Electronics and Computer Science
Faculty of Physical Sciences and Engineering
University of Southampton

Author: Apishan Thayananthan
Date: 29/04/24

Investigating How to Empower Clinicians with AI for CT Scan Classification

Project Supervisor: Dr. Mercedes Arguello Casteleiro
Second Examiner: Dr. Stuart Middleton

A project progress report submitted for the award of
MEng Computer Science with Industial Placement

Abstract

This paper details developing and evaluating a streamlined AI system designed to simplify the application of machine learning models for medical imaging diagnostics, specifically CT scan classification. This project's key breakthrough is simplifying the usual complex machine-learning processes into only three lines of code. The study aims to organise data, train and assess models, and classify medical images with minimal technical expertise. The system supports AI models such as CNNs, Swin Transformers, Hugging Face ViT and Custom Neural Network models, ensuring robust performance across varied datasets. The models were tested on medical imaging datasets for lung cancer and osteoporosis, demonstrating high accuracy and providing visual insights as we hypothesise that heat maps may support better interpretability. This study investigates the feasibility of lowering the technical barrier and, thus, facilitating the adoption of the latest trends in AI. The main contributions of this research lie in its potential to democratise advanced diagnostic tools, making them readily usable without requiring deep computational expertise.

Statement of Originality

- I have read and understood the ECS Academic Integrity information and the University's Academic Integrity Guidance for Students.
- I am aware that failure to act in accordance with the Regulations Governing Academic Integrity may lead to the imposition of penalties which, for the most serious cases, may include termination of programme.
- I consent to the University copying and distributing any or all of my work in any form and using third parties (who may be based outside the EU/EEA) to verify whether my work contains plagiarised material, and for quality assurance purposes.
- I have acknowledged all sources, and identified any content taken from elsewhere.
- I have not used any resources produced by anyone else.
- I did all the work myself, or with my allocated group, and have not helped anyone else.
- The material in the report is genuine, and I have included all my data/code/designs.
- I have not submitted any part of this work for another assessment.
- My work involved medical data, such as CT Scans. ERGO II, for Secondary Data Analysis was granted.

Acknowledgements

I would like to express my deepest gratitude to Dr. Mercedes Arguello Casteleiro, whose expertise, understanding, and patience, added considerably to my experience, and whose guidance supported me throughout my research and writing.

I would like to extend my heartfelt thanks to my friends and family, who provided unwavering support throughout this journey. Their encouragement during challenging times was invaluable.

Contents

1 Introduction

The use of Artificial Intelligence (AI) in diagnosing Computed Tomography (CT) scans has marked a significant shift in the medical industry's approach to diagnosing health issues [1]. Traditionally dependent on professionals' expertise, the field increasingly relies on AI for its quick nature of spotting conditions, enabling earlier intervention [2]. This trend is highlighted in a systematic review on AI in disease diagnosis, which provides an overview of AI applications in medical diagnostics [3]. Integrating AI in healthcare highlights critical challenges, especially for CT scan analysis. The complexity inherent in AI programming necessitates expertise often beyond many medical professionals' capabilities, constraining broader utilisation and the potential benefits of AI in medicine. This issue underscores the pressing need to bridge the gap between AI technical demands and medical practitioners' skills to leverage AI's advantages in healthcare fully [4].

The potential impact of making AI more available for medical diagnosis is profound. It could revolutionise the medical industry as we know it today and enhance patient care. When medical professionals' expertise is scarce or a second opinion is required, an AI system can lead to precise, timely diagnoses [2].

This project introduces a novel approach to address the technical complexity of developing AI models for CT scan analysis. Unlike traditional approaches that demand high coding expertise, this project pioneers a streamlined, user-friendly framework. This innovative solution significantly lowers the barrier for medical professionals with limited coding experience. This solution facilitates the creation and implementation of AI models in an accessible manner. Doing so paves the way for the democratisation of AI in medical diagnostics. Making it a more inclusive tool that enhances diagnostic capabilities without requiring extensive technical training [4].

The cornerstone of this project is its focus on simplicity. Through this innovative framework, we are setting a new paradigm in AI development for healthcare, enabling a broader spectrum of medical practitioners to benefit from advanced AI diagnostics.

However, simplifying the AI code without compromising its accuracy and reliability is difficult. This project proposes a novel approach, drawing inspiration from user-friendly AI models akin to those developed by Hugging Face and the efficient image classification methods utilised by LoRa. The primary aim is to refine the AI system to a coding simplicity consistent with its accuracy and reliability for medical diagnostic purposes. This involves reengineering the network's coding framework and integrating comprehensive evaluation methods to ascertain the system's effectiveness and reliability.

Acknowledging the challenges and limitations of this endeavour is essential. Striking the delicate balance between simplicity and sophistication in AI systems is a notable challenge. Additionally, our reliance on existing data and algorithms may introduce unavoidable biases or constraints, potentially impacting the system's overall performance and applicability in diverse clinical settings. These are areas that require careful consideration and ongoing research.

This project sets out to make significant strides towards enhancing the accessibility of AI in medical diagnostics. By making AI systems more accessible and user-friendly for medical professionals, this project aims to broaden the utilisation of AI in healthcare, thus contributing to advancing medical diagnostic practices.

Summary of Contributions

- Development of an AI system for CT scan analysis that combines ease of use with high diagnostic accuracy.
- Implement a novel coding framework to make AI more user-friendly, minising the technical expertise.
- Comprehensive evaluation of the system's effectiveness for different datasets.
- Identifying the balance between system simplicity and functionality, with insights into overcoming common limitations in AI diagnostics.
- Analyse models' performances with varying dataset sizes, demonstrating the AI system's capability to maintain high accuracy and reliability with large and small datasets.

2 Literature Review

2.1 Overview

Recently, AI has become one of the fastest-growing technologies, revolutionising various industries with its capability of handling large amounts of data [5]. AI's ability to process and analyse large amounts of data benefits the medical sector. The demand for healthcare services is ever-increasing, and medical practitioners need help to keep up with the increased workload. In addition, due to the complexity of medical images and their extensive variations, it may be challenging to give an efficient and accurate diagnosis [6]. Hence, integrating AI into the medical sector offers a promising solution to this challenge, potentially relieving the burden on medical professionals [7].

AIs can also be used for image classification. AI algorithms can swiftly analyse CT scans, identifying patterns and anomalies the human eye might miss [8]. This capability enhances diagnostic accuracy and significantly speeds up the process, allowing healthcare providers to address patient needs more promptly and effectively [9]. Due to this, medical professionals can focus on more complex and urgent cases where human expertise and judgment are essential. This shift in focus could lead to improved patient care, as doctors and nurses can dedicate more time to patient interaction and decision-making based on insights provided by AI [7].

However, developing an AI to aid with image classification has its limitations, mainly regarding its complexity. Healthcare professionals in resource-poor countries may need more training to operate the AI. Hence, there is an increasing focus on crafting AI systems that demand the most minor technical know-how from users. The deployment of such AI would be crucial, as it would facilitate broader adoption among medical professionals. Therefore, this approach aims to bridge the gap between advanced AI capabilities and practical usability in healthcare settings, making AI a more accessible and effective tool for medical professionals [10].

2.1.1 Simplification Through High-Level Abstractions

Significantly reducing the coding complexity in developing machine learning models has led to innovative approaches. That allows for creating these models with minimal coding—potentially as few as three lines [11]. This breakthrough is instrumental in making machine-learning technologies accessible to a broader range of professionals.

High-level abstractions in programming libraries and frameworks have been pivotal in this development. Specific libraries have introduced functions and methods that abstract away the complexities of model architecture, training, and evaluation processes. This allows for deploying sophisticated machine learning models with minimal input lines of code, drastically lowering the barrier to entry for utilising AI in diagnostic applications [11].

The significance of this simplification cannot be overstated, particularly in healthcare diagnostics. It enables medical professionals to leverage AI's power without requiring extensive technical training. This democratises the use of AI in medicine and opens up opportunities for more personalised and accurate diagnostics. Practitioners can more readily apply machine learning models to patient data, enhancing decision-making with AI-driven insights [8]. Moreover, the approach to partially or fully automate aspects of the machine learning workflow, such as data preprocessing and model tuning, emphasises augmented machine learning. This moves beyond merely automating model architecture search, focusing instead on facilitating other steps in the machine learning workflow to accommodate users with varying levels of expertise [11].

These principles underscore the potential for simplified coding frameworks to transform the application of AI in healthcare, making it more accessible and effective for medical professionals. Reducing the coding required to just a few lines bridges the gap between advanced AI capabilities and their practical usability in healthcare settings. This may enhance healthcare delivery through accessible and effective AI solutions [11].

The project aims to create an AI system that excels in analysing CT scans and is accessible and manageable for medical professionals. Drawing inspiration from platforms like Hugging Face, which have simplified the use of AI, the project seeks to streamline the diagnostic process, making it more efficient without compromising accuracy or effectiveness. This literature review will delve into the use of AI for medical imaging, focusing on the balance between simplifying AI code and maintaining its efficacy, aligning with the overarching goal of enhancing healthcare delivery.

2.2 AI in Medical Imaging and Diagnostics

A study was conducted to see how impactful AI is when diagnosing CT scans compared to radiologists. It involved a retrospective analysis of CT scans from 309 participants, encompassing 360 pulmonary nodules. These CT images were reviewed by radiologists using AI technology, in which they compared the diagnostic accuracy of AI with that of human radiologists [12].

One of the key findings of this study is the favourable accuracy rate of AI in diagnosing lung cancer from CT scans. This highlights AI's potential to enhance diagnostic accuracy in medical imaging, particularly in identifying complex conditions such as cancer. Furthermore, in this study, it was clear that AI required significantly less time for film reading, as it took (195.4±65.2 s) on average, compared to manual examination by human radiologists, which took (581.1±116.8 s). This aspect underscores AI's ability to increase the speed and efficiency of the diagnostic process. Rapid analysis of medical images is crucial in clinical settings, where timely diagnosis can directly impact patient treatment and outcomes [12].

This study shows that AI shows promising results in enhancing diagnostic accuracy, speed, and efficiency in medical imaging [12].

2.3 Simplifying AI Code for Medical Use

Simplifying the AI code is crucial in enhancing the accessibility and usability of advanced technologies for medical professionals. One way of doing this is through the use of transformer models. A study provides a comprehensive benchmarking of transformers against traditional convolutional neural networks (CNNs) in medical image classification, a key area in medical diagnostics. This research is pivotal in understanding how transformers can be optimised for medical imaging, addressing the challenge of simplifying AI code [13].

Typically known for their success in natural language processing, transformers have recently gained popularity in the computer vision community, including medical imaging. Platforms like Hugging Face have pioneered transformers, demonstrating their effectiveness in various applications. The study highlights that transformers can outperform CNNs in various medical classification tasks, suggesting a potential shift in the AI models used for medical imaging [13].

One of the critical advantages of transformers is their ability to handle large-scale data efficiently, which is crucial in medical imaging, where datasets can be vast and complex. This study also addresses the data-hungry nature of the transformers. By utilising unlabeled in-domain data, the researchers present a method to bridge the domain gap between photographic and medical images. This approach is particularly relevant in simplifying AI code, as it reduces the need for extensive and often labour-intensive data labelling, making AI models more accessible and easier to train [13].

2.4 Maintaining Efficacy in Simplified AI Systems

Building on transformers, just simplifying the code, would not be enough; we need to ensure that efficiency is not affected either. Another study provides valuable insights into this area, focusing on applying the Swin-Transformers for medical image quality assessment. This study is beneficial as it illustrates how to code simplification could occur without compromising its diagnostic accuracy [14].

This study trained the Swin-Transformers models to classify "X-Ray images with foreign objects and Cardiac MRI images with a 4-chamber view with Left Ventricular Outflow Tract (LVOT)". The study's application of these transformers to Chest X-rays and Cardiac MRI datasets highlights their potential to enhance the accuracy of medical imaging analysis. The results indicate that the Swin-Transformer not only simplifies the AI code but also maintains, and in some cases improves, the quality of image classification [14].

This research was significant as it shows that maintaining the efficacy of AI systems while simplifying their code is possible. The ability of transformers to handle complex medical imaging tasks with improved accuracy suggests that they can be a viable solution for simplifying AI code in healthcare applications.

This aligns with the goal of enhancing healthcare delivery through accessible and effective AI solutions. This ensures that medical professionals can leverage the benefits of AI without the need for extensive technical expertise [14].

2.5 Alternative Approach

2.5.1 Vision Transformers

There are other approaches to delivering a simplified AI code whilst upholding the integrity of the diagnostic ability; one way is using ViT. A recent study compared ViT and CNN [15].

ViTs have gained attention for their self-attention mechanism. This allows for comprehensive image analysis in patches. This approach is suitable for handling high-resolution images and smaller datasets. ViTs have shown promise, outperforming CNNs due to their ability to process entire images and "increase of relations created between images through the self-attention mechanism" [15].

On the other hand, CNN is a more traditional approach with robust local feature extraction capabilities, which are the standard in image processing. They excel in tasks requiring local context understanding, like object detection. However, CNNs may need help with global contextual information, a strength of ViTs. ViTs and CNNs offer unique advantages in image processing [15].

3 Materials and Methods

This section outlines how this project approaches the problem of AI code being too complex for professionals to utilise. This project aims to develop an advanced, simple AI code for classifying medical conditions from CT scans. The novelty of this approach lies in adapting successful technologies from other fields, such as Natural Language Processing (NLP), to the medical imaging context. Mainly to see how these technologies diagnose medical conditions through CT scans. This section outlines how we approach the problem of AI code being too complex for professionals to utilise.

This methodology strongly emphasises data preprocessing and leverages high-level programming abstractions that simplify development. Instead of requiring users to design and configure neural network architectures from scratch, our approach involves using pre-configured models that can be easily selected and deployed with minimal code.

For example, users can initialise and train a model by specifying the model name and number of classes in a single line of code. This streamlined command encapsulates complex processes, including model selection, configuration, training, and evaluation. It demonstrates this project's commitment to reducing the technical barriers to deploying sophisticated AI models. By abstracting the underlying complexities, we try to enable medical professionals who may need more extensive coding expertise to apply advanced machine learning techniques effectively in their diagnostic work.

This project aims to democratise access to powerful AI technologies in medical diagnostics. It aims to ensure that medical practitioners can leverage these tools to enhance efficiency without delving into the complexities of model architecture and coding.

Ultimately, this project also seeks to provide a clear, replicable, and detailed account of a streamlined AI deployment in medical imaging. This project's commitment extends to ensuring the reliability and accuracy of the findings, setting a new standard for ease of use in medical AI applications, and potentially transforming diagnostic practices in healthcare settings.

3.1 Ethical Approval

Given the involvement of medical datasets, mainly CT scans from individuals, obtaining ethical approval was essential in ensuring the project adhered to research integrity and data protection standards.

The datasets used in this study were sourced from Kaggle. These datasets are anonymised and, therefore, contain no personal information, thereby upholding the privacy of the data subjects.

The datasets chosen have specific licenses that permit their use for academic and non-commercial purposes. The licenses were thoroughly reviewed to guarantee that the project's use of the data adheres to these conditions, mitigating legal and ethical risks.

The ethical approval was successful, underlining the project's commitment to ethical research practices. This approval is a testament to the project's design, which respects ethical standards and data protection laws.

Aspect	**Description**
Ethical Approval	Applied and received for secondary data analysis, covering specified datasets.
Data Source	Multiple datasets from Kaggle, including those for Osteoporosis, Cervical Spine Fractures, Lung Cancer, and Liver Tumour.
Data Licensing	Compliant with licenses from Kaggle and other sources, permitting non-commercial use, academic research, and education.
Data Privacy	Datasets are anonymised, aligning with GDPR and minimising identification risks.
Anonymity in Results	Focus on evaluation metrics ensures minimal risk of individual identification.
Other Ethical Considerations	No additional ethical risks anticipated due to the study's focus on computational methods.
Public Domain Data	Data is publicly available on Kaggle and in the public domain.

Table 3.1: Ethical Considerations

3.2 Datasets

This project utilises a series of CT scans sourced from Kaggle, a platform commonly used for data science competitions. However, it also is recognised for its diverse and extensive data repositories. The datasets were explicitly chosen according to their relevance to the project's aim of classifying various medical conditions. There are multiple datasets. This ranges over the broad spectrum of health conditions such as osteoporosis, cervical spine fractures, lung cancer, and liver tumours. Each dataset comprises many CT scans, ensuring a robust and comprehensive analysis. The size of these datasets varies, providing a diverse range of data in terms of both image quantity and medical condition complexity. Below is a detailed overview of each dataset used in the this research.

Dataset	Description	Volume
Osteoporosis Knee X-ray	It consists of knee scans categorising three conditions: normal, osteopenia, and osteoporosis.	Contains 176 images: 24 normal, 110 osteopenia, and 42 osteoporosis. Dataset Split: Training: 153 Evaluation: 23
Lung Cancer	It encompasses a comprehensive set of CT scans from patients at various stages of lung cancer and healthy subjects. It's made to assist in differentiating between benign, malignant, and normal lung conditions.	Consists 1097 images: 416 normal, 120 benign, and 561 malignant. Dataset Split: Training: 877 Evaluation: 220

Table 3.2: Summary of Datasets Used in the Study

An 80-20 training-to-evaluation split was implemented. This standard strategy in machine learning ensures that models are neither overfitted nor undertrained. This proportion maintains a balance between model learning and validation capabilities.

As shown in Table 3.2, the dataset sizes selected for this study are tailored to demonstrate the AI system's adaptability and performance across different data scales. With only 23 images, to evaluate the model, and 176 total images, the Osteoporosis dataset represents smaller datasets. Smaller datasets were used to demonstrate the models' performances on specialised medical conditions where acquiring data can be challenging. This dataset evaluates how well the AI model performs under data-constrained scenarios, reflecting real-world situations where only limited data might be available.

The Lung Cancer dataset, on the other hand, is a testament to the AI model's ability to handle and learn from more extensive data volumes. Comprising 220 images, to evaluate the model, and 1097 total images, it demonstrates larger datasets often available for more common or well-documented conditions. This dataset is a powerful tool for assessing the AI model's robustness and scalability and showcasing the system's modularity.

Various dataset sizes illustrate the models' capability of maintaining high performance and reliability, whether deployed in large-scale or more restricted clinical settings.

The strategic choice of diverse dataset sizes is a cornerstone of this study. It supports a comprehensive evaluation of the AI system, ensuring its readiness for various medical diagnostic applications. The AI system is designed to adapt and perform effectively across different data scales, from those with sparse data availability to those benefiting from larger datasets.

3.3 Deep Learning Models

This project uses four deep learning models to explore the diagnostic capabilities of artificial intelligence. Each brings unique strengths to the medical image analysis of CT scans. Each model was chosen for its specific ability to handle different aspects of image processing and analysis, from feature extraction to complex pattern recognition. The models implemented include:

- **Traditional CNN Model:** A staple in image processing, known for its effectiveness in detecting spatial hierarchies in images through multiple layers of convolutions.
- **Swin-Transformer:** This architecture leverages shifted windows for self-attention, enhancing the model's capacity to focus on relevant features within CT scans.
- **Hugging Face ViT:** This technique uses transformers, which traditionally excel in NLP, adapted here for image analysis, highlighting their versatility and powerful representational capabilities.
- **Custom Neural Network Model:** This model integrates neural networks' robust feature extraction capabilities, followed by a reasoning layer that combines these features for accurate classification in medical diagnostics.

Each model's architecture is tailored to the nuances of medical imaging, aiming to optimise the accuracy and efficiency of diagnosing conditions through CT scans. The following sections delve into how these models are structured and functioned.

3.3.1 CNN

CNN is a foundational model in deep learning, especially for processing images. Its architecture is designed to automatically and adaptively learn spatial hierarchies of features through processing layers. Each layer in a CNN transforms one volume of activations to another through a differentiable function. The CNN model's layer composition consists of convolutional layers, activation functions, pooling layers, fully connected layers, and output, as shown in Figure 3.1.

At the core of the CNN model are the convolutional layers. The first layer applies 64 filters of size 3x3 to the input image. This layer captures low-level features such as edges and simple textures. The following two layers increase in complexity and depth, using 128 and 256 filters, respectively. Each convolutional layer uses a kernel that processes the image in small blocks, capturing local feature information without losing spatial relationships within the image. The stride, set to 1, ensures the filters slide over the image in a one-pixel step, providing a dense feature map output.

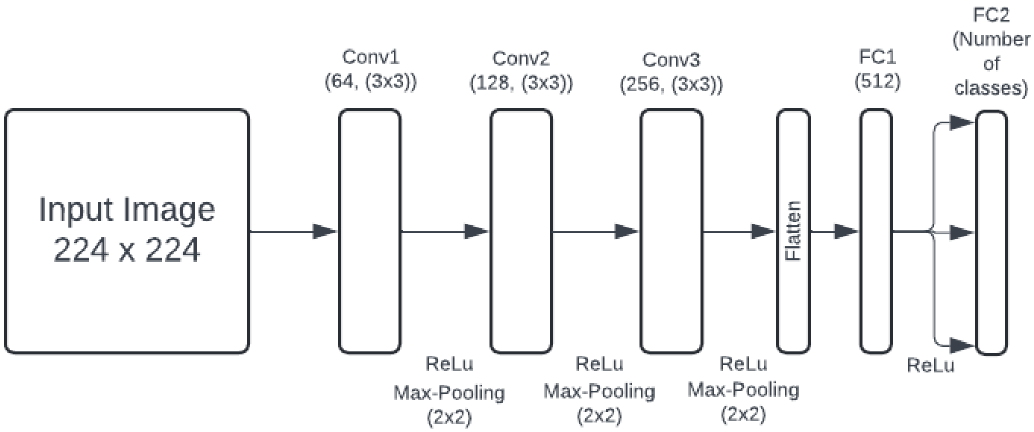


Figure 3.1: CNN Model

After that, an activation function, particularly the Rectified Linear Unit (ReLU), is used. This step is crucial as it allows the network to learn more complex patterns. ReLU is used due to its computational efficiency.

A pooling layer (MaxPool2d) is used following the convolutional and activation operations. This layer performs a down-sampling operation along the spatial dimensions, reducing the number of parameters and computations in the network. It helps detect consistent features to scale and orientation changes. In this model, max pooling is used, which extracts sharp features like edges by retaining the maximum value of the portion of the image covered by the kernel.

After several convolutional and pooling layers, the model transitions to fully connected layers. These are traditional neural network layers where every input is connected to every output by a learnable weight. Here, the first fully connected layer takes the flattened output from the last pooling layer and projects it onto a 512-dimensional space. The final fully connected layer maps these 512 features to the number of classes specified by the user. This layer is crucial for classification, combining all the learned features to predict the most likely class label.

3.3.2 Hugging Face ViT

Traditionally used in natural language processing, the principles of the transformer architecture have been adapted to make a transformer model, ViT, analyse images by Hugging Faces. This section delves into the mechanics of ViT, outlining how it adapts the transformer mechanism for visual data processing.

Unlike CNNs, ViTs treat images as sequences of patches, effectively transforming 2D images into a 1D sequence of embeddings. This adaptation allows the model to use transformers' powerful self-attention mechanism for image classification tasks. An example of a ViT is shown in Figure 3.2.

Image x, with dimensions $H \times W \times C$ (height, width, channels), is split into N flattened 2D patches x_p, each $P \times P \times C$, where $P \times P$ is the size of the new image. Then, it is linearly projected into a constant latent vector size D, a process similar to embedding tokens in NLP. This transformation is essential as it converts raw

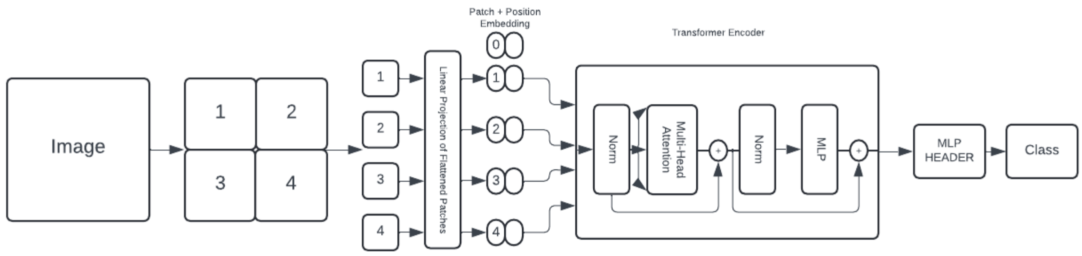


Figure 3.2: ViT Model inspired from Hugging Face Model [16]

pixel data into a form the transformer encoder can process while maintaining spatial hierarchies [16].

Afterwards, position embeddings are added to the patch embeddings to retain positional information within the image. Unlike some models that use complex 2D-aware positional embeddings, ViT employs standard learnable 1D position embeddings, which have shown sufficient effectiveness [16].

Then, this is fed into a Transformer encoder. This encoder consists of multiple layers of multi-headed self-attention (MSA) and multilayer perceptrons (MLPs), where each block is preceded by layer normalisation (LN) and followed by residual connections. This setup allows the model to attend to different parts of the image and integrate information across the entire field of view, enabling global context understanding from the outset [16].

Within the encoder, the classification token, a concept borrowed from BERT, aggregates information across the transformer layers [16].

The strategic use of the classification token and the tailored MLP header reflects the Vision Transformer's (ViT) adaptability and proficiency in image classification tasks. This framework leverages the global reach of self-attention mechanisms. While effectively learning and preserving spatial relationships inherent in image data. ViTs present a powerful paradigm shift from traditional CNN architectures, relying less on inductive biases and more on the model's intrinsic ability to interpret spatial hierarchies and correlations [16].

3.3.3 Swin-Transformers

Swin Transformers represent an innovative shift in applying Transformer models to classify images. They introduce a hierarchical Transformer whose representation is computed with shifted windows. This approach maintains the efficiency of self-attention computation within local windows yet enables cross-window connectivity for comprehensive feature integration [17].

The Swin Transformer architecture begins by partitioning images into non-overlapping patches, similar to what ViTs do. However, Swin Transformers then organise these patches into a hierarchical structure that allows for processing at various scales and complexities. This approach provides a more efficient way to handle high-resolution images compared to the standard Transformer. Several Transformer blocks follow this initial stage, modifying self-attention computation to accommodate the specific needs of image processing [17].

Afterwards, Swin Transformers employ a patch merging strategy to reduce the number of tokens and increase the feature dimension. Therefore, constructing a hierarchical representation of features. This hierarchical design is crucial for capturing features at different scales. It is also particularly effective for vision tasks that require understanding fine and coarse details.

An essential design element of the Swin Transformer is the shifted window strategy for self-attention. Initially, regular window partitioning is used, and then the windows are shifted to the subsequent layers. This shift allows for cross-window interaction and a more concentrated representation, as it prevents self-attention isolation within individual windows [17].

3.3.4 Custom Neural Network

As previously mentioned, the Custom Neural Network model starts with a Swin Transformer, which can extract a hierarchy of features from images. These features are crucial for understanding the intricate patterns, where precise details can indicate different conditions.

After that, the model employs an adaptive average pooling step to condense the spatial dimensions of the features. This step is essential for reducing the data's complexity while preserving the features' most informative parts. This is crucial for the subsequent reasoning process.

The core of the Custom Neural Network model is its reasoning layer, a series of fully connected layers that process the pooled features. This layer consists of a series of fully connected layers that transform the pooled features into a format suitable for classification. Each layer in this segment has learnable parameters that adjust through training. This enhances the model's ability to make accurate predictions based on the extracted features.

Finally, the output from the reasoning layer forms the basis for decision-making or classification tasks. In a medical imaging context, this would mean diagnosing various conditions based on CT scans.

3.4 Computional and Experimental Set-Up

This project uses a MacBook with an Apple M2 Pro processor and 32 GB RAM. The MacBook features an integrated GPU that significantly enhances its computational capabilities. While the Apple M2 Pro's GPU excels at parallel processing tasks, not all machine learning libraries currently support its architecture, which can pose challenges for model development.

Due to limited support from some machine learning libraries, the integrated GPU was not utilised for computational tasks. Hence, the CPU was used, effectively processing two of the originally planned four datasets. Due to impractically long training times, the cervical spine fractures and histopathologic cancer datasets were excluded from the study. Access to Iridis 5, a high-performance computing system at the University of Southampton, was sought to overcome these computational constraints.

3.4.1 Iridis 5

Iridis 5 was created as part of the University of Southampton's ongoing commitment to integrating high-performance computing into its research infrastructure. As one of the top 500 supercomputers globally, Iridis 5 significantly enhances computational capabilities over its predecessors.

Despite these capabilities, challenges with remote access via VPN led to frequent disconnections, making consistent use of Iridis 5 difficult. This connection instability necessitated reliance on the MacBook's CPU resources.

Benefits of Localised Computing Resources

On a positive note, due to these constraints, this project demonstrates that sophisticated AI models can be run on commonly available hardware. This success is significant as it aligns directly with the project's goal to develop an AI system that is not only accurate but also user-friendly and accessible to medical professionals.

The project also highlights the AI system's flexibility by proving that these models can be run on standard laptops depending on the dataset size. Therefore, it demonstrates that AI is accessible and feasible for everyday clinical use where such resources may be unavailable. Medical professionals can use these tools directly on personal or hospital computers. This ease of integration can accelerate the adoption of AI technologies in clinical settings, where quick and accurate diagnostics are crucial.

3.4.2 Tools and Technologies Used

This project employed various modern tools and technologies to facilitate the development and implementation of AI models for CT scan analysis. This section outlines the software used, highlighting how each part contributed to the project's success and aligned to make sophisticated AI accessible to medical professionals.

Python was the project's foundation due to its extensive support within the data science and machine learning communities. Its ecosystem includes numerous libraries tailored explicitly for data manipulation, machine learning, and deep learning, making it an ideal choice for developing complex AI models.

Several specific libraries were instrumental in this project:

- **PyTorch:** This library was central to building and training the neural network models. PyTorch allows for intuitive model development, which is critical for handling complex architectures.
- **Transformers:** This library incorporated advanced neural network architectures such as Vision Transformers. It provided access to pre-trained

models that could be fine-tuned to our specific dataset, significantly reducing development time and computational resources.

- **Timm:** This PyTorch image models library enabled quick access to cutting-edge model architectures and their pre-trained weights. This allowed the project to take advantage of the latest advances in deep learning for image classification.

The project developed and tested the models using Jupyter Notebook. Jupyter Notebook offers an interactive coding environment with visualisation and feedback, which is particularly beneficial for iterative testing and model tuning.

Careful selection of programming languages, libraries, and tools was pivotal in developing an accurate AI system for medical image analysis. These tools ensured the project was built on a solid and advanced technological foundation capable of supporting its ambitious goals.

3.5 Model Implementation and Framework Design

This project showcases a streamlined approach to creating AI systems for medical diagnostics. Complex model training and deployment processes are encapsulated and simplified using concise code. This technique employs high-level programming abstractions, enabling models to be quickly developed, tested, and deployed. It simplifies technology for medical professionals with limited programming experience, reducing development time and errors.

At the heart of this framework is the ability to perform significant tasks, such as data preparation, model training, and prediction, using straightforward commands. This design philosophy ensures that users can focus on model applications rather than the underlying technical complexities. For example, initialising and training a model, preparing datasets, and classifying new data points can be accomplished through brief and intuitive code snippets. This simplicity makes advanced AI tools practical for everyday clinical use. It aligns perfectly with our goal of enhancing diagnostic processes with AI without adding to the existing burdens of healthcare professionals.

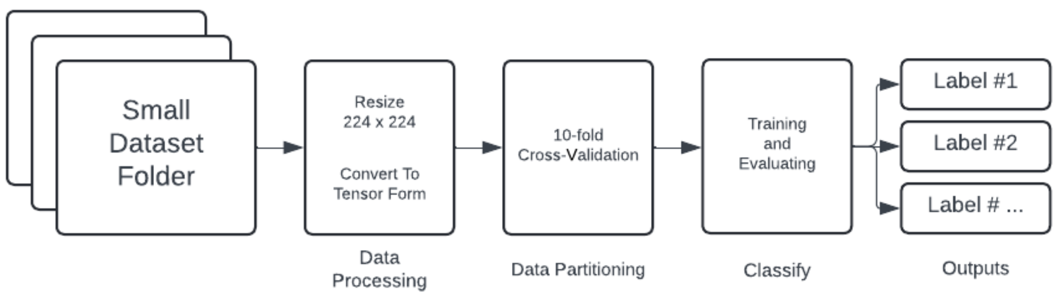


Figure 3.3: Suggested Framework for when the dataset has less than 1000 images.

Figures 3.3 and 3.4 present the suggested implementation for AI models, depending on the size of the dataset. The framework's implementation consists

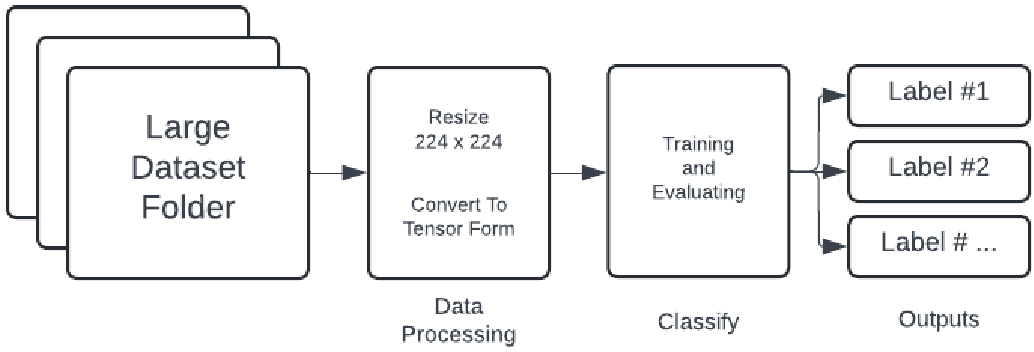


Figure 3.4: Suggested Framework for when the dataset has 1000 or more images.

of three main stages: data processing, model training and evaluation, and classification. This approach provides a clear and efficient pathway from raw data to practical insights.

3.5.1 Concept to Design

Each model was implemented and examined in isolation during the initial stages of developing the AI system. This step was crucial to determining each model's capabilities and operational strengths and identifying any constraints that could affect the design of a cohesive system. Table 3.3 outlines the approximate lines of code required to implement each model, highlighting the technical effort involved in their development:

Model	**Lines of Code**
CNN	225
Swin Transformer	110
ViT	187
Custom Neural Network	159

Table 3.3: Lines of code required to implement each model, demonstrating the relative complexity and coding effort involved.

After conducting tests on each model, significant similarities and differences were discovered, which helped understand the functionalities that could be standardised and those that required customisation. Certain aspects, such as data preprocessing and evaluation protocols, could be streamlined across all models to promote consistency and reduce complexity. At the same time, the unique characteristics of each model highlighted the need for a flexible architecture within the system - one that could adapt to the distinct requirements of different algorithms without compromising them.

The insights gained from the comparative analysis informed the final design of the AI system, as shown in Figure 3.5. The result is an integrated framework simplifying the transition between different end-user models. The system

```
dataset, class_names = prepare_dataset('Osteoporosis Knee X-ray')
model = train_and_evaluate({"model_name": "cnn", "num_classes": 3}, dataset, {})
classify('Osteoporosis Knee X-ray/osteoporosis/OS49.jpg', model, class_names)
```

Figure 3.5: An example of the final three lines of code necessary to prepare the dataset, train and evaluate the model, and classify the images.

architecture was refined to abstract model-specific operations' complexities, ensuring ease of use. This design philosophy ensures the final implementation aligns with the project's goal, enabling practitioners to leverage AI for accurate CT scan diagnostics without coding expertise.

3.5.2 Data Processing

During the data processing stage, the focus is streamlining medical professionals' workflow. The system is designed to handle image data stored in a structured manner effortlessly. The approach categorises CT scan images into folders for distinct medical conditions. This simple organisational step is the extent of their manual involvement in the data preparation process. The system takes over from there, automating the rest of the preparation sequence.

This command instructs the system to look into the specified directory. Then, it searches for this path, expecting to find a folder containing images organised into subfolders representing different classifications.

Upon locating the directory, they are seamlessly processed. The process standardises the images by resizing them to a uniform dimension, ensuring consistency. It then converts the images into a tensor format, a requirement for deep learning models. Finally, each image is associated with its corresponding class label based on the subfolder in which it resides.

After the dataset preparation process, a dataset and a set of class names are generated. The class names guide the model's output, linking the numerical predictions to the human-readable diagnoses, making it easier for medical professionals to interpret the results.

This transformation occurs through a series of automated steps that abstract away the complexities of data manipulation. The system recognises the organisation of the image folders, identifying and labelling the various classes based on the folder structure.

This data processing method is designed to be as unobtrusive as possible for the end user. It requires minimal technical know-how while guaranteeing that the data fed into the model is optimally prepped for high-performance outcomes. This setup underscores the commitment to creating a user-friendly AI tool that respects medical professionals' time and technical expertise.

3.5.3 Training and Evalutating

Integrating the training and evaluation processes into a unified framework has advantages, particularly in simplifying the development and application of AI models. This integration ensures that the model aligns with the training objectives, moving from optimisation to performance assessment. Maintaining consistency in the metrics and goals across both stages is crucial to achieving reliable and predictable model behaviour.

Moreover, this approach enhances efficiency by streamlining the model development process. Merging training and evaluation reduces the need for repetitive data handling and processing, thus accelerating iterations and adjustments. This is particularly beneficial where swift decision-making based on accurate model predictions is imperative.

The simplified process enhances the user experience, especially for medical professionals needing extensive knowledge of AI technology. It allows them to concentrate more on the AI model's outcomes and usefulness than its underlying complexities. This ease of use is crucial in promoting the use of AI technology in healthcare environments where user-friendliness and dependability are critical.

Lastly, having evaluation as a continuous part of the training process serves as an ongoing quality control mechanism. It provides real-time insights into the model's learning progress and generalisation ability. This will help identify and address issues such as overfitting early on. This proactive approach to model validation ensures that the AI system remains robust and reliable throughout its deployment.

The Process

Training and evaluating machine learning models for medical image analysis involves a thorough setup to ensure the model's accuracy and robustness. At first, the user has to pick one of the four models. This foundational step-up allows experimentation with different models to determine the best fit for specific diagnostic tasks.

The computational resources are evaluated based on availability to configure the model to use a GPU or CPU. The system's adaptability allows it to be suited for different hardware configurations, thus increasing its accessibility. Afterwards, the dataset is split into training and validation sets to avoid overfitting and guarantee that the model can properly generalise to new data.

The process of training a model entails going through multiple epochs. Each epoch represents a complete cycle of the training data and involves updating the model weights to minimise loss. This iterative process is crucial for refining the model's accuracy.

```python
custom_config = {
    "split_ratio": 0.7,
    "batch_size": 16,
    "epochs": 5,
    "lr": 5e-5,
    "k_folds": None,
}

dataset, class_names = prepare_dataset('lung cancer/The IQ-OTHNCCD lung cancer dataset/The IQ-OTHNCCD lung cancer dataset')
model = train_and_evaluate({"model_name": "cnn", "num_classes": 3}, dataset, custom_config)
classify('lung cancer/The IQ-OTHNCCD lung cancer dataset/The IQ-OTHNCCD lung cancer dataset/Normal cases/Normal case (11).jpg', model, class_names)
```

Figure 3.6: An example of how users with more coding expertise can modify the parameters.

The framework for training and evaluation is intentionally built to be flexible and meet the needs of users with different levels of expertise. For instance, users with experience can customise the parameters such as split ratio, batch size, number of epochs, learning rate, optimiser type, and k-folds in cross-validation to tailor the model according to their specific requirements. This is shown in Figure 3.6.

The **split ratio** parameter determines how the dataset is split into training and validation sets. By default, 80% of the data is allocated for training the model, while the remaining 20% is kept aside for validation. This ratio is chosen strategically to ensure that the model has enough data for learning and a distinct subset for performance evaluation. Balancing these sets helps prevent overfitting and ensures the model can generalise well to new data.

The **batch size** specifies the number of examples the model processes before it updates its weights. The default setting is 32, which strikes a balance between computational efficiency and the stability of the training process. A moderate batch size helps smooth out the learning updates, reducing the noise in the gradient estimates, which can be more pronounced with smaller batch sizes.

The default configuration sets the number of **epochs** at 15. An epoch represents one complete pass through the entire training dataset, and this setting is crucial in determining how long the model will train.

The **learning rate** is critical to how quickly a model adjusts its weights to minimise the loss. The default setting is 1e-4, which controls the step size at each iteration while moving toward a minimum loss function. A carefully set learning rate ensures the model converges to an optimal solution efficiently without overshooting or getting stuck in local minima.

The 'Adam' **optimiser** is used by default. It is known for its effectiveness in various types of neural networks. Adam calculates adaptive learning rates for each parameter to handle sparse gradients on noisy problems, making it a robust choice for training deep learning models.

The **k-fold cross-validation** method is valuable for evaluating model performance, especially when data is limited, as shown in Figure 3.3. This approach divides the dataset into k subsets, or folds, using k-1 for training and the remaining fold for testing. This process is repeated k times, with each fold being the test set once. The default setup employs ten folds, ensuring thorough testing that assesses the model across various data segments.

K-fold is only used when the dataset contains less than 1000 images. This criterion is crucial for smaller datasets, such as the Osteoporosis Knee X-ray dataset, which may need to provide more data to train robust models effectively. By implementing k-fold cross-validation, each data point is utilised for training and testing, maximising the dataset's usefulness and ensuring more reliable and generalisable model performance despite the smaller data size.

Evaluation Metrics

To evaluate the model's performance, numerous metrices were used. These included accuracy, precision, recall, F1-score, ROC curves and confusion matrixs.

These metrics not only measure how well the model performs but also guide the refinement of its accuracy and reliability across various applications.

Accuracy is a crucial metric that evaluates how often the model correctly predicts an outcome. It is calculated by dividing the number of correct predictions by the total number of predictions made.

$$\text{Accuracy} = \frac{\text{True Positives} + \text{True Negatives}}{\text{Total Observations}}$$

Precision measures the accuracy of positive predictions. It calculates the ratio of true positives to the sum of true and false positives. This metric is crucial in scenarios where the consequences of false positives are significant, such as medical diagnoses. It ensures that the model's positive predictions are reliable.

$$\text{Precision} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Positives}}$$

Recall quantifies the model's ability to identify all actual positives. It is vital in fields like medicine, where failing to detect a condition could have severe implications. High recall indicates that the model captures most positive cases, minimising the chances of false negatives.

$$\text{Recall} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

As shown below, the F1-Score uses both precision and recall. The two metrics are combined into a single measure. This score provides a holistic view of the model's performance.

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

ROC and its corresponding Area Under Curve (AUC) measure the model's discriminative ability between positive and negative outcomes. The ROC curve plots the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings.

$$\text{TPR} = \frac{\text{True Positives}}{\text{True Positives} + \text{False Negatives}}$$

$$\text{FPR} = \frac{\text{False Positives}}{\text{False Positives} + \text{True Negatives}}$$

Lastly, the Confusion Matrix shows a model's performance by comparing actual and predicted classifications. It helps identify where the model is making errors and is instrumental in refining its accuracy by highlighting the types of errors.

3.5.4 Classify

The following section outlines the AI model's process for classifying medical images. The aim is to identify the type of medical condition and provide insights into the critical areas of the image that influenced the model's decision.

The classification begins by loading and processing the image to match the neural network's input requirements. This involves resizing the image and converting it into a tensor. Once correctly formatted, the image is passed into the model, which uses its trained parameters to predict its category.

The models used in this project handle outputs slightly differently:

- **CNN and Custom Neural Network Models:** iThese models output logits directly. Logits are raw prediction scores of the model outputs before applying softmax.

- **Swin Transformer and Hugging Face ViT:** Although these models also generate logits, the structure of the outputs can vary. Sometimes, the logits are wrapped within another object or structure, necessitating a check to extract them correctly for further processing.

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

- $\mathbf{z}$ is the vector of logits from the model.

- K is the number of classes.

- e is the base of the natural logarithm.

- z_i is the logit corresponding to class i.

Regardless of the model used, the logits obtained are converted into probabilities using the softmax function. The function below converts the raw logits into a probability distribution over the predicted classes, which helps determine the most likely class. The softmax function is crucial because it quantifies the model's confidence in each class prediction.

Integrated gradients are used to understand further how different parts of the image influence the model's output. This method identifies which pixels were the most significant in the decision-making process. It calculates the image's gradient to determine how each pixel value changes the model's prediction confidence value.

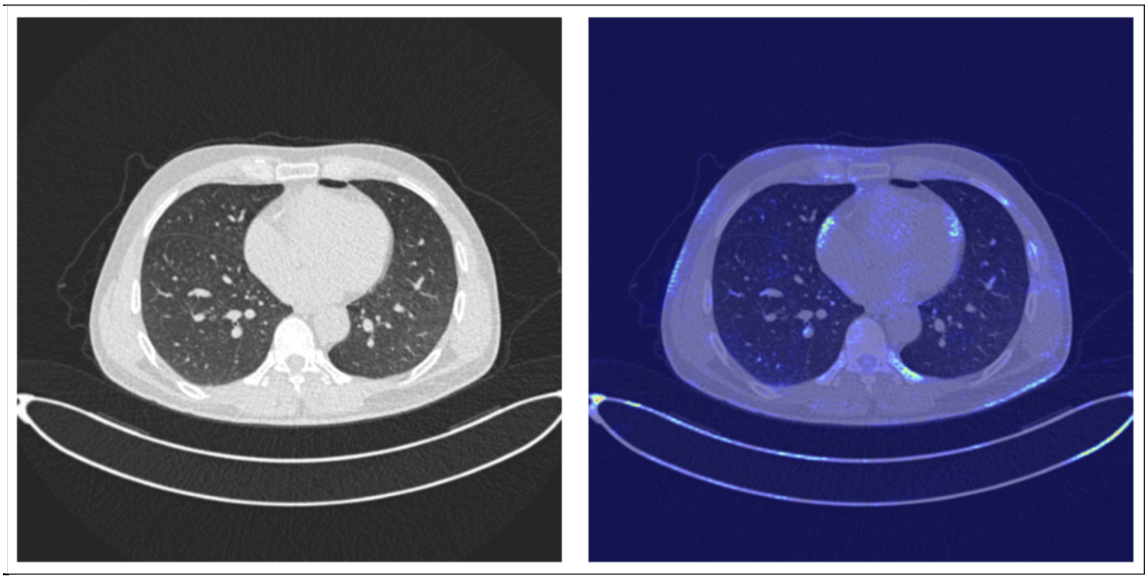


Figure 3.7: The visual result obtained after parsing an image from the Lung Cancer dataset through the CNN model

The results from the Integrated Gradients are then visualised as a heatmap. This heatmap highlights the regions of the image that significantly influenced the model's prediction, as shown in Figure 3.6. These visualisations may allow medical professionals to understand why the model made a certain decision. This can be particularly useful in medical diagnostics to validate AI-driven insights with human expertise. Therefore, AI would act as a second opinion.

The classification process concludes with the display of the original image and its heatmap overlay side by side. The model also outputs the predicted class and the confidence level of the prediction, providing clear and actionable results.

4 Results

In this chapter, we evaluate the performance of the developed models against two medical datasets: the Lung Cancer and Osteoporosis Knee X-ray datasets. The Osteoporosis Knee X-ray Dataset has a target accuracy range between 89% and 95%, and the Lung Cancer Dataset's target accuracy range is 89% to 99%. This benchmark aligns with the performance of models on Kaggle.

These targets serve as benchmarks that guide the training process. This ensures that the models are competitive with state-of-the-art solutions in medical image analysis, aiming to enhance the reliability and accuracy of diagnostics in healthcare settings.

Model	Dataset	Accuracy	Precision	Recall	F1-Score
Swin Transformer	Lung Cancer	99.55%	99.55%	99.55%	99.54%
Swin Transformer	Osteoporosis	86.96%	85.36%	86.96%	85.88%
CNN	Lung Cancer	99.55%	99.56%	99.55%	99.55%
CNN	Osteoporosis	95.65%	96.38%	95.65%	95.78%
Custom Neural Network	Lung Cancer	99.55%	99.57%	99.55%	99.55%
Custom Neural Network	Osteoporosis	95.65%	97.10%	95.65%	96.02%
Hugging Face ViT	Lung Cancer	98.18%	98.22%	98.18%	98.19%
Hugging Face ViT	Osteoporosis	82.61%	88.26%	82.61%	84.00%

Table 4.1: Model Performance on Datasets

Table 4.1 shows the performance of the AI models for medical imaging. It presents each model's accuracy, precision, recall, and F1-score across two datasets: Lung Cancer and Osteoporosis Knee X-ray. These results highlight the ability of advanced AI models to achieve high accuracy and reliability, which is crucial for clinical adoption. Most of the models have achieved remarkable results aligning with the target accuracy of the datasets. Notably, these results underline the capabilities of advanced AI models to achieve high accuracy and reliability, which is may be essential for clinical adoption.

The performance metrics shown in Table 4.1 also represent the streamlined and efficient coding approach shown in Appendix B. The contrast between the coding demands detailed in Table 3.3 and the code structure in Appendix B illustrates a significant reduction in complexity.

By simplifying the models to fewer lines of code, we maintain their impressive accuracy while vastly improving the system's usability. This coding framework aims to bridge the gap between complex AI algorithms and practical, user-friendly applications. The minimised coding effort may enable healthcare professionals to leverage advanced AI tools without being hindered by technical constraints.

4.1 Performance Analysis of Swin Transformer

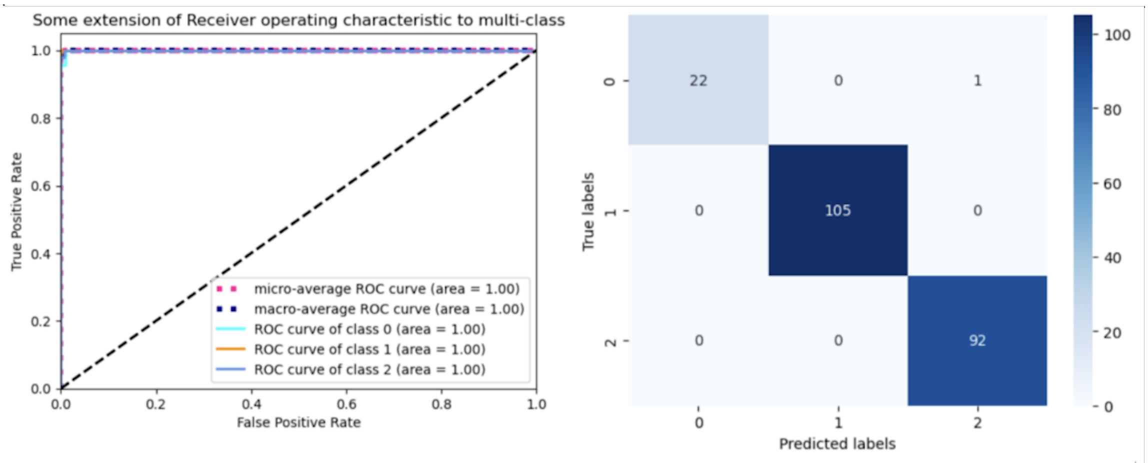


Figure 4.1: ROC curve and confusion matrix for the Swin Transformer model evaluated on the Lung Cancer dataset.

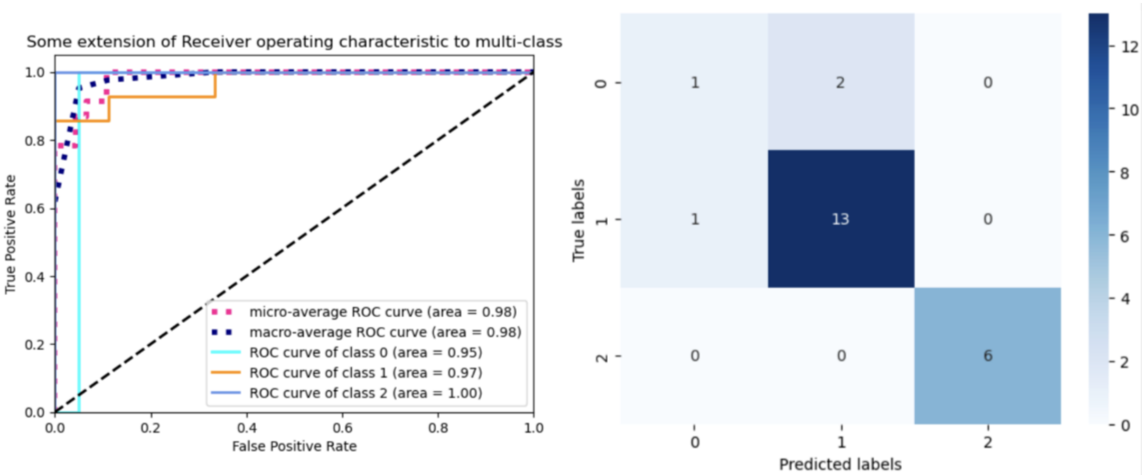


Figure 4.2: ROC curve and confusion matrix for the Swin Transformer model evaluated on the Osteoporosis Knee X-ray dataset.

The Swin Transformer's performance on the two datasets illustrates the impact of dataset size and complexity on model effectiveness. For the Lung Cancer dataset, the Swin Transformer achieves good performance, with metrics nearing perfect scores across accuracy, precision, recall, and F1-score. The model's success in detecting lung cancer is its ability to learn from a large pool of data, which permits it to recognise significant intricate patterns and features.

On the contrary, the model's performance on the Osteoporosis dataset could have been better, as it did not meet the intended accuracy range of 89% to 95%. This could be attributed to the Osteoporosis Knee X-ray dataset's small size, which consists of only 242 images, posing difficulty for the Swin Transformer to analyse it precisely. This can be attributed to the Swin Transformer, which leverages complex self-attention mechanisms and generally requires substantial data to fine-tune their parameters for accurate predictions. The limited data size can lead to a less adept model for generalising from the training data to new, unseen images.

The ROC curves and confusion matrixes, shown in Figures 4.1 and 4.2, visually represent the model's performance. For the Lung Cancer dataset, the near-perfect ROC curve area close to 1 indicates a high true positive rate with a low false positive rate, which is ideal for medical diagnostics. On the other hand, the lower performance of the Osteoporosis dataset is visually evident in the confusion matrix, with some misclassifications impacting the overall metrics.

These results demonstrate that while the Swin Transformer is a powerful tool for medical image analysis, its performance is heavily dependent on the availability of comprehensive training data. This model will only perform well when large labelled datasets are available.

4.2 Performance Analysis of Custom Neural Network

The training duration for the Custom Neural Network model and the Swin Transformer is approximately the same. Integrating a reasoning layer in the Custom Neural Network model does not add computational overhead compared to the Swin Transformer. For instance, the lung cancer dataset takes about 23 minutes to process on both models. This efficiency is notable because the reasoning layer enhances the model's interpretative capabilities without compromising training time.

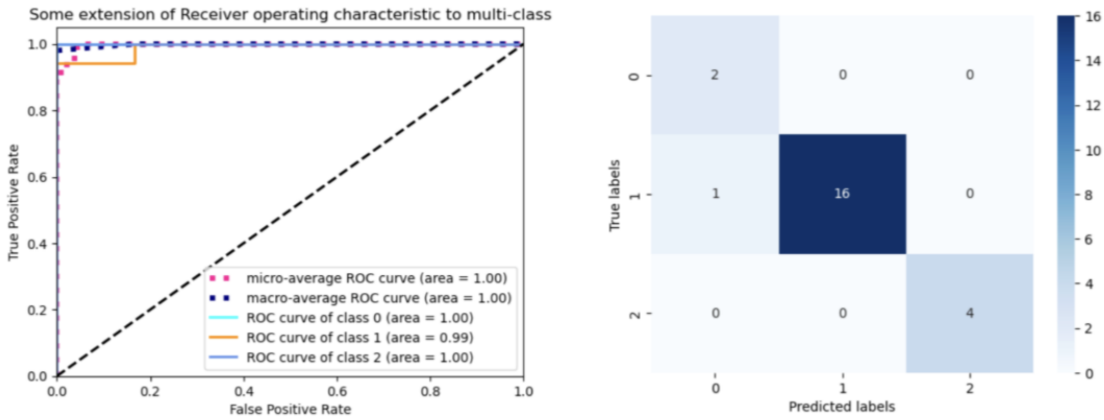


Figure 4.3: ROC curve and confusion matrix for the Custom Neural Network model evaluated on the Osteoporosis Knee X-ray dataset.

The Custom Neural Network model achieves a commendable F1-score of 96.02% for the Osteoporosis Knee X-ray dataset and surpasses the targeted accuracy range. This achievement is particularly significant given that the model includes a Swin Transformer layer, indicating that the additional reasoning layer is critical. The reasoning layer's ability to interpret complex patterns and features within the data likely contributes to improved accuracy, allowing the model to make more nuanced distinctions between different classes despite the smaller dataset size. This suggests that the reasoning layer effectively complements the Swin Transformer's capabilities, providing a more refined analysis. The comparison of the confusion matrix and ROC shown in Figures 4.2 and 4.3 further elaborates on the benefits of the reasoning layer.

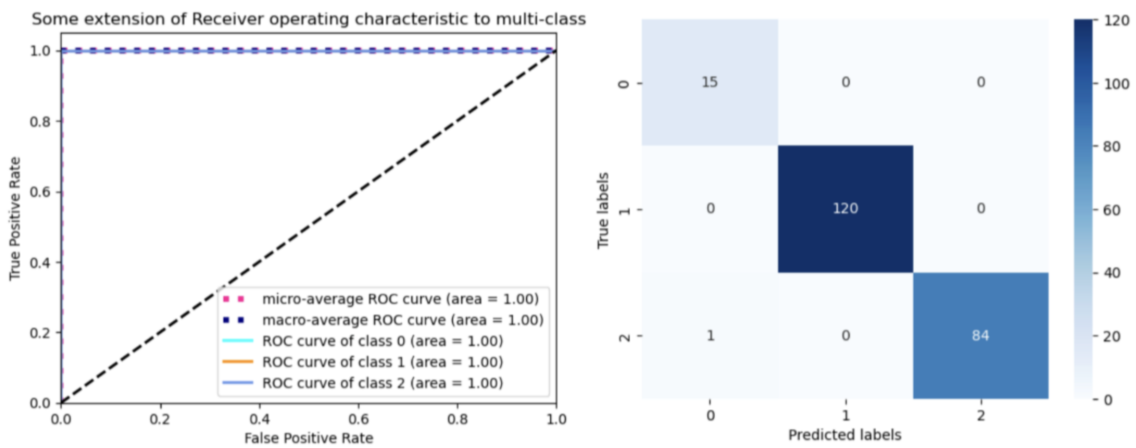


Figure 4.4: ROC curve and confusion matrix for the Custom Neural Network model evaluated on the Lung Cancer dataset.

The Custom Neural Network model accurately classifies lung cancer stages from CT scans, as shown in Figure 4.4. Its robust feature extraction and reasoning capabilities enable accurate predictions with complex data. The model's transformer layer handles extensive data and captures nuanced patterns critical for diagnosis. Nevertheless, the Custom Neural Network model's reasoning layer does not appear to significantly differentiate its performance from that of the standalone Swin Transformer model. This could be because when ample data is available, as with the Lung Cancer dataset, the primary advantage lies in the transformer's ability to process and learn from large amounts of complex data. The reasoning layer's contribution may be less pronounced in this scenario because the model is not constrained by data quantity and can rely on the Swin Transformer layer to capture the intricacies necessary for high performance.

These observations suggest that the Custom Neural Network model brings added value through its reasoning capabilities. In scenarios with limited data, the reasoning layer can markedly elevate performance, highlighting its importance in enhancing the model's interpretative power. However, with substantial data, the model's reasoning element maintains performance without detracting from training efficiency. This is a promising attribute for scaling the model's application to larger, more complex datasets.

4.3 Performance Analysis of CNN

The CNN model achieved impressive metrics of 99.55% accuracy, precision, recall, and F1-score for the Lung Cancer dataset, indicating its ability to distinguish between classes accurately. Moreover, the area under the curve (AUC) for all classes in the ROC curves, shown in Figure .5, further validates the model's performance. This demonstrates a high true positive rate with a low false positive rate, a critical factor for diagnostic tools where misdiagnosis costs are high.

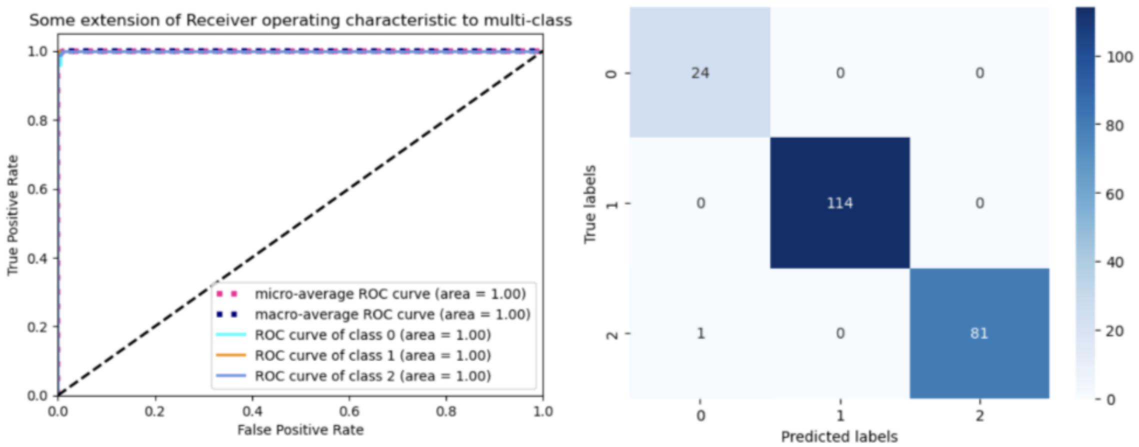


Figure 4.5: ROC curve and confusion matrix for the CNN model evaluated on the Lung Cancer dataset.

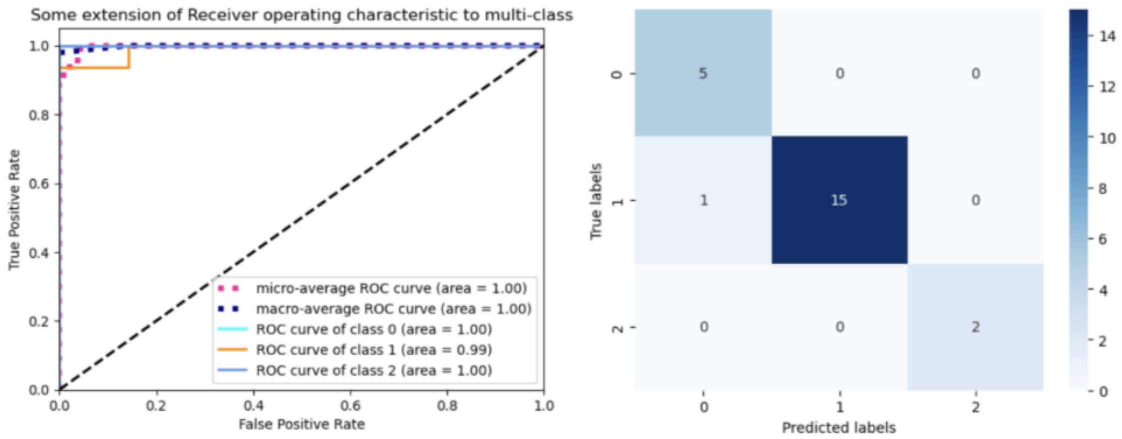


Figure 4.6: ROC curve and confusion matrix for the CNN model evaluated on the Osteoporosis Knee X-ray dataset.

Moving to the Osteoporosis dataset, the CNN model still shows outstanding performance, achieving an accuracy of 95.65%, a precision of 96.38%, a recall of 95.65%, and an F1-score of 95.78%. Despite this dataset being smaller, the CNN model excelled within the desired target range set for the project, demonstrating its robustness even with fewer data points. The confusion matrix and ROC curve, Figure 4.6, indicate each class's low misclassifications and high AUC values.

4.4 Performance Analysis of Hugging Face ViT

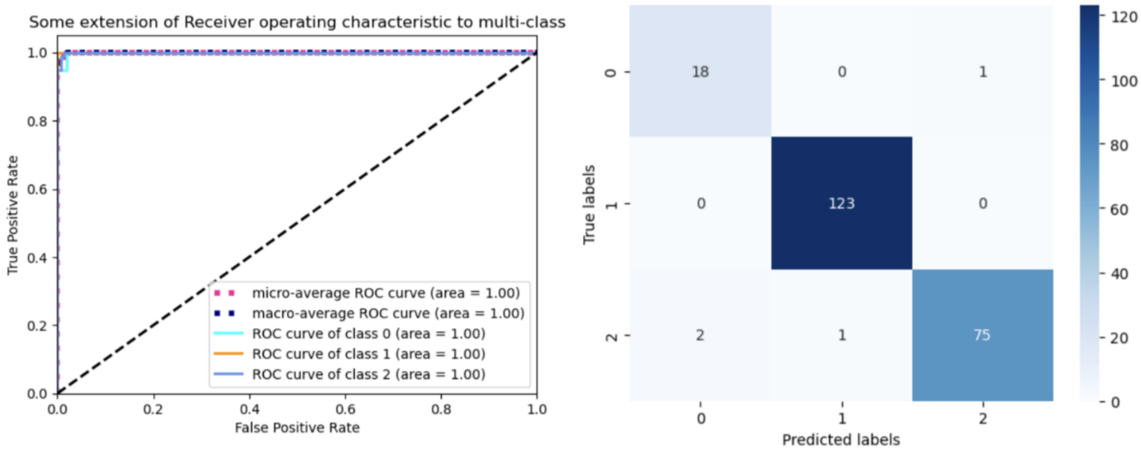


Figure 4.7: ROC curve and confusion matrix for the Hugging Face ViT model evaluated on the Lung Cancer dataset.

The Hugging Face ViT demonstrates significant differences in performance when evaluated on the Lung Cancer and Osteoporosis datasets. While the model performs well on the Lung Cancer dataset, with an accuracy of 98.18%, its results on the Osteoporosis dataset are less impressive, at 82.61% accuracy.

The model's architecture for handling complex patterns and large datasets is attributed to its high accuracy on the lung cancer dataset. ViT's self-attention mechanism improves its ability to distinguish patterns in CT scans. The dataset's ROC curve, from Figure 4.7, shows strong discriminative power for all classes.

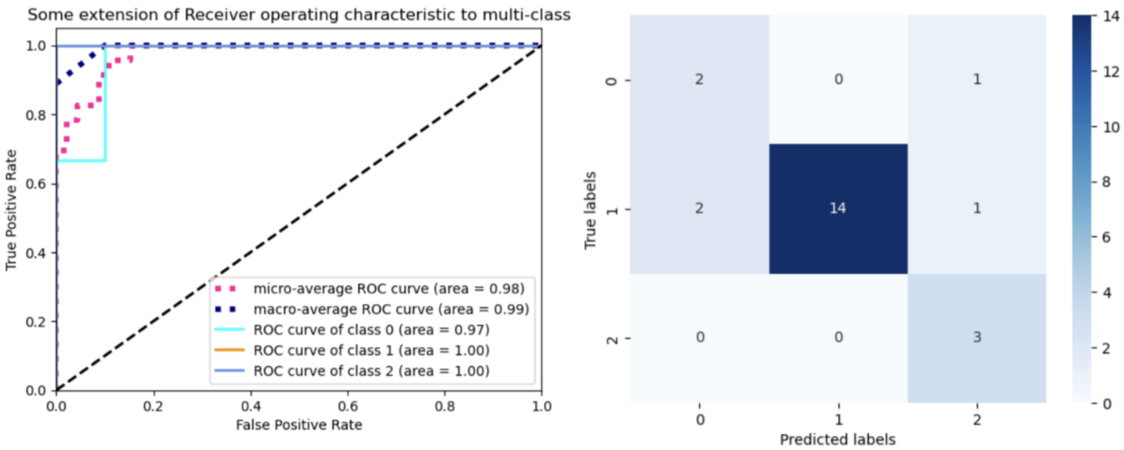


Figure 4.8: ROC curve and confusion matrix for the Hugging Face ViT model evaluated on the Osteoporosis Knee X-ray dataset.

The confusion matrix in Figure 4.8 for the Osteoporosis dataset indicates misclassifications, which could signal an overfitting to the training data or an inability to generalise well on a smaller dataset. Furthermore, although the ROC curve in Figure 4.8 shows high micro-average and macro-average AUC values, it suggests room for improvement in the model's capacity to distinguish between the specific classes of the Osteoporosis dataset.

Several factors may have contributed to the Osteoporosis dataset's decreased performance. Firstly, the Vision Transformer may have yet to fully exhibit its capabilities due to the relatively small dataset size. Like the Swin-Transofmers, it typically requires more data to fine-tune its parameters. Secondly, the X-ray images of osteoporosis contain unique visual patterns that are often more subtle than those found in lung CT scans, requiring the model to identify more nuanced features.

5 Discussion

Simplifying complex AI processes into three core lines of code, as shown in Appendix B, represents an innovative stride in making machine learning more accessible in medical imaging. This section critically evaluates the effectiveness, limitations, and broader implications of this streamlined approach for medical diagnostics.

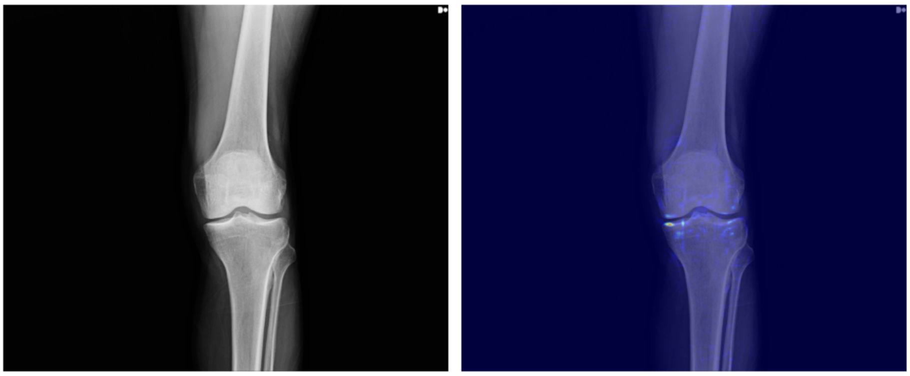
The initial line of code transforms the lengthy task of dataset preparation into a single command. This simplification enables quick model training with minimal preparation, making it user-friendly. However, this convenience may come at the cost of flexibility and precision. Diverse medical datasets with unique characteristics may require more tailored preprocessing to optimise model performance. The challenge is maintaining this simplicity while offering the option for customisation to handle various medical imaging data. However, as shown in the previous section, the current process seems successful with different datasets.

The design choice to reduce the AI model's training and evaluation process into a single line of code creates a streamlined and approachable interface for medical professionals. It hands them the power of advanced AI without requiring a deep understanding of its inner workings. It provides immediate access to advanced diagnostic tools, significantly lowering the entry threshold for practitioners.

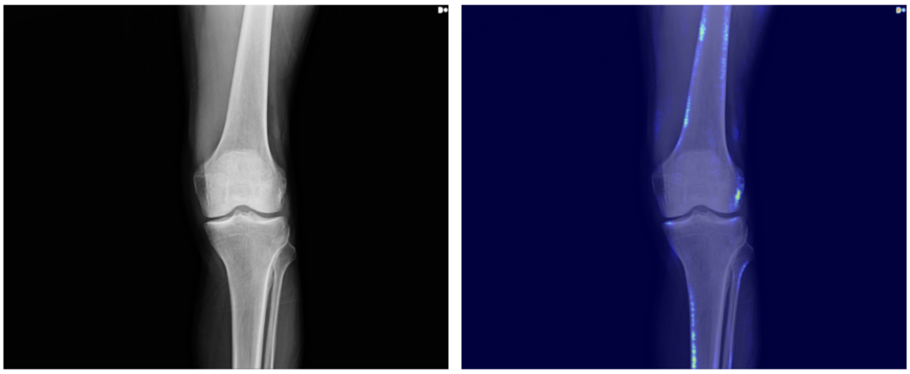
However, this simplicity may also limit opportunities for those wishing to explore the model's deeper aspects. The current implementation does not allow modifications at the model's architectural level. While the system does offer some freedom through adjustable parameters like learning rate and training epochs, these controls might not satisfy users looking to make many changes to the learning algorithm or model structure. The system's design prioritises ease of use and accessibility. It could also lead to a potential trade-off regarding the depth of user engagement and customisation.

The classification functionality within the AI system plays a critical role in delivering insights from medical images. It is designed to be robust and intuitive, providing medical professionals immediate diagnostic probabilities and visual explanations in heat maps. Each model within the system offers a unique approach to image classification, resulting in different heatmap visualisations even when the final diagnostic outcomes are the same.

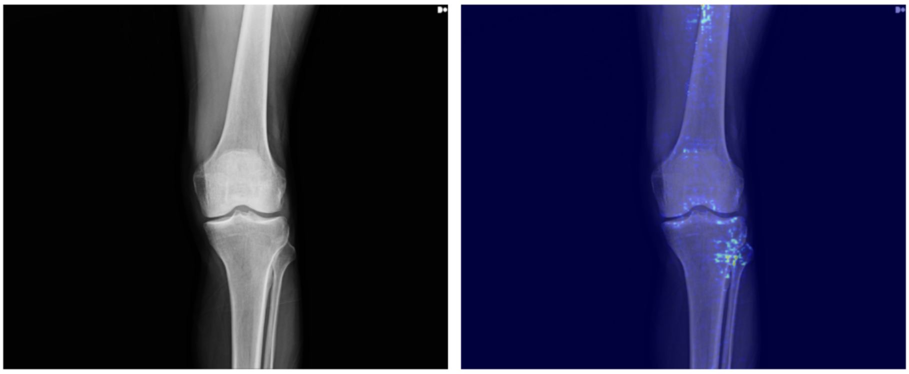
Different heatmaps across models occur because each type interprets image data differently. CNNs detect localised patterns, emphasising contrasts. Swin Transformers process in patches, often highlighting broader regions due to their context-focused attention. The Custom Neural Network model seeks abstract patterns, potentially pinpointing complex relationships. ViTs look at the image holistically.



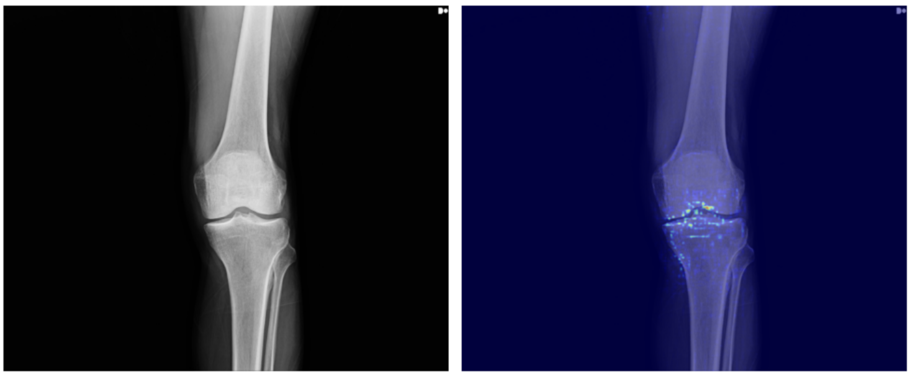
(a) Hugging Face ViT



(b) CNN



(c) Custom Neural Network



(d) Swin-Transformer

Figure 5.1: Outputs from various models on Osteoporosis Knee X-ray - Original image (left) with AI-generated heatmap overlay (right) indicating areas influencing the model's diagnosis.

6 Project Plan

The original project brief, showcased in Appendix A, outlined the proposed scope, objectives, outcomes, and methodologies. It is a reference for evaluating the project's evolution and adherence to its original vision.

6.1 Gantt Chart: Expected vs. Actual Progress

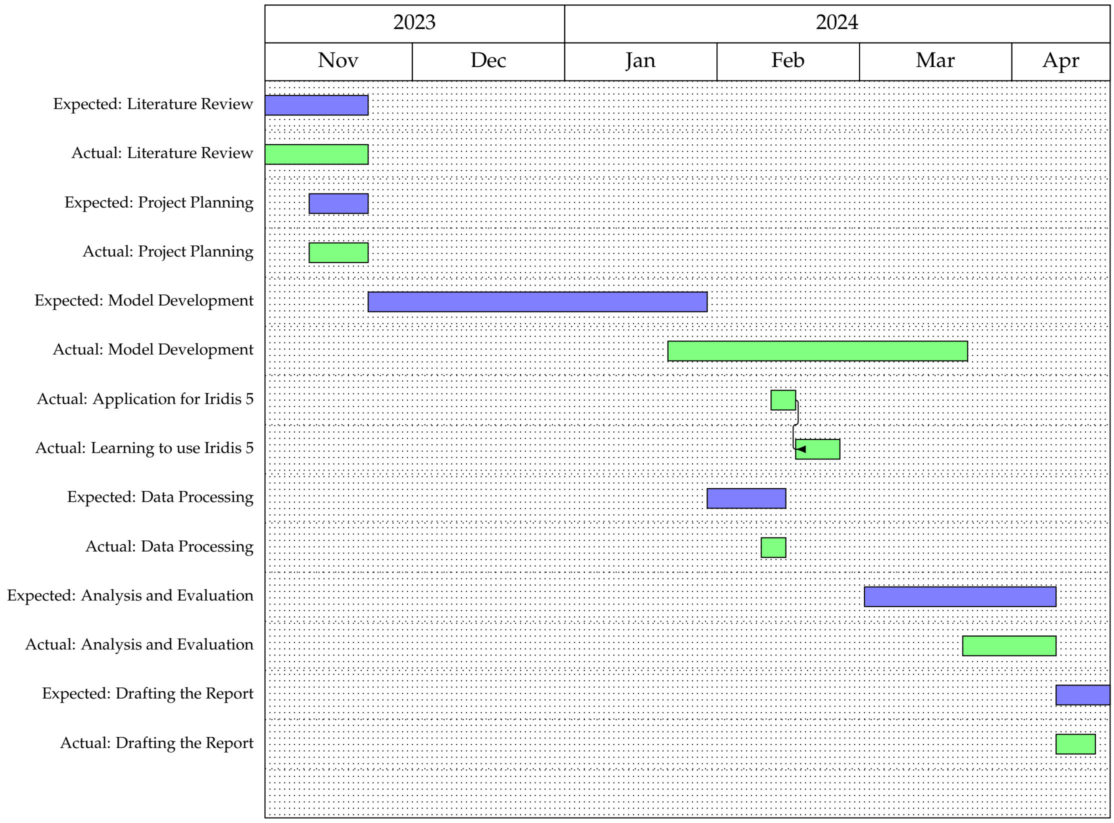


Figure 6.1: Gantt Chart showing expected and actual task durations

The Gantt chart in Figure 6.1 compares expected versus actual progress across various project stages. It reflects the practical challenges faced in the project timeline, notably model development. Initially, the model development was scheduled to start earlier but was postponed due to prioritising other work. Therefore, the planned model development and data processing time were adjusted, showcasing the project's flexibility and ability to adapt to technical constraints. One of these is the need for more computational resources. To combat this, as mentioned before, Iridis 5 was sought after. However, due to connectivity issues, it was not used. Although this led to a compressed timeline for subsequent tasks, the progress shows the ability to adapt and quicken later phases to align with the project timeline. This adaptability is a crucial component of successful project management, especially when navigating the uncertainties of research and development.

6.2 Project Management

The Gantt chart also reflects an agile approach to project management. This strategy was vital when unexpected challenges arose, such as the need for increased computational power. The project was designed with an adaptive methodology, highlighting its dynamic nature and the importance of maintaining flexibility.

The Gantt chart shows differences from the expected plan, particularly in the computational resource acquisition and utilisation phases. The application for Iridis 5 was separate from the original plan but was thought necessary to manage large datasets' computational demands. However, connectivity issues led to its non-utilisation.

Therefore, the scope of data processing was adjusted to align with the available computational resources. This decision excluded two extensive datasets from the analysis, which affected the later stages of the project. The reduced data processing demands allowed for a reallocation of time, accelerating the pace of other project components. This agile approach of making informed decisions under technical constraints enabled the project's continued progress without compromising the research's integrity and objectives.

The final report drafting phase was initiated earlier than planned, as the project's outcomes began to consolidate sooner than expected.

This project plan section highlights the importance of agility and resilience in research. It demonstrates the project's ability to adapt to the dynamic nature of technical research, where not all variables can be predicted or controlled. In conclusion, the project management approach adopted for this initiative allowed for a responsive and adaptable execution while remaining true to the core objectives and outcomes outlined in the original project brief.

7 Conclusion

This project has proven the power of simplification in AI, especially within medical diagnostics. Reducing complex machine learning processes into three core lines of code marks a significant milestone in democratising AI technologies for medical professionals. The outcomes have substantiated that advanced AI can be accessible and robust, providing high accuracy and reliability for critical diagnostic applications.

Throughout the project, these streamlined commands have enabled the rapid and intuitive analysis of medical images, yielding high-accuracy diagnostic predictions complemented by heatmaps for interpretability. This will reduce the entry barrier for medical practitioners and set a precedent for the future of AI-assisted diagnostics.

This design can be improved by exploring model customisation options and enhancing AI decision-making transparency within the simplified code structure.

Future Work:

- **Customisation:** Develop a way to adjust model parameters, making them accessible to users of varying expertise.
- **Explainable AI:** Enhance the system to provide deeper insights into model decisions, reinforcing trust and understanding.
- **Extended Validation:** Implement the models in clinical trials to study their real-world effectiveness.
- **Data Diversity:** To train the models, focus on incorporating a more diverse set of data, including different imaging modalities.

The project's success sets a benchmark for future endeavours in the healthcare industry and AI. It embodies the potential to bridge the gap between complex data science and everyday clinical practice, promising a future where AI augments medical diagnostics with unprecedented user-friendliness.

References

[1] F. Shi, J. Wang, J. Shi, Z. Wu, Q. Wang, Z. Tang, K. He, Y. Shi, and D. Shen, "Review of artificial intelligence techniques in imaging data acquisition, segmentation, and diagnosis for covid-19," *IEEE Reviews in Biomedical Engineering*, vol. 14, pp. 4–15, 2021.

[2] D. Lee and S. N. Yoon, "Application of artificial intelligence-based technologies in the healthcare industry: Opportunities and challenges," *International journal of environmental research and public health*, vol. 18, no. 1, p. 271, 2021.

[3] Y. Kumar, A. Koul, R. Singla, and M. F. Ijaz, "Artificial intelligence in disease diagnosis: a systematic literature review, synthesizing framework and future research agenda," *Journal of ambient intelligence and humanized computing*, vol. 14, no. 7, pp. 8459–8486, 2023.

[4] T. P. Quinn, M. Senadeera, S. Jacobs, S. Coghlan, and V. Le, "Trust and medical ai: the challenges we face and the expertise needed to overcome them," *Journal of the American Medical Informatics Association*, vol. 28, no. 4, pp. 890–894, 2021.

[5] Y. K. Dwivedi, L. Hughes, E. Ismagilova, G. Aarts, C. Coombs, T. Crick, Y. Duan, R. Dwivedi, J. Edwards, A. Eirug, *et al.*, "Artificial intelligence (ai): Multidisciplinary perspectives on emerging challenges, opportunities, and agenda for research, practice and policy," *International Journal of Information Management*, vol. 57, p. 101994, 2021.

[6] H. Yu, L. T. Yang, Q. Zhang, D. Armstrong, and M. J. Deen, "Convolutional neural networks for medical image analysis: state-of-the-art, comparisons, improvement and perspectives," *Neurocomputing*, vol. 444, pp. 92–110, 2021.

[7] A. Bohr and K. Memarzadeh, "The rise of artificial intelligence in healthcare applications," in *Artificial Intelligence in healthcare*, pp. 25–60, Elsevier, 2020.

[8] K. B. Johnson, W.-Q. Wei, D. Weeraratne, M. E. Frisse, K. Misulis, K. Rhee, J. Zhao, and J. L. Snowdon, "Precision medicine, ai, and the future of personalized health care," *Clinical and translational science*, vol. 14, no. 1, pp. 86–93, 2021.

[9] D. D. Miller and E. W. Brown, "Artificial intelligence in medical practice: the question to the answer?," *The American journal of medicine*, vol. 131, no. 2, pp. 129–133, 2018.

[10] A. Kaur, R. Garg, and P. Gupta, “Challenges facing ai and big data for resource-poor healthcare system,” in *2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pp. 1426–1433, IEEE, 2021.

[11] A. S. Maiya, “ktrain: A low-code library for augmented machine learning,” *Journal of Machine Learning Research*, vol. 23, no. 158, pp. 1–6, 2022.

[12] W. Pan, X. Fang, Z. Zang, B. Chi, X. Wei, and C. Li, “Diagnostic efficiency of artificial intelligence for pulmonary nodules based on ct scans,” *American Journal of Translational Research*, vol. 15, no. 5, p. 3318, 2023.

[13] D. Ma, M. R. Hosseinzadeh Taher, J. Pang, N. U. Islam, F. Haghighi, M. B. Gotway, and J. Liang, “Benchmarking and boosting transformers for medical image classification,” in *MICCAI Workshop on Domain Adaptation and Representation Transfer*, pp. 12–22, Springer, 2022.

[14] C. Özer, A. Güler, A. T. Cansever, D. Alis, E. Karaarslan, and İ. Öksüz, “Shifted windows transformers for medical image quality assessment,” in *International Workshop on Machine Learning in Medical Imaging*, pp. 425–435, Springer, 2022.

[15] J. Maurício, I. Domingues, and J. Bernardino, “Comparing vision transformers and convolutional neural networks for image classification: A literature review,” *Applied Sciences*, vol. 13, no. 9, p. 5521, 2023.

[16] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, *et al.*, “An image is worth 16x16 words: Transformers for image recognition at scale,” *arXiv preprint arXiv:2010.11929*, 2020.

[17] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, “Swin transformer: Hierarchical vision transformer using shifted windows,” in *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 10012–10022, 2021.

A Original Project Brief

Investigating How to Empower Clinicians with AI for CT Scan Classification

Name	Supervisor
Apishan Thayananthan (at16g21)	Dr. Mercedes Arguello Casteleiro

Problem:

The medical field is witnessing an increasing prevalence of conditions detectable through CT scans. The challenge lies not only in harnessing advanced technologies to aid in the rapid and accurate diagnosis of these scans but also in ensuring that these technologies are user-friendly. The complexity of AI systems, with numerous adjustable parameters, can deter medical professionals and limit clinic adoption.
Accurate and early diagnosis is crucial for effective treatment. An AI system that can classify multiple conditions from CT scans can streamline the diagnostic process, making it more efficient and manageable for medical professionals.

Goal:

This project aims to develop a versatile, user-friendly AI system capable of classifying and distinguishing between various conditions using CT scan images. By allowing simple parameter adjustments, the system will empower medical professionals to tailor the AI's diagnostic process to specific conditions. The ultimate aspiration is to revolutionise the diagnostic process, making it more accessible, accurate, and efficient for medical professionals.
To ensure the system's user-friendliness, we'll minimise the number of adjustable parameters. By automating most decisions and offering intuitive controls for essential tweaks, we aim to simplify the user interface. This approach will allow medical professionals to harness AI's capabilities without navigating complex technicalities, streamlining the diagnostic process.

Goals and Scope:

The dataset from Kaggle comprises of CT scans labelled for various conditions, for example:

- Osteoporosis (https://www.kaggle.com/datasets/chzpan/bone-lab)
- Cervical Spine Fractures (https://www.kaggle.com/competitions/rsna-2022-cervical-spine-fracture-detection/data)
- Lung Cancer (https://www.kaggle.com/datasets/adityamahimkar/iqothnccd-lung-cancer-dataset/data)
- Liver Tumour (https://www.kaggle.com/datasets/andrewmvd/liver-tumor-segmentation/data).

Existing diagnostic methods rely on radiologists' expertise, which can be time-consuming and subject to human error. This project will investigate multiple AI techniques, including:

- Convolutional Neural Networks (CNNs)
- Support Vector Machines (SVM)
- Random Forest

The choice of these methods is motivated by their proven efficacy in image analysis and classification tasks. Models will be trained using robust cloud-based platforms like AWS or Google Cloud. The primary focus will be on accuracy, sensitivity/specificity, and F1 score for each method. Additionally, computational efficiency, ease of implementation, and model interpretability will be considered.

B Final Design

Swin-Transformer

```
# Import the framework
import framework as fr

# Prepare the dataset
dataset, class_names = fr.prepare_dataset('File Path To Dataset')

# Train and evaluate model of choice
model = fr.train_and_evaluate({"model_name": "Model Name", "num_classes": "Number of
    Classes"}, dataset, {"Custom Config"})

# Classify certain Image
fr.classify('File Path To Image', model, class_names)
```

Small Dataset: Osteoporosis Knee X-ray

```
# Import the framework
import framework as fr

# Prepare the dataset
dataset, class_names = fr.prepare_dataset('Osteoporosis Knee X-ray')

# Train and evaluate model
model = fr.train_and_evaluate({"model_name": "swin_transformer", "num_classes": 3},
dataset, {})

# Classify certain Image
fr.classify('Osteoporosis Knee X-ray/osteoporosis/OS49.jpg', model, class_names)
```

Large Dataset: Lung Cancer

```
# Import the framework
import framework as fr

# Prepare the dataset
dataset, class_names = fr.prepare_dataset('lung cancer/The IQ-OTHNCCD lung cancer
dataset/The IQ-OTHNCCD lung cancer dataset')

# Train and evaluate model
model = fr.train_and_evaluate({"model_name": "swin_transformer", "num_classes": 3},
dataset, {})

# Classify certain Image
fr.classify('lung cancer/The IQ-OTHNCCD lung cancer dataset/The IQ-OTHNCCD lung
cancer dataset/Normal cases/Normal case (11).jpg', model, class_names)
```

Custom Neural Network

```
# Import the framework
import framework as fr

# Prepare the dataset
dataset, class_names = fr.prepare_dataset('File Path To Dataset')

# Train and evaluate model of choice
model = fr.train_and_evaluate({"model_name": "Model Name", "num_classes": "Number of
   Classes"}, dataset, {"Custom Config"})

# Classify certain Image
fr.classify('File Path To Image', model, class_names)
```

Small Dataset: Osteoporosis Knee X-ray

```
# Import the framework
import framework as fr

# Prepare the dataset
dataset, class_names = fr.prepare_dataset('Osteoporosis Knee X-ray')

# Train and evaluate model
model = fr.train_and_evaluate({"model_name": "custom_neural_network", "num_classes":
  3}, dataset, {})

# Classify certain Image
fr.classify('Osteoporosis Knee X-ray/osteoporosis/OS49.jpg', model, class_names)
```

Large Dataset: Lung Cancer

```
# Import the framework
import framework as fr

# Prepare the dataset
dataset, class_names = fr.prepare_dataset('lung cancer/The IQ-OTHNCCD lung cancer
 dataset/The IQ-OTHNCCD lung cancer dataset')

# Train and evaluate model
model = fr.train_and_evaluate({"model_name": "custom_neural_network", "num_classes":
  3}, dataset, {})

# Classify certain Image
fr.classify('lung cancer/The IQ-OTHNCCD lung cancer dataset/The IQ-OTHNCCD lung
 cancer dataset/Normal cases/Normal case (11).jpg', model, class_names)
```

CNN

```
# Import the framework
import framework as fr

# Prepare the dataset
dataset, class_names = fr.prepare_dataset('File Path To Dataset')

# Train and evaluate model of choice
model = fr.train_and_evaluate({"model_name": "Model Name", "num_classes": "Number of
  Classes"}, dataset, {"Custom Config"})

# Classify certain Image
fr.classify('File Path To Image', model, class_names)
```

Small Dataset: Osteoporosis Knee X-ray

```
# Import the framework
import framework as fr

# Prepare the dataset
dataset, class_names = fr.prepare_dataset('Osteoporosis Knee X-ray')

# Train and evaluate model
model = fr.train_and_evaluate({"model_name": "cnn", "num_classes": 3}, dataset, {})

# Classify certain Image
fr.classify('Osteoporosis Knee X-ray/osteoporosis/OS49.jpg', model, class_names)
```

Large Dataset: Lung Cancer

```
# Import the framework
import framework as fr

# Prepare the dataset
dataset, class_names = fr.prepare_dataset('lung cancer/The IQ-OTHNCCD lung cancer
dataset/The IQ-OTHNCCD lung cancer dataset')

# Train and evaluate model
model = fr.train_and_evaluate({"model_name": "cnn", "num_classes": 3}, dataset, {})

# Classify certain Image
fr.classify('lung cancer/The IQ-OTHNCCD lung cancer dataset/The IQ-OTHNCCD lung
cancer dataset/Normal cases/Normal case (11).jpg', model, class_names)
```

Hugging Face ViT

```
# Import the framework
import framework as fr

# Prepare the dataset
dataset, class_names = fr.prepare_dataset('File Path To Dataset')

# Train and evaluate model of choice
model = fr.train_and_evaluate({"model_name": "Model Name", "num_classes": "Number of
  Classes"}, dataset, {"Custom Config"})

# Classify certain Image
fr.classify('File Path To Image', model, class_names)
```

Small Dataset: Osteoporosis Knee X-ray

```
# Import the framework
import framework as fr

# Prepare the dataset
dataset, class_names = fr.prepare_dataset('Osteoporosis Knee X-ray')

# Train and evaluate model
model = fr.train_and_evaluate({"model_name": "hugging_face_vit", "num_classes": 3},
 dataset, {})

# Classify certain Image
fr.classify('Osteoporosis Knee X-ray/osteoporosis/OS49.jpg', model, class_names)
```

Large Dataset: Lung Cancer

```
# Import the framework
import framework as fr

# Prepare the dataset
dataset, class_names = fr.prepare_dataset('lung cancer/The IQ-OTHNCCD lung cancer
 dataset/The IQ-OTHNCCD lung cancer dataset')

# Train and evaluate model
model = fr.train_and_evaluate({"model_name": "hugging_face_vit", "num_classes": 3},
 dataset, {})

# Classify certain Image
fr.classify('lung cancer/The IQ-OTHNCCD lung cancer dataset/The IQ-OTHNCCD lung
 cancer dataset/Normal cases/Normal case (11).jpg', model, class_names)
```