weights for training the pretext model. Here, we adopted anti-CL with sample decomposition to solve the pretext task. Based on the outcomes from Table 5.2 and Table 5.3 in Chapter 5, *CLOG-CD* with $\triangle = 1$ and $\beta = 1$ has achieved the highest classification performance for all the datasets. Therefore, we employed the same process for training the pretext model, where the model starts training at the highest granularity level and gradually the acquired knowledge is transferred to the next lower granularity level until reaching the lowest level, then returns again to the highest level.

6.3.2 Supervised Downstream Task Learning

After the pretext model has gained meaningful representations from the unlabelled dataset, it transfers to a smaller, labelled dataset (downstream task) to leverage the learnt features for enhanced generalisation and accurate predictions. By observing the model's performance on the downstream task, we can assess the quality of the feature representations produced by the self-supervised pretext task. To generate different granularities of CD, we also followed the same scenario to extract the feature representations from the latent space of the CAE by using k-means clustering. In addition, the model was trained based on anti-CL with the CD method ($\triangle = 1$ and $\beta = 1$). Finally, class relabelling is performed to correct the classification predictions made during the CD process, and ensure that the final output corresponds to the initial classification problem.

6.4 Experimental Study

In this section, we describe the datasets used to evaluate the effectiveness and robustness of our model. We then detail the experimental procedures conducted on each dataset, including model parameter configurations and evaluation metrics. Furthermore, we provide a comparative analysis of our model's performance against other training strategies.

6.4.1 Datasets Used

In this study, we used three different medical image datasets to evaluate *CURVETE*: brain tumours, the digital knee x-ray, and digital mammogram datasets. *CURVETE* leverages two types of data: unlabelled data for training a pretext model and extracting rich information, and labelled data for training and evaluating the downstream task. The describtion of each dataset was discussed in Chapter 1 Section 1.7.

For the brain tumour dataset, we used the same labelled and unlabelled datasets which were applied in Chapter 4 Section 4.5.1. Regarding the knee dataset, the Osteoarthritis dataset was selected as unlabelled samples [33]. The dataset contains a

total of 9786 images categorised into five grades. We generated more samples by applying several AUG techniques, such as reflection, shifting, sharpening, and rotation to produce 68,502 samples. For the labelled dataset, we used the same dataset in Chapter 5 Section 5.4.1.

Finally, the digital mammograms dataset: MIAS mammograms dataset was used as unlabelled samples [35]. Different AUG processes were applied to get 47,334 samples of the mammogram dataset, such as cropping, zooming, reflection, shifting, and rotation. For the labelled dataset, we used the Mini-DDSM dataset, which is a subset of the larger Digital Database for Screening Mammography (DDSM) [34]. The dataset is divided into three classes: 2048 Normal, 2,716 Cancer, and 2,684 Benign, and all images come in JPEG format with dimensions between 125 and 320 pixels. Fig. 6.2 shows an example of the images from the labelled Mini-DDSM dataset.

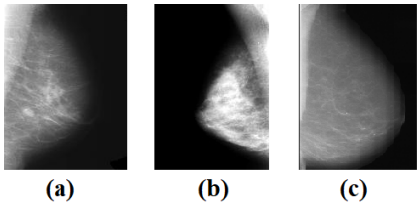

FIGURE 6.2: Example images from the Mini-DDSM dataset: a) Benign, b) Cancer, and c) Normal.

6.4.2 Hyperparameter Settings

We first built a CAE model with two convolutional layers to extract the feature representation from the encoder. For the brain tumour and Mini-DDSM datasets, the number of filters in the first layer was set to 16, and the second one was 8. For the digital knee x-ray dataset, the number of filters in the first and second layers was set to 32 and 16, respectively. Each model was trained for 50 epochs using a kernel size of 3×3, a mini-batch size of 50 and a learning rate of 0.001. The extracted features from the latent representation are then fed into the k-means cluster algorithm with two different components (5 and 10) to generate sample decomposition as pseudo-labelled. Two baseline networks, ResNet-50 and DenseNet-121, were used as initial weights to train the pretext model. The model was trained using anti-CL learning with sample decomposition ($\triangle = 1$ and $\beta = 1$), starting at the highest granularity level, then the gained knowledge is passed down to the next level and continues down until it reaches the original classes, before going back to the most detailed level again.

In our experiments, we took into consideration both the available GPU memory and the limitations of computational resources, so the training process was repeated in both directions over (10) times based on deep-tuning mode with 50 epochs and 50 for mini-batch size with 0.001 for the learning rate, and weight decay was 0.9 every 10 epochs. The last stage is to transfer the acquired information from the self-supervised pretext task into the downstream task. The same scenario was used to

generate different granularities of decomposition for each dataset with component (5). For training the supervised downstream task, we also used the anti-CL with CD method ($\triangle = 1$ and $\beta = 1$) starting training from the maximum level (g_5), until a significant performance boost is achieved, then the converged learnt weights are transformed in a sequential manner until we reach the original classes (g_1). This process was repeated in both directions over (20) times for the ResNet-50 baseline and (10) times for the DenseNet-121 baseline. Based on trial and error experiments, the learning rate for training the brain tumour dataset was set to 0.001, with a weight decay of 0.85 applied every 15 epochs. For the digital knee x-ray images, the learning rate was set to 0.01, with a weight decay of 0.90 applied every 15 epochs. Finally, for the Mini-DDSM dataset, the model was trained with a learning rate of 0.001 and a weight decay of 0.90 applied every 15 epochs.

6.4.3 Performance Evaluation

We adopt accuracy, precision, recall and F1-score, which are defined before in Section 4.3.5.

6.4.4 Performance of *CURVETE* Model

To evaluate the impact of CL with sample decomposition in training the unlabelled dataset and the effectiveness of *CURVETE*, we conducted experiments on the test sets both with and without applying CL with granularity of decomposition in the downstream task.

Table 6.1 presents the performance of the *CURVETE* model, evaluated with two decomposition components, 5 and 10. The model was evaluated with two pre-trained networks, ResNet-50 and DenseNet-121. As shown in Table 6.1, values in bold indicate the highest performance scores achieved. For brain tumour classification, the best accuracy is 96.60% using ResNet-50 and 95.77% using DenseNet-121. The digital knee x-ray dataset achieved an accuracy of 76.60% and 80.36% using ResNet-50 and DenseNet-121. For Mini-DDSM, the best classification accuracy is 93.35% and 93.22% using ResNet-50 and DenseNet-121 networks.

To ensure the effectiveness of *CURVETE*, we evaluated its performance on test sets without applying the CL with the granularity of decomposition in the training of the downstream task. The results are reported in Table 6.2. As shown, the classification performance without using CL and CD in training the downstream task was consistently lower compared to the results in Table 6.1. For example, the brain tumour dataset achieved an overall accuracy of 94.31% without the CL and CD method, compared to 96.60% in the *CURVETE* model. Similarly, the digital knee x-ray and Mini-DDSM datasets achieved an accuracy of 67.86% and 68.93%, respectively, which are lower than the accuracy in Table 6.1.

The results demonstrate that incorporating anti-CL with sample decomposition in the self-supervised pretext task enables *CURVETE* to effectively learn and capture feature representations and complex patterns within unlabelled data. By starting training at the highest granularity level and gradually progressing to lower levels, the model can initially simplify complex patterns and better understand relationships between data points. This approach results in robust, highly transferable representations that enhance the downstream task performance, even with limited labelled examples.

TABLE 6.1: Classification performance of *CURVETE* using two baseline networks, ResNet-50 and DenseNet-121, for all the datasets.

Dataset	ResNet-50								DenseNet-121							
	CURVETE (G=5)				*CURVETE* (G=10)				*CURVETE* (G=5)				*CURVETE* (G=10)			
	ACC (%)	PR (%)	RE (%)	F1 (%)	ACC (%)	PR (%)	RE (%)	F1 (%)	ACC (%)	PR (%)	RE (%)	F1 (%)	ACC (%)	PR (%)	RE (%)	F1 (%)
Brain tumour	95.12	94.11	95.04	94.57	**96.60**	**95.82**	**96.56**	**96.19**	**95.77**	**95.18**	**95.15**	**95.16**	93.01	91.79	92.48	92.13
digital knee x-ray	**75.60**	**76.54**	**73.54**	**75.01**	73.21	75.06	73.07	74.05	**80.36**	**83.24**	**78.64**	**80.87**	72.62	71.04	68.14	69.56
Mini-DDSM	**93.35**	**93.35**	**93.55**	**93.45**	91.94	92.04	92.12	92.08	92.58	92.63	92.79	92.71	**93.22**	**93.29**	**93.40**	**93.35**

TABLE 6.2: classification performance of *CURVETE* model without using curriculum learning with class decomposition method on the downstream task.

Dataset	ResNet-50								DenseNet-121							
	CURVETE (G=5)				*CURVETE* (G=10)				*CURVETE* (G=5)				*CURVETE* (G=10)			
	ACC (%)	PR (%)	RE (%)	F1 (%)	ACC (%)	PR (%)	RE (%)	F1 (%)	ACC (%)	PR (%)	RE (%)	F1 (%)	ACC (%)	PR (%)	RE (%)	F1 (%)
brain tumour	93.66	92.92	92.53	92.72	94.31	93.26	93.63	93.45	87.97	86.38	86.20	86.29	88.13	86.57	86.29	86.43
digital knee x-ray	66.07	67.86	63.91	65.83	67.86	66.36	61.72	63.95	55.95	65.84	59.80	62.67	57.74	60.80	59.20	59.98
Mini-DDSM	66.24	67.02	66.38	66.83	68.93	69.51	69.35	69.43	71.99	75.82	71.87	73.80	66.50	80.25	66.85	72.94

6.4.5 Ablation Study

To assess the effectiveness of the proposed *CURVETE* model, we conducted a comprehensive comparison with three different training strategies: a) traditional transfer learning, using two pre-trained baselines, ResNet-50 and DenseNet-121; b) the *CLOG-CD* model, introduced in Chapter 5, which applies an anti-CL strategy combined with CD in the downstream task; and c) *CURVETE*(WO/CL, W/CD), which uses SSL with sample decomposition for pretext training but does not apply CL in the pretext learning phase. Tables 6.3 and 6.4 report the classification performance of these strategies across three datasets, using both ResNet-50 and DenseNet-121 as

backbone networks. The results show that traditional transfer learning consistently yields the lowest performance across all datasets and backbones. This highlights its limited generalisation capability when fine-tuned with scarce labelled data, particularly in medical imaging tasks with complex and irregular distributions.

On the other hand, *CLOG-CD* outperforms traditional transfer learning by leveraging anti-CL and class decomposition at different levels of granularity. This progressive structure improves generalisation and the training process by gradually increasing class complexity in a structured way. Finally, *CURVETE*(WO/CL, W/CD), shows notable improvement, especially in the digital knee x-ray dataset. This confirms that utilising SSL with sample decomposition for training unlabelled data encourages the transformation of coarse features from general samples to specific tasks by simplifying the complex patterns and local structure of the dataset, providing more effective knowledge before fine-tuning for the subsequent task.

In addition, Table 6.5 shows the statistical significance results (p-values) of *CURVETE* against traditional transfer learning, *CLOG-CD*($\triangle = 1$), and *CURVETE*(WO/CL, W/CD) using both ResNet-50 and DenseNet-121 backbones.

By comparing the results presented in Tables 6.3 and 6.4, along with the outcomes in Table 6.1, it is clear that *CURVETE* consistently achieved the highest accuracy on all datasets. This demonstrates that *CURVETE* offers a promising solution for enhancing feature transferability from the pretext task to a new classification task. This is achieved by integrating the CL strategy with SSL and sample decomposition during the training of unlabelled data. This process enables the model to extract more informative and meaningful representations across different levels of granularity.

The confusion matrices for *CURVETE* and other training strategies with ResNet-50 are presented in Fig. 6.3, Fig. 6.5, and Fig. 6.7 for the brain tumour, digital knee x-ray, and Mini-DDSM datasets, respectively. In addition, Fig. 6.4, Fig. 6.6, and Fig. 6.8 show the confusion matrices using DenseNet-121 for the same set of datasets.

TABLE 6.3: Classification performance of other training strategies using the baseline ResNet-50 for all the datasets.

Dataset	Transfer learning (ResNet-50)				*CLOG-CD* ($\triangle = 1$)				*CURVETE* (WO/CL, W/CD)			
	ACC (%)	PR (%)	RE (%)	F1 (%)	ACC (%)	PR (%)	RE (%)	F1 (%)	ACC (%)	PR (%)	RE (%)	F1 (%)
brain tumour	91.22	89.85	90.72	90.28	93.98	93.54	92.94	93.24	93.66	93.63	91.89	92.76
digital knee x-ray	61.31	60.66	59.57	60.11	70.83	72.71	68.47	70.53	71.43	72.55	70.98	71.76
Mini-DDSM	66.88	67.42	67.46	67.44	91.05	91.09	91.30	91.20	91.94	91.96	92.16	92.06

TABLE 6.4: Classification performance of other training strategies using the baseline DenseNet-121 for all the datasets.

Dataset	Transfer learning (ResNet-50)				*CLOG-CD*($\triangle = 1$)				*CURVETE*(WO/CL, W/CD)			
	ACC (%)	PR (%)	RE (%)	F1 (%)	ACC (%)	PR (%)	RE (%)	F1 (%)	ACC (%)	PR (%)	RE (%)	F1 (%)
brain tumour	89.59	88.35	88.15	88.25	91.87	90.45	92.12	91.28	93.33	92.23	92.33	92.28
digital knee x-ray	69.05	72.63	68.00	70.24	67.26	67.15	64.14	65.61	73.21	71.13	70.43	70.78
Mini-DDSM	66.75	67.41	67.19	67.30	84.65	84.90	84.86	84.88	86.32	86.60	86.57	86.58

TABLE 6.5: Statistical significance (p-values) of *CURVETE* compared with traditional transfer learning, *CLOG-CD*($\triangle = 1$), and *CURVETE*(WO/CL, W/CD) models on all datasets using ResNet-50 and DenseNet-121 backbones.

Dataset Name	*CURVETE*(ResNet-50) vs			*CURVETE*(DenseNet-121) vs		
	Traditional learning	*CLOG-CD*	*CURVETE*(WO/CL, W/CD)	Traditional learning	*CLOG-CD*	*CURVETE*(WO/CL, W/CD)
brain tumour	0.038	0.001	0.0039	0.0217	0.0014	0.0037
Knee X-ray	0.0025	0.00021	0.0091	0.0079	0.0032	0.0081
Mini-DDSM	0.0010	0.0053	0.0010	0.0029	00173	0.0035

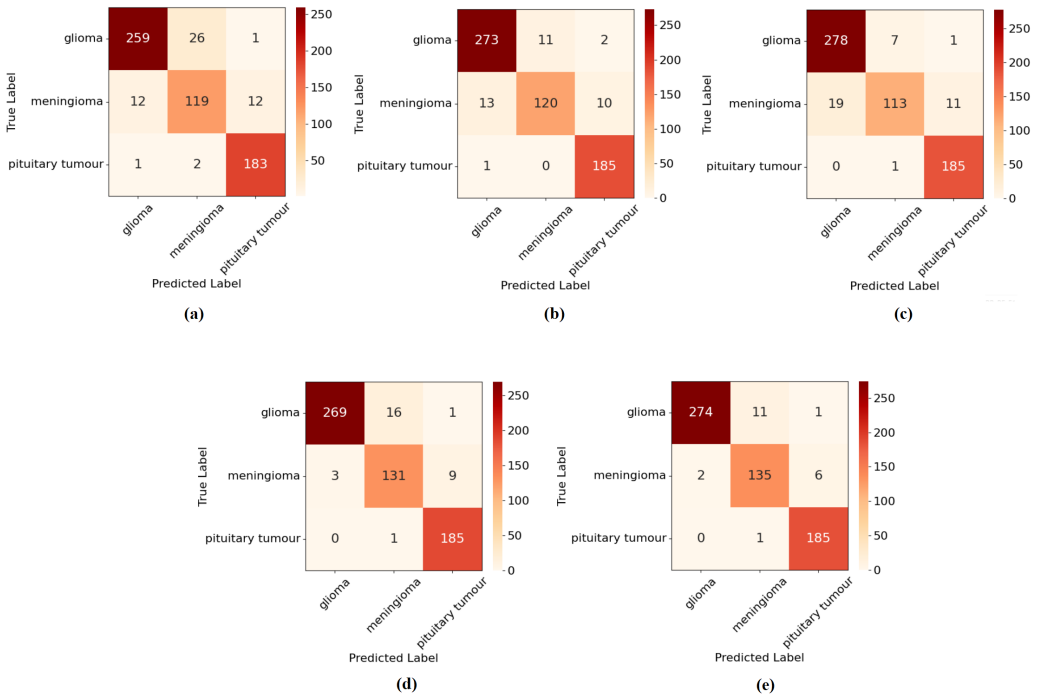

FIGURE 6.3: The confusion matrix results of the brain tumour dataset obtained by: a) Transfer learning with ResNet-50, b) *CURVETE*(WO/CLCD), c) *CURVETE*(WO/CL, W/CD), and d) *CURVETE*(W/CLCD) **G**=5, and e) *CURVETE*(W/CLCD) **G**=10.

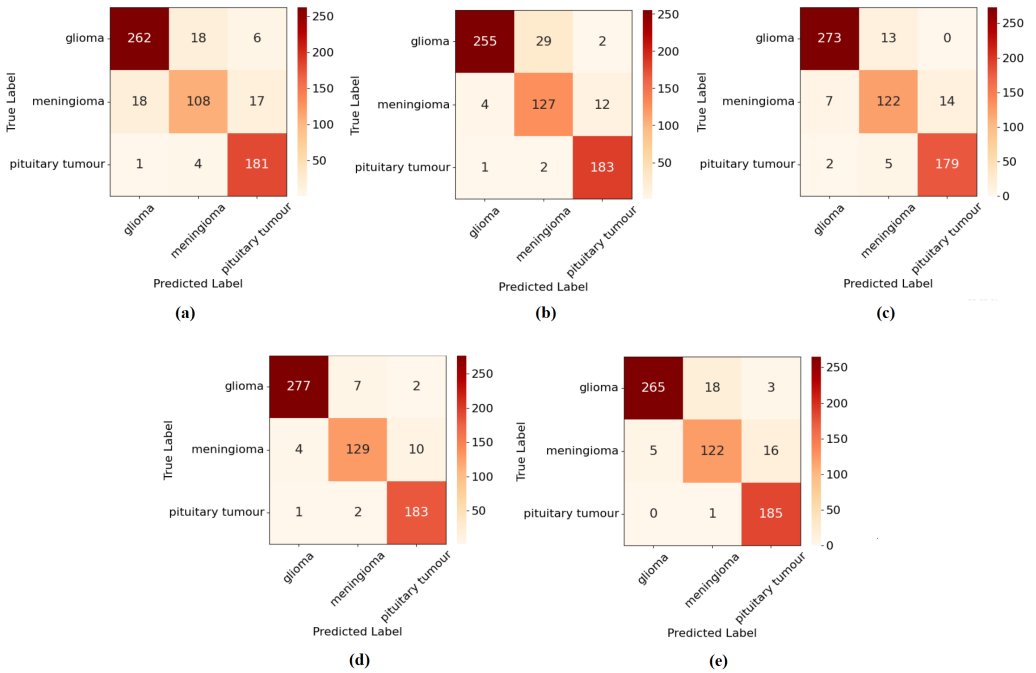

FIGURE 6.4: The confusion matrix results of the brain tumour dataset obtained by: a) transfer learning with DenseNet-121, b) *CURVETE*(WO/CLCD), c) *CURVETE*(WO/CL, W/CD), and d) *CURVETE*(W/CLCD) **G**=5, and e) *CURVETE* (W/CLCD) **G**=10.

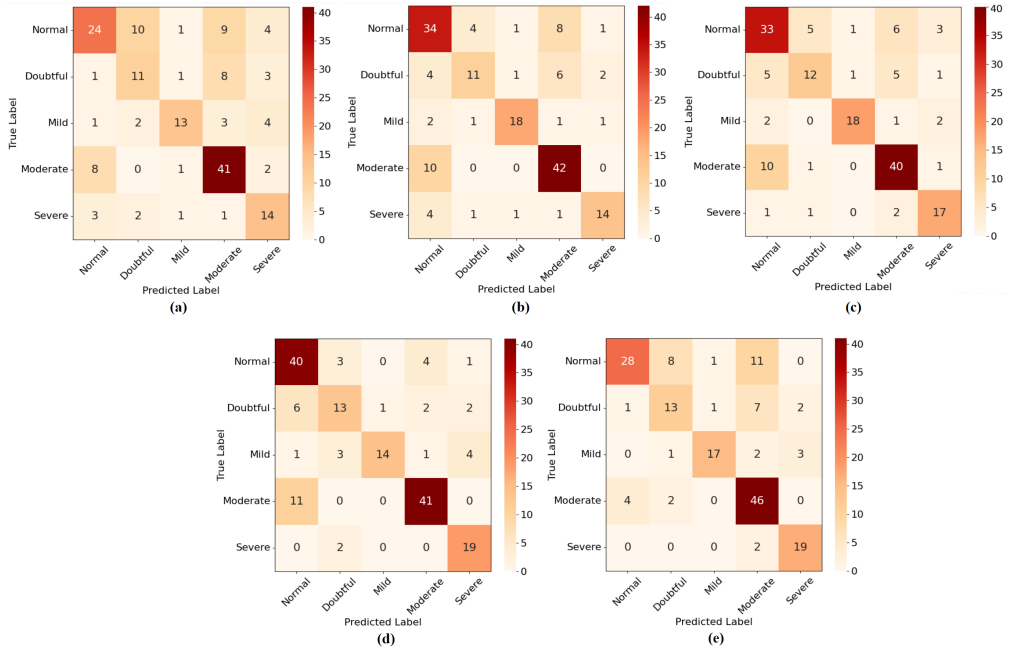

FIGURE 6.5: The confusion matrix results of the digital knee x-ray dataset obtained by: a) transfer learning with ResNet-50, b) *CURVETE*(WO/CLCD), c) *CURVETE*(WO/CL, W/CD), and d) *CURVETE*(W/CLCD) **G**=5, and e) *CURVETE*(W/CLCD) **G**=10.

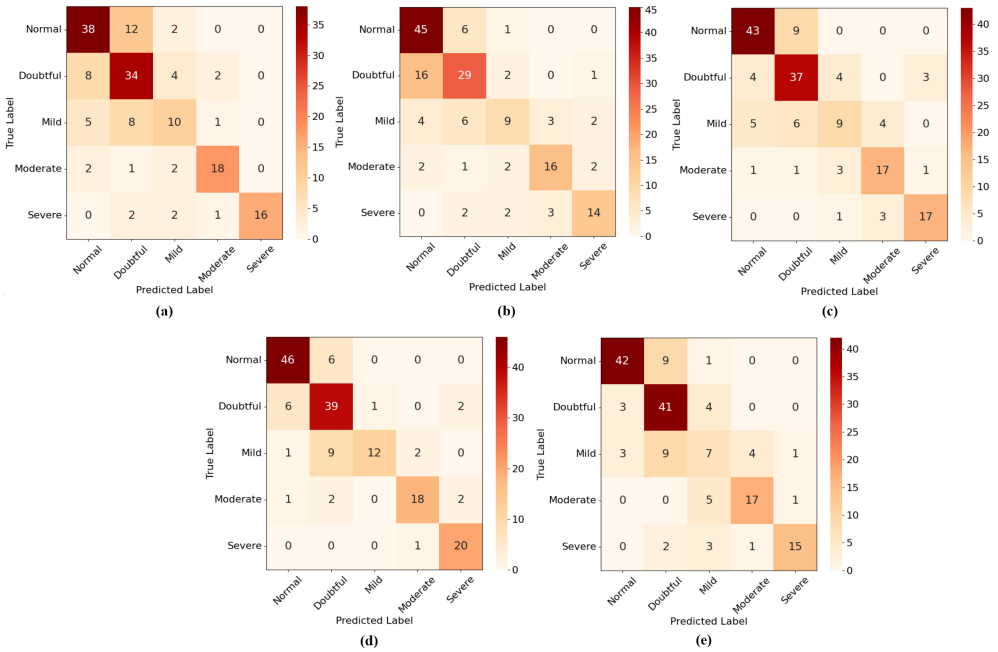

FIGURE 6.6: The confusion matrix results of the digital knee x-ray dataset obtained by: a) transfer learning with DenseNet-121, b) *CURVETE* (WO/CLCD), c) *CURVETE*(WO/CL, W/CD), and d) *CURVETE*(W/CLCD) **G**=5, and e) *CURVETE*(W/CLCD) **G**=10.

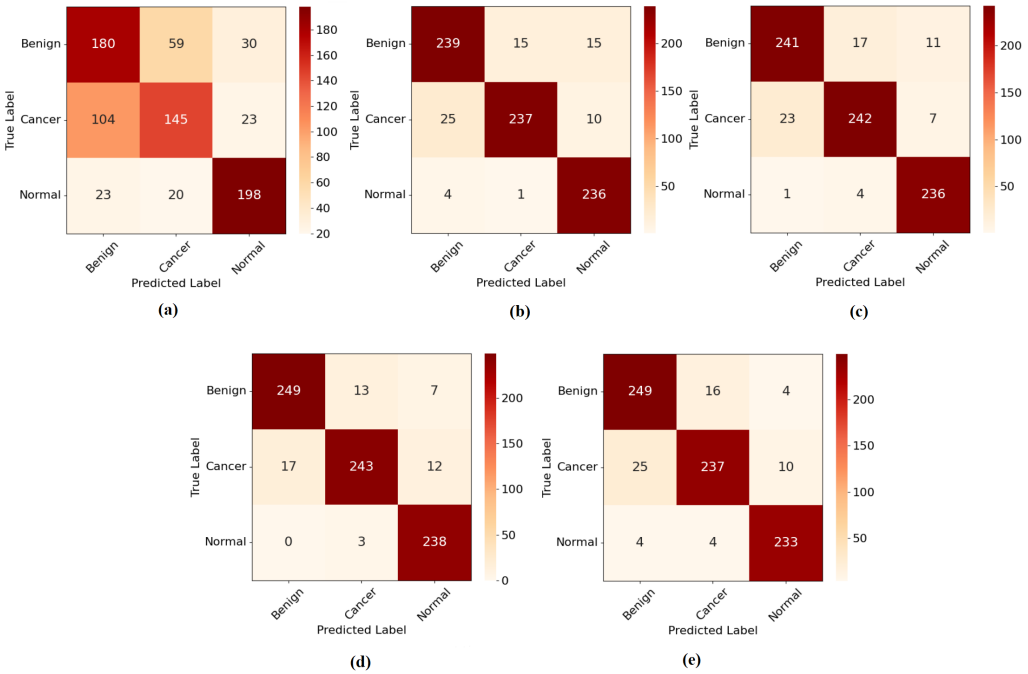

FIGURE 6.7: The confusion matrix results of the Mini-DDSM dataset obtained by: a) transfer learning with ResNet-50, b) *CURVETE*(WO/CLCD), c) *CURVETE*(WO/CL, W/CD), and d) *CURVETE*(W/CLCD) **G**=5, and e) *CURVETE*(W/CLCD) **G**=10.

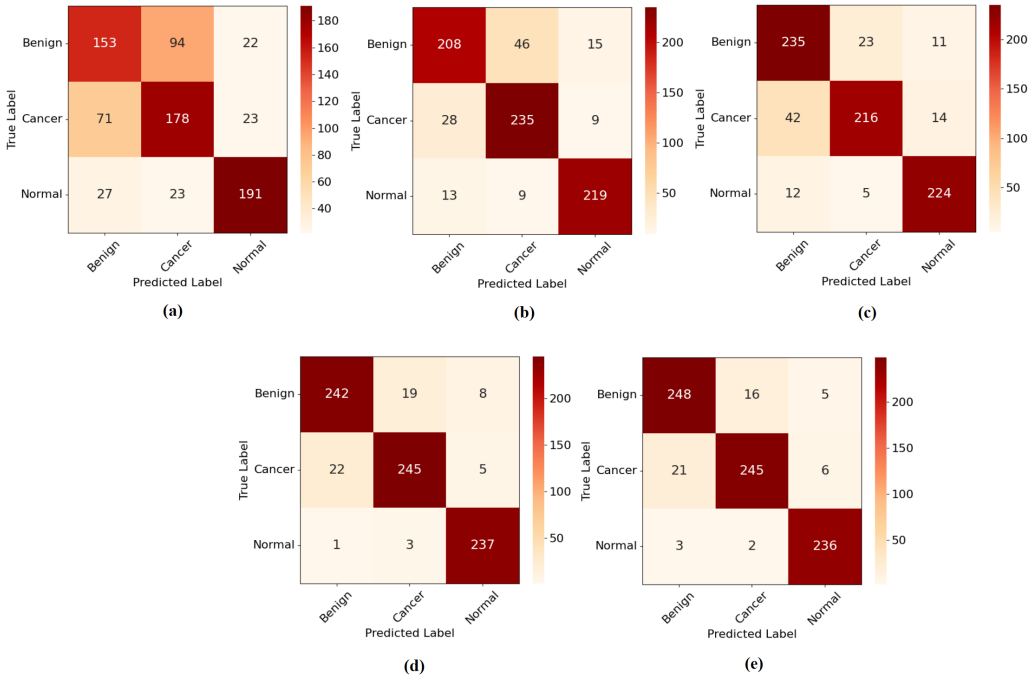

FIGURE 6.8: The confusion matrix results of the Mini-DDSM dataset obtained by: a) transfer learning with DenseNet-121, b) *CURVETE*(WO/CLCD), c) *CURVETE*(WO/CL, W/CD), and d) *CURVETE*(W/CLCD) **G**=5, and e) *CURVETE*(W/CLCD) **G**=10.

6.4.6 Comparison with State-of-the-art Methods

We compared the performance of *CURVETE* with other works on all the datasets; see Table 6.6. First, we compared *CURVETE* with *4S-DT*, which utilised self-supervised sample decomposition to improve the detection of COVID-19. As shown in Table 6.6, *CURVETE* consistently outperforms *4S-DT* across all datasets, with particularly notable improvements on the Mini-DDSM and digital knee x-ray datasets. The key difference lies in the integration of CL within *CURVETE*. While *4S-DT* focuses solely on decomposition during both pretext and downstream training phases, *CURVETE* introduces curriculum learning to guide the decomposition process through multiple levels of granularity. This enables the model to progressively refine feature transferability, leading to better generalisation on unseen data.

Second, in [188], the authors examined three SSL techniques using different pretrained networks to extract feature representations. For comparison, we implemented RotNet with ResNet-20 and DenseNet-121 as backbone networks. RotNet utilised SSL to train a model for learning image representations and then predicting the rotation angles applied to input images. Although this approach helps the model learn useful features, it relies heavily on the assumption that objects have clear and consistent orientations. Consequently, when images are noisy, contain complex patterns, or lack distinct shapes, RotNet often struggles to learn meaningful representations. In contrast, *CURVETE* focuses on the local patterns and refines the meaningful features gradually on different levels of complexity, making it more robust in the presence of irregular or complex data distributions. Finally, the authors in [189] investigated several SSL strategies for fine-grained image classification, including Jigsaw solving, SRGAN, and SimCLR. In our comparison, we focused on SimCLR and SRGAN, where SimCLR uses contrastive learning on augmented image pairs, and SRGAN applies super-resolution to enhance image details. However, SimCLR is sensitive to augmentation and often struggles with background distractions, while SRGAN introduces architectural complexity and risks overfitting on subtle features. In contrast, *CURVETE* avoids these issues by directly learning from the original images, guided by a structured curriculum and a decomposition mechanism. This leads to better generalisation performance, particularly in noisy or fine-grained datasets.

TABLE 6.6: *CURVETE*: Comparison with other state-of-the-art methods.

Reference	Method	brain tumour				digital knee x-ray				Mini-DDSM			
		ACC (%)	PR (%)	RE (%)	F1 (%)	ACC (%)	PR (%)	RE (%)	F1 (%)	ACC (%)	PR (%)	RE (%)	F1 (%)
[139]	4S-DT	91.38	90.20	90.61	90.41	68.45	70.29	69.29	29.79	71.36	71.29	71.85	71.57
[188]	RotNet-DenseNet-121	95.28	94.47	94.81	94.60	45.24	34.62	41.40	30.45	36.32	38.55	37.80	28.77
[188]	RotNet-ResNet20	79.35	77.87	75.92	75.47	58.93	72.22	58.93	56.87	60.36	63.36	60.70	60.79
[189]	SSL (SimCLR)	64.07	56.00	63.28	57.02	33.93	21.70	33.89	22.88	48.34	49.41	48.34	46.67
[189]	SSL (SRGAN)	43.41	25.64	43.41	29.01	65.03	77.40	65.19	57.40	72.90	73.89	71.99	73.20
Ours	*CURVETE*-ResNet-50	**96.60**	**95.82**	**96.56**	**96.19**	**75.60**	**76.54**	**73.54**	**75.01**	**93.35**	**93.35**	**93.55**	**93.45**
Ours	*CURVETE*-DenseNet-121	**95.77**	**95.18**	**96.15**	**95.16**	**80.36**	**83.24**	**78.64**	**80.87**	**93.22**	**93.29**	**93.40**	**93.35**

6.5 Summary

In this chapter, we introduced an SSL model called Curriculum Learning and Progressive Self-supervised Training for Medical Image Classification (*CURVETE*) to enhance the feature transferability from a large number of unlabelled samples to a new dataset with limited labelled samples of medical image datasets. *CURVETE* designed to employ the anti-CL strategy with sample decomposition for training the pretext task, where the model starts training with the highest level of granularity decomposition, allowing the model to simplify the complex pattern first and understand the local structure with the dataset before going to a more complex task with few subclasses within the granularity. By this process, the optimiser has more flexibility to discover a wide range of solutions and recognise meaningful patterns within the data, which enhances the learnt features to be more effective for a new supervised task. In addition, *CURVETE* utilised anti-CL with CD in the downstream task, which effectively handles the challenge of irregular class distributions, making it a highly adaptable solution for tasks like medical image classification. We experimentally demonstrated that *CURVETE* improves the generalisability across various medical image datasets compared to traditional transfer learning and other training models.

Chapter 7

Conclusion and Future Work

7.1 Overview

This thesis studies the impact of the data decomposition method on improving the performance of medical image classification tasks and overcoming the challenges of training medical image processing. The aim and objectives presented in this thesis focus on developing deep learning models to improve model performance and address the challenges of medical image datasets, such as the limited number of samples and overlapping classes.

The developed methods include: 1) developing a deep convolutional neural network (DCNN) model to enhance feature transferability by simplifying class structures, enabling effective learning of complex patterns within medical datasets; 2) introducing explainable and interpretable techniques to increase trust and usability in the medical imaging field; 3) addressing overlapping class boundaries and mitigating the challenges posed by limited sample sizes; and 4) improving the extraction of feature representations from the latent space to enable the model for better generalisation and making the features more effective for various tasks.

Chapter 1 presents an introduction to artificial intelligence techniques, emphasising their applications in healthcare and the primary challenges encountered in implementing these technologies. The motivation for addressing these challenges in medical imaging is then highlighted, followed by the identification of the aims and objectives for the work carried out in this thesis. Chapter 2 provides the necessary background and theoretical explanations of key concepts and methods essential for developing the contributions presented in this thesis. Chapter 3 reviews the literature work conducted on medical image classification using transfer learning strategies, as well as state-of-the-art methods that employ self-supervised learning and recent advancements in curriculum learning strategies for medical image processing. Chapter 4 discusses the first contribution, which addresses the first three objectives in Section 1.5. This chapter presents the *4S-DT* model and its advanced version, *XDecompo*, which enhances feature transferability through affinity-propagation-based class decomposition for downstream tasks. *XDecompo* has the ability to improve the classification performance and overcome the issue of overlapping distributions. Chapter 5 covers the second contribution, which addresses the

first and third objectives in Section 1.5. This chapter introduces the *CLOG-CD* model, utilising a curriculum learning strategy based on class decomposition to improve the learning process and handle class imbalance. Finally, Chapter 6 addresses the first and last two objectives in Section 1.5. It presents a self-supervised model that employs a curriculum-learning-based sample decomposition strategy to train a large set of unlabelled samples. This approach enhances feature representations from the pretext model and ultimately improves classification performance on downstream tasks with small datasets.

Overall, this thesis is summarised through two final sections. First, we discuss how the objectives of this work have been accomplished, highlighting the development of our three contributions. This section also considers potential implications and perspectives on the contributions presented. Finally, we outline prospective directions for future research to build upon this work.

7.2 Research Summary

In the context of medical image processing, convolutional neural networks (CNNs) have achieved remarkable success and gained the trust of many researchers. However, challenges remain that can complicate the training of deep learning models and limit their clinical applicability and reliability. One major challenge is the scarcity of labelled data, as expert annotation is both time-consuming and resource-intensive. Another significant challenge is the overlapping between classes, which is commonly seen in medical datasets. This issue can affect the model's ability to accurately distinguish between classes, leading to poor performance. Addressing these challenges is critical for increasing model robustness, generalisability, and eventually usability in real-world clinical applications.

To effectively address the challenges outlined in Section 1.5, several solution elements have been proposed, see Fig. 1.6. Class decomposition has gained attention as a preprocessing step in machine learning pipelines to handle irregular data distributions more effectively. By focusing on smaller, more homogenous groups, this method enables the model to better capture the distinct features of each subclass, leading to improved generalisation when handling data irregularities [8]. In addition, self-supervised learning addresses the challenge of limited labelled data by leveraging large amounts of unlabelled data to learn meaningful representations without extensive manual annotation. These learnt representations can then be fine-tuned for downstream tasks, improving performance and reducing the dependency on labelled datasets. Another notable approach is the curriculum learning strategy. It aims to improve the training process and enhance the generalisability of deep learning models, particularly when working with complex image datasets. Instead of presenting all information at once, curriculum learning organises the learning process into stages. These stages start with simpler information and gradually introduce more challenging concepts. This incremental approach helps the model build

its understanding progressively and accelerates the convergence of the training process.

We achieved our research objectives in Section 1.5 by introducing three research contributions. We designed, developed, and implemented our first contribution, *4S-DT* and its developed version, *XDecompo*, which is presented in Chapter 4. This contribution addresses the first three objectives in Section 1.5 which are: (1) developing a DCNN model to improve feature transferability between decomposed classes and simplify the complex structure within medical datasets; (2) incorporating explainable techniques into the machine learning model; and (3) addressing the challenges of overlapping classes and limited sample sizes. *4S-DT* and *XDecompo* are designed to address challenges related to limited labelled samples and irregular distributions. They achieve this by utilising self-supervised learning based on the sample decomposition method for training a large number of unlabelled samples. For the downstream task, *4S-DT* employs a predefined clustering algorithm to divide each class into a specific number of sub-classes. In contrast, *XDecompo* is guided by a non-parametric clustering method in the downstream task. This approach enables *XDecompo* to automatically identify meaningful class boundaries. As a result, it enhances the model's ability to capture relevant patterns that might be missed by traditional parametric clustering methods. Furthermore, *XDecompo* incorporates an explainability component that highlights critical pixels that contribute to classification, providing insights into how class decomposition improves the precision of extracted features. The effectiveness and implementation of the class decomposition method motivated us to introduce the second contribution in this thesis as follows:

- Introducing the class decomposition process as a powerful tool for curriculum learning strategy can lead to enhancing the learning process and improving the classification performance.
- The usage of curriculum learning with different granularities of decomposition allows for handling irregularities and complexities within the dataset. This approach simplifies the local structure within the classes, making the data easier to manage and analyse.
- Adapting anti-curriculum learning with different oscillations of granularity decomposition enables the model to capture meaningful features at different levels and enhances its understanding of specific patterns within the dataset. This ultimately enriches feature learning and can improve performance, especially on complex data such as medical imaging.

Therefore, we designed, developed, and implemented "CLOG-CD: Curriculum Learning based on Oscillating Granularity of Class Decomposed Medical Image Classification" as our second contribution in Chapter 5. This contribution aims to achieve the first and third objectives outlined in Section 1.5 which are specifically about: (1) improving the feature transferability between classes and simplifying the

complex structure; and (2) designing a DCNN model to enhance the classification performance and handling the overlapping distributions. *CLOG-CD* was designed to improve classification performance by simplifying the complex classification task and coping with any irregularities in the data distribution, which is a very common problem in the medical imaging domain. The model starts training at a high granularity level, where it faces a challenging classification task involving the maximum number of sub-classes. By leveraging the class decomposition approach, this challenge is mitigated as the complex classes are broken down into smaller, more homogeneous sub-classes. This enables the model to focus on learning the most relevant features within these simpler structures, making the classification task more manageable. Gradually, the model integrates this learnt knowledge as it transitions to fewer sub-classes (low granularity), refining its understanding of the class boundaries. This strategy not only speeds up training and promotes faster convergence but also handles irregularities and complexities in the dataset, resulting in improved performance. In *CLOG-CD*, deep local features are extracted from the dataset through the encoder layer of a convolutional auto-encoder. These features are then clustered using the k-means algorithm to generate different granularities of decomposition. For downstream task training, an anti-curriculum learning strategy is combined with class decomposition, starting at the highest granularity level (i.e., the most challenging task with the maximum number of sub-classes), and gradually the convergence weights are fine-tuned toward lower granularity levels (simpler classification tasks with fewer classes). *CLOG-CD* model has been compared with different training strategies and relevant models in the field, as shown in Table 5.2 and Table 5.3 in Chapter 5. The obtained results demonstrated that *CLOG-CD* outperformed all other models and showed how the integration of the class decomposition method and curriculum learning strategy with different oscillation steps has significantly improved the model's performance.

Both *XDecompo* and *CLOG-CD* illustrate the effectiveness of the class decomposition method in improving feature transferability and classification accuracy for medical images, which faced challenges such as irregular class representation. In addition, utilising the anti-curriculum learning strategy with the class decomposition method in *CLOG-CD* allows the model to control the learning process by first addressing the most complex tasks and simplifying complex patterns within each sub-class before moving on to easier tasks. This progressive approach enables the model to leverage what it has learnt from complex patterns, promoting a deeper understanding. These observations motivated us to introduce the third contribution in this thesis including:

- Integrating curriculum learning with sample decomposition in training the unlabelled samples allows the model to discover more meaningful patterns and latent structures within the data. This strategy improves the feature transferability to a new task, resulting in better performance in downstream applications.

- Utilising the class decomposition method with the curriculum learning strategy in the downstream task enhances the model's performance and effectively addresses the irregular distribution within the dataset.
- Implementing a single oscillation step between the granularity levels enables the model to capture more detailed information through different levels of granularity decomposition, leading to better performance.

Therefore, based on the above observations, we designed, developed, and implemented a model named "CURVETE: Curriculum Learning and Progressive Self-supervised Training for Medical Image Classification" presented in Chapter 6. This contribution achieves the first and last two objectives outlined in Section 1.5 which are about: (1) developing a CNN model that simplifies class structures, enabling better feature transferability to new tasks; (2) designing a DCNN model that has the ability to address the challenges of limited annotated datasets and manage irregularities in data distribution; and (3) developing a generalisable model that enhances the extraction of feature representations from the source task, ensuring their effectiveness and adaptability for various target tasks. *CURVETE* is a developed self-supervised framework that employs curriculum learning with a sample decomposition to train on a large set of unlabelled samples within a pretext model to improve robustness and enhance its capability to extract meaningful information from the data. The learnt features are then fine-tuned for a downstream task with a limited number of labelled samples, where the anti-curriculum learning combined with the class decomposition method effectively addresses irregular data distribution.

Based on these objectives, we achieved our aim of introducing DCNN models to boost convergence and enhance the classification performance of medical image datasets, particularly those with overlapping distributions and limited annotated samples.

7.3 Limitations

Although this thesis has introduced developed CNN models to improve medical image classification, it has some challenges. First, the training process was computationally intensive and time-consuming, which may limit scalability. Therefore, an optimiser is required to effectively fine-tune the large number of parameters and reduce the model's complexity.

Additionally, although the datasets used were imbalanced, the proposed models aimed to reduce the impact of overlapping within classes through different techniques such as curriculum learning and data decomposition. However, they did not specifically address class imbalance as an isolated issue. Integrating alternative techniques to solve imbalanced datasets could potentially enhance the quality of decomposition and further improve model performance.

7.4 Future Work

In computer vision, CNNs remain dominant and have achieved significant success across various fields due to their inductive biases, which enable them to effectively capture relationships between neighbouring features of an image using pooling and filters. However, the Vision Transformer (ViT) has recently demonstrated remarkable results, often outperforming convolutional neural networks in certain scenarios while requiring fewer computational resources for pre-training [190]. Unlike traditional CNNs, which rely on local receptive fields and convolution operations to extract features, VIT utilises self-attention mechanisms to capture long-range dependencies within images. This ability to model global relationships makes ViT particularly promising for complex medical image datasets, where fine-grained pattern recognition and accurate feature extraction are crucial for improving classification accuracy.

In future work, we plan to experiment with the ViT architecture in combination with data decomposition and a curriculum learning strategy to evaluate its potential to enhance model performance for medical image classification tasks. Due to its ability to capture long-range dependencies and model global relationships, it holds great promise in medical imaging, where fine-grained pattern recognition is crucial. This work will offer valuable insights into the applicability of ViT within the medical imaging domain, particularly for complex datasets.

Moreover, future work could explore the use of advanced strategies for solving the issues of the imbalanced dataset [191]. Class imbalance in medical image datasets can be addressed using various preprocessing techniques, such as sampling methods, which aim to balance the class distribution by directly modifying the data space. Another effective method is hybrid sampling, which combines different sampling methods (e.g., oversampling and undersampling) to overcome the limitations of each individual approach. In our future work, we will investigate the use of hybrid sampling techniques to evaluate their effectiveness in improving classification accuracy and handling class imbalance more efficiently.

In conclusion, this thesis contributes to the field of medical image analysis by developing a DCNN model that enhances feature transferability, enabling more accurate and efficient classification performance. Moreover, it introduces novel models incorporating a curriculum learning strategy with data decomposition to improve the training process and enhance generalisation across a variety of medical image datasets.

Bibliography

[1] Kunio Doi. "Computer-aided diagnosis in medical imaging: historical review, current status and future potential". In: *Computerized medical imaging and graphics* 31.4-5 (2007), pp. 198–211.

[2] Geoff Currie et al. "Machine learning and deep learning in medical imaging: intelligent imaging". In: *Journal of medical imaging and radiation sciences* 50.4 (2019), pp. 477–487.

[3] Nima Tajbakhsh et al. "Convolutional neural networks for medical image analysis: Full training or fine tuning?" In: *IEEE transactions on medical imaging* 35.5 (2016), pp. 1299–1312.

[4] Samir S Yadav and Shivajirao M Jadhav. "Deep convolutional neural network based medical image classification for disease diagnosis". In: *Journal of Big data* 6.1 (2019), pp. 1–18.

[5] Hee E Kim et al. "Transfer learning for medical image classification: a literature review". In: *BMC medical imaging* 22.1 (2022), p. 69.

[6] Rajat Raina et al. "Self-taught learning: transfer learning from unlabeled data". In: *Proceedings of the 24th international conference on Machine learning*. 2007, pp. 759–766.

[7] José A Sáez, Mikel Galar, and Bartosz Krawczyk. "Addressing the overlapping data problem in classification using the one-vs-one decomposition strategy". In: *IEEE Access* 7 (2019), pp. 83396–83411.

[8] Asmaa Abbas, Mohammed M Abdelsamea, and Mohamed Medhat Gaber. "Detrac: Transfer learning of class decomposed medical images in convolutional neural networks". In: *IEEE Access* 8 (2020), pp. 74901–74913.

[9] Ricardo Vilalta, M-K Achari, and Christoph F Eick. "Class decomposition via clustering: a new framework for low-variance classifiers". In: *Third IEEE International Conference on Data Mining*. IEEE. 2003, pp. 673–676.

[10] Lior Rokach. "Decomposition methodology for classification tasks: a meta decomposer framework". In: *Pattern Analysis and Applications* 9 (2006), pp. 257–271.

[11] Long Gao et al. "Handling imbalanced medical image data: A deep-learning-based one-class classification approach". In: *Artificial intelligence in medicine* 108 (2020), p. 101935.

[12] Eyad Elyan, Carlos Francisco Moreno-Garcia, and Chrisina Jayne. "CDSMOTE: class decomposition and synthetic minority class oversampling technique for imbalanced-data classification". In: *Neural computing and applications* 33 (2021), pp. 2839–2851.

[13] Mingsheng Long et al. "Learning transferable features with deep adaptation networks". In: *International conference on machine learning*. PMLR. 2015, pp. 97–105.

[14] Jason Yosinski et al. "How transferable are features in deep neural networks?" In: *Advances in neural information processing systems* 27 (2014).

[15] Fouzia Altaf et al. "Going deep in medical image analysis: concepts, methods, challenges, and future directions". In: *IEEE access* 7 (2019), pp. 99540–99572.

[16] William R Crum, Oscar Camara, and Derek LG Hill. "Generalized overlap measures for evaluation and validation in medical image analysis". In: *IEEE transactions on medical imaging* 25.11 (2006), pp. 1451–1461.

[17] Longlong Jing and Yingli Tian. "Self-supervised visual feature learning with deep neural networks: A survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2020).

[18] Liang Chen et al. "Self-supervised learning for medical image analysis using image context restoration". In: *Medical image analysis* 58 (2019), p. 101539.

[19] Maryam M Najafabadi et al. "Deep learning applications and challenges in big data analytics". In: *Journal of big data* 2 (2015), pp. 1–21.

[20] Travers Ching et al. "Opportunities and obstacles for deep learning in biology and medicine". In: *Journal of the royal society interface* 15.141 (2018), p. 20170387.

[21] Yoshua Bengio et al. "Curriculum learning". In: *Proceedings of the 26th annual international conference on machine learning*. 2009, pp. 41–48.

[22] Ioannis D Apostolopoulos and Tzani A Mpesiana. "Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks". In: *Physical and Engineering Sciences in Medicine* (2020), p. 1.

[23] Joseph Paul Cohen, Paul Morrison, and Lan Dao. "COVID-19 image data collection". In: *arXiv preprint arXiv:2003.11597* (2020).

[24] Stefan Jaeger et al. "Automatic tuberculosis screening using chest radiographs". In: *IEEE transactions on medical imaging* 33.2 (2013), pp. 233–245.

[25] Sema Candemir et al. "Lung segmentation in chest radiographs using anatomical atlases with nonrigid registration". In: *IEEE transactions on medical imaging* 33.2 (2013), pp. 577–590.

[26] Daniel S Kermany et al. "Identifying medical diagnoses and treatable diseases by image-based deep learning". In: *Cell* 172.5 (2018), pp. 1122–1131.

[27] Xiaosong Wang et al. "Chestx-ray8: Hospital-scale chest x-ray database and benchmarks on weakly-supervised classification and localization of common thorax diseases". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 2097–2106.

[28] Muhammad EH Chowdhury et al. "Can AI help in screening viral and COVID-19 pneumonia?" In: *IEEE Access* 8 (2020), pp. 132665–132676.

[29] Tawsifur Rahman et al. "Exploring the effect of image enhancement techniques on COVID-19 detection using chest X-ray images". In: *Computers in biology and medicine* 132 (2021), p. 104319.

[30] Milica M Badža and Marko Č Barjaktarović. "Classification of brain tumors from MRI images using a convolutional neural network". In: *Applied Sciences* 10.6 (2020), p. 1999.

[31] Marc Macenko et al. "A method for normalizing histology slides for quantitative analysis". In: *2009 IEEE International Symposium on Biomedical Imaging: From Nano to Macro*. IEEE. 2009, pp. 1107–1110.

[32] S Gornale and P Patravali. "Digital Knee X-ray Images". In: *Mendeley Data* 1 (2020).

[33] Pingjun Chen. "Knee osteoarthritis severity grading dataset". In: *Mendeley Data* 1 (2018), pp. 21–23.

[34] Charitha Dissanayake Lekamlage et al. "Mini-DDSM: Mammography-based automatic age estimation". In: *2020 3rd International Conference on Digital Medicine and Image Processing*. 2020, pp. 1–6.

[35] P SUCKLING J. "The mammographic image analysis society digital mammogram database". In: *Digital Mammo* (1994), pp. 375–386.

[36] Imad A Basheer and Maha Hajmeer. "Artificial neural networks: fundamentals, computing, design, and application". In: *Journal of microbiological methods* 43.1 (2000), pp. 3–31.

[37] Charu C Aggarwal et al. *Neural networks and deep learning*. Vol. 10. 978. Springer, 2018.

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

[39] Yoshua Bengio, Ian Goodfellow, Aaron Courville, et al. *Deep learning*. Vol. 1. MIT press Cambridge, MA, USA, 2017.

[40] Sagar Sharma, Simone Sharma, and Anidhya Athaiya. "Activation functions in neural networks". In: *Towards Data Sci* 6.12 (2017), pp. 310–316.

[41] Xavier Glorot and Yoshua Bengio. "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings. 2010, pp. 249–256.

[42] Pieter-Tjerk De Boer et al. "A tutorial on the cross-entropy method". In: *Annals of operations research* 134 (2005), pp. 19–67.

[43] Robert Hecht-Nielsen. "Theory of the backpropagation neural network". In: *Neural networks for perception*. Elsevier, 1992, pp. 65–93.

[44] Jeffrey Dean et al. "Large scale distributed deep networks". In: *Advances in neural information processing systems* 25 (2012).

[45] Connor Shorten and Taghi M Khoshgoftaar. "A survey on image data augmentation for deep learning". In: *Journal of big data* 6.1 (2019), pp. 1–48.

[46] Xue Ying. "An overview of overfitting and its solutions". In: *Journal of physics: Conference series*. Vol. 1168. IOP Publishing. 2019, p. 022022.

[47] Lutz Prechelt. "Early stopping-but when?" In: *Neural Networks: Tricks of the trade*. Springer, 2002, pp. 55–69.

[48] Nitish Srivastava et al. "Dropout: a simple way to prevent neural networks from overfitting". In: *The journal of machine learning research* 15.1 (2014), pp. 1929–1958.

[49] Ilya Loshchilov and Frank Hutter. "Decoupled weight decay regularization". In: *arXiv preprint arXiv:1711.05101* (2017).

[50] Iqbal H Sarker. "Deep learning: a comprehensive overview on techniques, taxonomy, applications and research directions". In: *SN computer science* 2.6 (2021), p. 420.

[51] Iqbal Muhammad and Zhu Yan. "SUPERVISED MACHINE LEARNING APPROACHES: A SURVEY." In: *ICTACT Journal on Soft Computing* 5.3 (2015).

[52] Muhammad Usama et al. "Unsupervised machine learning for networking: Techniques, applications and research challenges". In: *IEEE access* 7 (2019), pp. 65579–65615.

[53] Beatriz Flamia Azevedo, Ana Maria AC Rocha, and Ana I Pereira. "Hybrid approaches to optimization and machine learning methods: a systematic literature review". In: *Machine Learning* 113.7 (2024), pp. 4055–4097.

[54] Rayan Krishnan, Pranav Rajpurkar, and Eric J Topol. "Self-supervised learning in medicine and healthcare". In: *Nature Biomedical Engineering* 6.12 (2022), pp. 1346–1352.

[55] Jie Gui et al. "A survey on self-supervised learning: Algorithms, applications, and future trends". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2024).

[56] Veenu Rani et al. "Self-supervised learning: A succinct review". In: *Archives of Computational Methods in Engineering* 30.4 (2023), pp. 2761–2775.

[57] Amelia Jiménez-Sánchez et al. "Medical-based deep curriculum learning for improved fracture classification". In: *Medical Image Computing and Computer Assisted Intervention–MICCAI 2019: 22nd International Conference, Shenzhen, China, October 13–17, 2019, Proceedings, Part VI 22*. Springer. 2019, pp. 694–702.

[58] Daphna Weinshall, Gad Cohen, and Dan Amir. "Curriculum learning by transfer learning: Theory and experiments with deep networks". In: *International conference on machine learning*. PMLR. 2018, pp. 5238–5246.

[59] Batta Mahesh et al. "Machine learning algorithms-a review". In: *International Journal of Science and Research (IJSR).[Internet]* 9.1 (2020), pp. 381–386.

[60] Yann LeCun et al. "Gradient-based learning applied to document recognition". In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.

[61] Emerald U Henry, Onyeka Emebob, and Conrad Asotie Omonhinmin. "Vision transformers in medical imaging: A review". In: *arXiv preprint arXiv:2211.10043* (2022).

[62] Larry R Medsker, Lakhmi Jain, et al. "Recurrent neural networks". In: *Design and Applications* 5.64-67 (2001), p. 2.

[63] Kangrui Lu, Yuanrun Xu, and Yige Yang. "Comparison of the potential between transformer and CNN in image classification". In: *ICMLCA 2021; 2nd International Conference on Machine Learning and Computer Application*. VDE. 2021, pp. 1–6.

[64] Waseem Rawat and Zenghui Wang. "Deep convolutional neural networks for image classification: A comprehensive review". In: *Neural computation* 29.9 (2017), pp. 2352–2449.

[65] Manjunath Jogin et al. "Feature extraction using convolution neural networks (CNN) and deep learning". In: *2018 3rd IEEE international conference on recent trends in electronics, information & communication technology (RTEICT)*. IEEE. 2018, pp. 2319–2323.

[66] Manli Sun et al. "Learning pooling for convolutional neural network". In: *Neurocomputing* 224 (2017), pp. 96–104.

[67] Hugo Larochelle et al. "Exploring strategies for training deep neural networks." In: *Journal of machine learning research* 10.1 (2009).

[68] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. "A survey of transfer learning". In: *Journal of Big data* 3 (2016), pp. 1–40.

[69] Kaiming He et al. "Spatial pyramid pooling in deep convolutional networks for visual recognition". In: *IEEE transactions on pattern analysis and machine intelligence* 37.9 (2015), pp. 1904–1916.

[70] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. "Imagenet classification with deep convolutional neural networks". In: *Advances in neural information processing systems*. 2012, pp. 1097–1105.

[71] François Chollet. "Xception: Deep learning with depthwise separable convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 1251–1258.

[72] Gao Huang et al. "Densely connected convolutional networks". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 4700–4708.

[73] Christian Szegedy et al. "Going deeper with convolutions". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015, pp. 1–9.

[74] Kaiming He et al. "Deep residual learning for image recognition". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 770–778.

[75] Lorenzo Putzu, Luca Piras, and Giorgio Giacinto. "Convolutional neural networks for relevance feedback in content based image retrieval: A Content based image retrieval system that exploits convolutional neural networks both for feature extraction and for relevance feedback". In: *Multimedia Tools and Applications* 79.37 (2020), pp. 26995–27021.

[76] Vadthe Narasimha and Dr M Dhanalakshmi. "Detection and severity identification of Covid-19 in chest X-ray images using deep learning". In: *International Journal of Electrical and Electronics Research* 10.2 (2022), pp. 364–369.

[77] Gyan Singh Sujawat and Awanit Kumar. "A Deep Learning-Based Methodology For Plant Disease Identification And Diagnosis". In: *Webology (ISSN: 1735-188X)* 18.1 (2021).

[78] Geert Litjens et al. "A survey on deep learning in medical image analysis". In: *Medical image analysis* 42 (2017), pp. 60–88.

[79] DR Sarvamangala and Raghavendra V Kulkarni. "Convolutional neural networks in medical image understanding: a survey". In: *Evolutionary intelligence* 15.1 (2022), pp. 1–22.

[80] Prabira Kumar Sethy and Santi Kumari Behera. "Detection of Coronavirus Disease (COVID-19) Based on Deep Features". In: (2020).

[81] Tawsifur Rahman et al. "Transfer learning with deep convolutional neural network (CNN) for pneumonia detection using chest X-ray". In: *Applied Sciences* 10.9 (2020), p. 3233.

[82] Alhassan Mabrouk et al. "Pneumonia detection on chest X-ray images using ensemble of deep convolutional neural networks". In: *Applied Sciences* 12.13 (2022), p. 6448.

[83] Shudong Wang et al. "Classification of pathological types of lung cancer from CT images by deep residual neural networks with transfer learning strategy". In: *Open Medicine* 15.1 (2020), pp. 190–197.

[84] Ravi K Samala et al. "Multi-task transfer learning deep convolutional neural network: application to computer-aided diagnosis of breast cancer on mammograms". In: *Physics in Medicine & Biology* 62.23 (2017), p. 8894.

[85] SanaUllah Khan et al. "A novel deep learning based framework for the detection and classification of breast cancer using transfer learning". In: *Pattern Recognition Letters* 125 (2019), pp. 1–6.

[86] Mohammad Alkhaleefah et al. "The influence of image augmentation on breast lesion classification using transfer learning". In: *2020 International Conference on Artificial Intelligence and Signal Processing (AISP)*. IEEE. 2020, pp. 1–5.

[87] Gelan Ayana et al. "A novel multistage transfer learning for ultrasound breast cancer image classification". In: *Diagnostics* 12.1 (2022), p. 135.

[88] Abeer Saber et al. "A novel deep-learning model for automatic detection and classification of breast cancer using the transfer-learning technique". In: *IEEe Access* 9 (2021), pp. 71194–71209.

[89] Laith Alzubaidi et al. "Optimizing the performance of breast cancer classification by employing the same domain transfer learning from hybrid deep convolutional neural network model". In: *Electronics* 9.3 (2020), p. 445.

[90] Junaid Malik et al. "Colorectal cancer diagnosis from histology images: A comparative study". In: *arXiv preprint arXiv:1903.11210* (2019).

[91] Elene Firmeza Ohata et al. "A novel transfer learning approach for the classification of histological images of colorectal cancer". In: *The Journal of Supercomputing* (2021), pp. 1–26.

[92] Naresh Kumar et al. "An empirical study of handcrafted and dense feature extraction techniques for lung and colon cancer classification from histopathological images". In: *Biomedical Signal Processing and Control* 75 (2022), p. 103596.

[93] Javeria Amin et al. "A new approach for brain tumor segmentation and classification based on score level fusion using transfer learning". In: *Journal of medical systems* 43 (2019), pp. 1–16.

[94] S Deepak and PM Ameer. "Brain tumor classification using deep CNN features via transfer learning". In: *Computers in biology and medicine* 111 (2019), p. 103345.

[95] Srinath Kokkalla et al. "Three-class brain tumor classification using deep dense inception residual network". In: *Soft Computing* 25.13 (2021), pp. 8721–8729.

[96] Guo Haixiang et al. "Learning from class-imbalanced data: Review of methods and applications". In: *Expert systems with applications* 73 (2017), pp. 220–239.

[97] Sotiris Kotsiantis, Dimitris Kanellopoulos, Panayiotis Pintelas, et al. "Handling imbalanced datasets: A review". In: *GESTS international transactions on computer science and engineering* 30.1 (2006), pp. 25–36.

[98] Eyad Elyan and Mohamed Medhat Gaber. "A fine-grained random forests using class decomposition: an application to medical diagnosis". In: *Neural computing and applications* 27.8 (2016), pp. 2279–2288.

[99] Axel H Masquelin et al. "Wavelet decomposition facilitates training on small datasets for medical image classification by deep learning". In: *Histochemistry and cell biology* 155.2 (2021), pp. 309–317.

[100] Inese Polaka et al. "Clustering algorithm specifics in class decomposition". In: *No: Applied Information and Communication Technology* (2013).

[101] Pattaramon Vuttipittayamongkol and Eyad Elyan. "Overlap-based undersampling method for classification of imbalanced medical datasets". In: *Artificial Intelligence Applications and Innovations: 16th IFIP WG 12.5 International Conference, AIAI 2020, Neos Marmaras, Greece, June 5–7, 2020, Proceedings, Part II 16*. Springer. 2020, pp. 358–369.

[102] Kemal Polat. "Similarity-based attribute weighting methods via clustering algorithms in the classification of imbalanced medical datasets". In: *Neural Computing and Applications* 30 (2018), pp. 987–1013.

[103] Kouhei Shimizu et al. "Four-class classification of skin lesions with task decomposition strategy". In: *IEEE transactions on biomedical engineering* 62.1 (2014), pp. 274–283.

[104] Tunc Gultekin et al. "Two-tier tissue decomposition for histopathological image representation and classification". In: *IEEE transactions on medical imaging* 34.1 (2014), pp. 275–283.

[105] Anjan Gudigar et al. "Automated categorization of multi-class brain abnormalities using decomposition techniques with MRI images: A comparative study". In: *IEEE Access* 7 (2019), pp. 28498–28509.

[106] Maha M Alwuthaynani, Zahraa S Abdallah, and Raul Santos-Rodriguez. "A robust class decomposition-based approach for detecting Alzheimer's progression". In: *Experimental Biology and Medicine* 248.24 (2023), pp. 2514–2525.

[107] Nassima Dif et al. "Transfer learning from synthetic labels for histopathological images classification". In: *Applied Intelligence* 52.1 (2022), pp. 358–377.

[108] Asmaa Abbas, Mohammed M Abdelsamea, and Mohamed Medhat Gaber. "Classification of COVID-19 in chest X-ray images using DeTraC deep convolutional neural network". In: *Applied Intelligence* 51 (2021), pp. 854–864.

[109] Afnan M Alhassan. "Thresholding Chaotic Butterfly Optimization Algorithm with Gaussian Kernel (TCBOGK) based segmentation and DeTrac deep convolutional neural network for COVID-19 X-ray images". In: *Multimedia Tools and Applications* 83.26 (2024), pp. 68317–68340.

[110] Hoo-Chang Shin et al. "Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning". In: *IEEE transactions on medical imaging* 35.5 (2016), pp. 1285–1298.

[111] Maithra Raghu et al. "Transfusion: Understanding transfer learning for medical imaging". In: *Advances in neural information processing systems* 32 (2019).

[112] Xingyi Yang et al. "Transfer learning or self-supervised learning? a tale of two pretraining paradigms". In: *arXiv preprint arXiv:2007.04234* (2020).

[113] Guang Li et al. "COVID-19 detection based on self-supervised transfer learning using chest X-ray images". In: *International Journal of Computer Assisted Radiology and Surgery* 18.4 (2023), pp. 715–722.

[114] Matej Gazda et al. "Self-supervised deep convolutional neural network for chest X-ray classification". In: *IEEE Access* 9 (2021), pp. 151972–151982.

[115] Hari Sowrirajan et al. "Moco pretraining improves representation and transferability of chest x-ray models". In: *Medical Imaging with Deep Learning*. PMLR. 2021, pp. 728–744.

[116] Kyungjin Cho et al. "Chess: Chest x-ray pre-trained model via self-supervised contrastive learning". In: *Journal of Digital Imaging* 36.3 (2023), pp. 902–910.

[117] Xinlei Chen et al. "Improved baselines with momentum contrastive learning". In: *arXiv preprint arXiv:2003.04297* (2020).

[118] Ozan Ciga, Tony Xu, and Anne Louise Martel. "Self supervised contrastive learning for digital histopathology". In: *Machine Learning with Applications* 7 (2022), p. 100198.

[119] Navid Alemi Koohbanani et al. "Self-path: Self-supervision for classification of pathology images with limited annotations". In: *IEEE Transactions on Medical Imaging* 40.10 (2021), pp. 2845–2856.

[120] Xuan-Bac Nguyen et al. "Self-supervised learning based on spatial awareness for medical image analysis". In: *IEEE access* 8 (2020), pp. 162973–162981.

[121] Yueyue Wang et al. "Self-supervised learning and semi-supervised learning for multi-sequence medical image classification". In: *Neurocomputing* 513 (2022), pp. 383–394.

[122] Animesh Mishra, Ritesh Jha, and Vandana Bhattacharjee. "SSCLNet: a self-supervised contrastive loss-based pre-trained network for brain MRI classification". In: *IEEE Access* 11 (2023), pp. 6673–6681.

[123] Mathilde Caron et al. "Deep clustering for unsupervised learning of visual features". In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 132–149.

[124] William Lotter, Greg Sorensen, and David Cox. "A multi-scale CNN and curriculum learning strategy for mammogram classification". In: *Deep Learning in Medical Image Analysis and Multimodal Learning for Clinical Decision Support: Third International Workshop, DLMIA 2017, and 7th International Workshop, ML-CDS 2017, Held in Conjunction with MICCAI 2017, Québec City, QC, Canada, September 14, Proceedings 3*. Springer. 2017, pp. 169–177.

[125] Andrew Jesson et al. "CASED: curriculum adaptive sampling for extreme data imbalance". In: *International conference on medical image computing and computer-assisted intervention*. Springer. 2017, pp. 639–646.

[126] Jun Luo et al. "Deep curriculum learning in task space for multi-class based mammography diagnosis". In: *Medical Imaging 2022: Computer-Aided Diagnosis*. Vol. 12033. SPIE. 2022, pp. 71–76.

[127] Beomhee Park et al. "A curriculum learning strategy to enhance the accuracy of classification of various lesions in chest-PA X-ray screening for pulmonary abnormalities". In: *Scientific reports* 9.1 (2019), p. 15352.

[128] Mei Yang et al. "Su-micl: severity-guided multiple instance curriculum learning for histopathology image interpretable classification". In: *IEEE Transactions on Medical Imaging* 41.12 (2022), pp. 3533–3543.

[129] Yuxing Tang et al. "Attention-guided curriculum learning for weakly supervised classification and localization of thoracic diseases on chest radiographs". In: *Machine Learning in Medical Imaging: 9th International Workshop, MLMI 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Proceedings 9*. Springer. 2018, pp. 249–258.

[130] Jerry Wei et al. "Learn like a pathologist: curriculum learning by annotator agreement for histopathology image classification". In: *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 2021, pp. 2473–2483.

[131] Amelia Jiménez-Sánchez et al. "Curriculum learning for improved femur fracture classification: Scheduling data with prior knowledge and uncertainty". In: *Medical Image Analysis* 75 (2022), p. 102273.

[132] Chetan L Srinidhi and Anne L Martel. "Improving self-supervised learning with hardness-aware dynamic curriculum learning: an application to digital pathology". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2021, pp. 562–571.

[133] Fengbei Liu et al. "Acpl: Anti-curriculum pseudo-labelling for semi-supervised medical image classification". In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2022, pp. 20697–20706.

[134] Mihail Burduja and Radu Tudor Ionescu. "Unsupervised medical image alignment with curriculum learning". In: *2021 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2021, pp. 3787–3791.

[135] Mohammad Alsharid et al. "A curriculum learning based approach to captioning ultrasound images". In: *Medical Ultrasound, and Preterm, Perinatal and Paediatric Image Analysis: First International Workshop, ASMUS 2020, and 5th International Workshop, PIPPI 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 4-8, 2020, Proceedings 1*. Springer. 2020, pp. 75–84.

[136] Yiru Wang et al. "Dynamic curriculum learning for imbalanced data classification". In: *Proceedings of the IEEE/CVF international conference on computer vision*. 2019, pp. 5017–5026.

[137] Xiangyu Li et al. "Curriculum label distribution learning for imbalanced medical image segmentation". In: *Medical Image Analysis* 89 (2023), p. 102911.

[138] Rongchang Zhao et al. "Diagnosing glaucoma on imbalanced data with self-ensemble dual-curriculum learning". In: *Medical image analysis* 75 (2022), p. 102295.

[139] Asmaa Abbas, Mohammed M Abdelsamea, and Mohamed Medhat Gaber. "4S-DT: Self-Supervised Super Sample Decomposition for Transfer Learning With Application to COVID-19 Detection". In: *IEEE Transactions on Neural Networks and Learning Systems* (2021).

[140] Asmaa Abbas, Mohamed Medhat Gaber, and Mohammed M Abdelsamea. "Xdecompo: explainable decomposition approach in convolutional neural networks for tumour image classification". In: *Sensors* 22.24 (2022), p. 9875.

[141] Veronika Cheplygina, Marleen De Bruijne, and Josien PW Pluim. "Not-so-supervised: a survey of semi-supervised, multi-instance, and transfer learning in medical image analysis". In: *Medical image analysis* 54 (2019), pp. 280–296.

[142] S Sharma et al. "Stacked autoencoders for medical image search". In: *International Symposium on Visual Computing*. Springer. 2016, pp. 45–54.

[143] Martin Ester et al. "A density-based algorithm for discovering clusters in large spatial databases with noise." In: *kdd*. Vol. 96. 34. 1996, pp. 226–231.

[144] Marina Sokolova and Guy Lapalme. "A systematic analysis of performance measures for classification tasks". In: *Information processing & management* 45.4 (2009), pp. 427–437.

[145] Frank Wilcoxon. "Individual comparisons by ranking methods". In: *Breakthroughs in statistics*. Springer, 1992, pp. 196–202.

[146] Min Chen et al. "Deep features learning for medical image analysis with convolutional autoencoder neural network". In: *IEEE Transactions on Big Data* (2017).

[147] Brendan J Frey and Delbert Dueck. "Clustering by passing messages between data points". In: *science* 315.5814 (2007), pp. 972–976.

[148] Alejandro Barredo Arrieta et al. "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI". In: *Information Fusion* 58 (2020), pp. 82–115.

[149] Matthew D Zeiler and Rob Fergus. "Visualizing and understanding convolutional networks". In: *European conference on computer vision*. Springer. 2014, pp. 818–833.

[150] Riccardo Guidotti et al. "A survey of methods for explaining black box models". In: *ACM computing surveys (CSUR)* 51.5 (2018), pp. 1–42.

[151] David Gunning et al. "XAI—Explainable artificial intelligence". In: *Science Robotics* 4.37 (2019).

[152] Amina Adadi and Mohammed Berrada. "Peeking inside the black-box: a survey on explainable artificial intelligence (XAI)". In: *IEEE access* 6 (2018), pp. 52138–52160.

[153] Rich Caruana et al. "Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission". In: *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*. 2015, pp. 1721–1730.

[154] Luisa M Zintgraf et al. "Visualizing deep neural network decisions: Prediction difference analysis". In: *arXiv preprint arXiv:1702.04595* (2017).

[155] Sarmad Maqsood, Robertas Damaševičius, and Rytis Maskeliūnas. "Multi-modal brain tumor detection using deep neural network and multiclass SVM". In: *Medicina* 58.8 (2022), p. 1090.

[156] Morteza Esmaeili et al. "Explainable artificial intelligence for human-machine interaction in brain tumor localization". In: *Journal of personalized medicine* 11.11 (2021), p. 1213.

[157] Linda Wang, Zhong Qiu Lin, and Alexander Wong. "Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest x-ray images". In: *Scientific reports* 10.1 (2020), p. 19549.

[158] Mohan Bhandari et al. "Explanatory classification of CXR images into COVID-19, Pneumonia and Tuberculosis using deep learning and XAI". In: *Computers in Biology and Medicine* 150 (2022), p. 106156.

[159] Patrik Sabol et al. "Explainable classifier for improving the accountability in decision-making for colorectal cancer diagnosis from histopathological images". In: *Journal of biomedical informatics* 109 (2020), p. 103523.

[160] Saumya Jetley et al. *Learn To Pay Attention*. 2018. arXiv: 1804.02391 [cs.CV].

[161] Zhaoyang Niu, Guoqiang Zhong, and Hui Yu. "A review on the attention mechanism of deep learning". In: *Neurocomputing* 452 (2021), pp. 48–62.

[162] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. *Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps*. 2014. arXiv: 1312.6034 [cs.CV].

[163] Bolei Zhou et al. "Learning deep features for discriminative localization". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2921–2929.

[164] Ramprasaath R Selvaraju et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 618–626.

[165] Ramprasaath R Selvaraju et al. "Grad-CAM: Why did you say that?" In: *arXiv preprint arXiv:1611.07450* (2016).

[166] Tingying Peng et al. "Multi-task learning of a deep k-nearest neighbour network for histopathological image classification and retrieval". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2019, pp. 676–684.

[167] Sourodip Ghosh et al. "Colorectal histology tumor detection using ensemble deep neural network". In: *Engineering Applications of Artificial Intelligence* 100 (2021), p. 104202.

[168] Xingyu Li et al. "Colorectal cancer survival prediction using deep distribution based multiple-instance learning". In: *Entropy* 24.11 (2022), p. 1669.

[169] Jakob Nikolas Kather et al. "Predicting survival from colorectal cancer histology slides using deep learning: A retrospective multicenter study". In: *PLoS medicine* 16.1 (2019), e1002730.

[170] Nyoman Abiwinanda et al. "Brain tumor classification using convolutional neural network". In: *World congress on medical physics and biomedical engineering 2018*. Springer. 2019, pp. 183–189.

[171] Parnian Afshar, Konstantinos N Plataniotis, and Arash Mohammadi. "Capsule networks for brain tumor classification based on MRI images and coarse tumor boundaries". In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2019, pp. 1368–1372.

[172] Jun Cheng et al. "Enhanced performance of brain tumor classification via tumor region augmentation and partition". In: *PloS one* 10.10 (2015), e0140381.

[173] Ali Pashaei, Hedieh Sajedi, and Niloofar Jazayeri. "Brain Tumor Classification via Convolutional Neural Network and Extreme Learning Machines". In: *2018 8th International Conference on Computer and Knowledge Engineering (ICCKE)*. 2018, pp. 314–319. DOI: 10.1109/ICCKE.2018.8566571.

[174] Tahia Tazin et al. "A robust and novel approach for brain tumor classification using convolutional neural network". In: *Computational Intelligence and Neuroscience* 2021 (2021).

[175] Petru Soviany et al. "Curriculum self-paced learning for cross-domain object detection". In: *Computer Vision and Image Understanding* 204 (2021), p. 103166.

[176] Bradley Efron. "Better bootstrap confidence intervals". In: *Journal of the American statistical Association* 82.397 (1987), pp. 171–185.

[177] Chaoyi Li et al. "Dynamic Curriculum Learning via In-Domain Uncertainty for Medical Image Classification". In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2023, pp. 747–757.

[178] Asma Naseer et al. "Deep learning classifiers for computer-aided diagnosis of multiple lungs disease". In: *Journal of X-Ray Science and Technology* 31.5 (2023), pp. 1125–1143.

[179] Emtiaz Hussain et al. "CoroDet: A deep learning based classification for COVID-19 detection using chest X-ray images". In: *Chaos, Solitons & Fractals* 142 (2021), p. 110495.

[180] Tahia Tazin et al. "[Retracted] A Robust and Novel Approach for Brain Tumor Classification Using Convolutional Neural Network". In: *Computational Intelligence and Neuroscience* 2021.1 (2021), p. 2392395.

[181] Neelum Noreen et al. "Brain Tumor Classification Based on Fine-Tuned Models and the Ensemble Method." In: *Computers, Materials & Continua* 67.3 (2021).

[182] Weiqiang Liu et al. "A novel focal ordinal loss for assessment of knee osteoarthritis severity". In: *Neural Processing Letters* 54.6 (2022), pp. 5199–5224.

[183] Pingjun Chen et al. "Fully automatic knee osteoarthritis severity grading using deep neural networks with a novel ordinal loss". In: *Computerized Medical Imaging and Graphics* 75 (2019), pp. 84–92.

[184] Rima Tri Wahyuningrum et al. "A new approach to classify knee osteoarthritis severity from radiographic images based on CNN-LSTM method". In: *2019 IEEE 10th International Conference on Awareness Science and Technology (iCAST)*. IEEE. 2019, pp. 1–6.

[185] Wentao Hu et al. "Learning from Incorrectness: Active Learning with Negative Pre-training and Curriculum Querying for Histological Tissue Classification". In: *IEEE Transactions on Medical Imaging* (2023).

[186] Yun Liu et al. "Detecting cancer metastases on gigapixel pathology images". In: *arXiv preprint arXiv:1703.02442* (2017).

[187] David Bau et al. "Network dissection: Quantifying interpretability of deep visual representations". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017, pp. 6541–6549.

[188] Alan Preciado-Grijalva et al. "Self-supervised learning for sonar image classification". In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2022, pp. 1499–1508.

[189] Farha Al Breiki, Muhammad Ridzuan, and Rushali Grandhe. "Self-supervised learning for fine-grained image classification". In: *arXiv preprint arXiv:2107.13973* (2021).

[190] Alexey Dosovitskiy et al. "An image is worth 16x16 words: Transformers for image recognition at scale". In: *arXiv preprint arXiv:2010.11929* (2020).

[191] M Kumar and HS Sheshadri. "On the classification of imbalanced datasets". In: *International Journal of Computer Applications* 44.8 (2012), pp. 1–7.