Myocardial Infarction Detection Based on ECG: A Combined Approach using Variational Autoencoders (VAE) and Transfer learning

A DISSERTATION REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF

MASTER OF SCIENCE

IN

DATA ANALYTICS

SUBMITTED BY

BHUPINDER SINGH
ID NO. 10634583

SUBMITTED TO:

MS. TERRI HOARE

DUBLIN BUSINESS SCHOOL
8th JANUARY, 2024

Dedication

I dedicate this thesis for my Master of Science in Data Analytics to Professor Ms. Terri Hoare for her tremendous assistance. I sincerely appreciate my family's consistent support and faith in my abilities. This accomplishment is equally theirs and mine. I am especially grateful to my friends, whose friendship made this academic journey even more enjoyable.

Declaration

I, Bhupinder Singh, hereby declare that the work presented in this report is entirely my own and has not been submitted to any other university or educational institution. This work was carried out in fulfillment of the requirements for the Master of Science in Data Analytics degree.

Bhupinder Singh

Table of Contents

List of figures

List of tables

List of algorithms

1. Convolutional Neural Network (CNN)
2. Convolutional Variational Autoencoder (CVAE)
3. InceptionV3
4. Residual Neural Network (ResNet50 and ResNet152)
5. Variational Autoencoder (VAE)
6. Visual Geometry Group (VGG19)

Acknowledgements

I would like to express my sincere gratitude to Professor Terri Hoare for her outstanding leadership and unwavering assistance during this research effort. Her knowledge has been crucial in determining the study's approach and course. Her leadership and commitment to supporting academic performance are really appreciated.

I also want to express my gratitude to the people who created and contributed to the "ECG Images dataset of Cardiac Patients," which is a publicly accessible dataset on the Mendeley website. It has been made possible for the collecting and analysis of ECG images for this research by the dataset's availability under the Creative Commons Attribution 4.0 International licence. This appreciation emphasises how crucial open data projects are to the advancement of science and the encouragement of scholarly cooperation. It is an impressive gift to the world of medical study that someone is prepared to provide this invaluable resource.

Abstract

This work introduces the use of convolutional variational autoencoders (CVAEs) to effectively decrease the dimensionality of electrocardiogram (ECG) images without sacrificing diagnostic information, addressing the critical requirement for early and accurate myocardial infarction (MI) identification. The study uses CNN, InceptionV3, VGG19, ResNet152, and ResNet50 deep learning models to examine how resolution affects model performance. Higher resolution images improve accuracy and specificity more uniformly across models, according to the findings. VGG19 performs particularly well, despite requiring longer training cycles. Notably, the study shows that CNN is efficient and presents itself as a viable model for rapid and reliable MI diagnosis. It also reveals that CVAEs offer a balanced approach, lowering dimensionality while retaining diagnostic accuracy.

Keywords: Convolutional Variational Autoencoders (CVAEs), Electrocardiogram (ECG) Images, Dimensionality Reduction, Myocardial Infarction (MI), Diagnosis, Deep Learning Models, Image Resolution.

List of important abbreviations

1. ECG – Electrocardiography
2. MI - Myocardial Infarction
3. CNN - Convolutional Neural Network
4. CVAE - Convolutional Variational Autoencoder
5. VGG19 - Visual Geometry Group 19 (a specific deep learning model architecture)
6. ResNet152 - Residual Network with 152 layers (a specific deep learning model architecture)
7. ResNet50 - Residual Network with 50 layers (a specific deep learning model architecture)
8. CRISP DM: Cross-Industry Standard Process for Data Mining
9. ReLU: Rectified Linear Unit
10. GAP: Global Average Pooling
11. SGD: Stochastic Gradient Descent
12. MSE: Mean Squared Error
13. MAE: Mean Absolute Error

1. Introduction

1.1 Background

Electrocardiography (ECG) is a key diagnostic technology that provides non-invasive information on the electrical activity of the heart. The scarcity of broad and comprehensive medical datasets is a serious barrier in the field of cardiac research. Limited data availability is a widespread problem that hampers the creation of strong machine learning models, particularly in the setting of cardiovascular illnesses. Furthermore, the inherent sensitivity of medical information needs careful consideration of privacy considerations, emphasizing the significance of efficient algorithms in dealing with limited datasets.

1.2 Problem Statement

This study addresses the fundamental difficulty of constructing reliable and efficient machine learning models for myocardial infarction (MI) identification in the face of limited medical data and hardware constraints. The emphasis is on using Convolutional Variational Autoencoders (CVAEs) for dimensionality reduction to guarantee that critical diagnostic information is kept while minimizing the processing demands imposed by constrained hardware.

1.3 Research Question

Can Convolutional Variational Autoencoders (CVAEs) be effectively employed to reduce the dimensionality of Ch. Pervaiz Elahi Institute of Cardiology's ECG images, enabling the development of accurate deep learning models that can differentiate between MI and non-MI cases, particularly when operating within the confines of hardware limitations?

1.4 Research Objectives

- Examine the effectiveness of CVAEs in reducing the dimensionality of ECG images at Ch. Pervaiz Elahi Institute of Cardiology while keeping critical diagnostic information.
- Examine the performance of several deep learning models (CNN, InceptionV3, VGG19, ResNet152, and ResNet50) on different image resolutions (90x120, 360x480) and their CVAE-reduced equivalents in categorizing MI and non-MI ECG images from the Ch. Pervaiz Elahi Institute of Cardiology.
- Examine the effect of dimensionality reduction on MI detection accuracy under hardware constraints.

1.5 Research Hypothesis

Convolutional Variational Autoencoders (CVAEs) are hypothesized to efficiently reduce the dimensionality of Ch. Pervaiz Elahi Institute of Cardiology's ECG images, allowing deep

learning models to accurately distinguish between MI and non-MI cases even when using hardware with limited computational resources.

1.6 Research Significance

This study intends to give a solution for the creation of accurate and efficient automated systems for MI identification in ECG images from the Ch. Pervaiz Elahi Institute of Cardiology. The suggested technique might considerably improve the efficiency and accuracy of MI diagnosis by addressing both dataset limits and hardware constraints, eventually leading to improved patient outcomes.

1.7 Scope of Research

The study will look at the usefulness of CVAEs and certain deep learning models (CNN, InceptionV3, VGG19, ResNet152, and ResNet50) in distinguishing MI from non-MI ECG images from the Ch. Pervaiz Elahi Institute of Cardiology. The study will cover three distinct image resolutions (90x120, 360x480, and CVAE-reduced 90x120 from 360x480), with a wide range of scenarios considered for robust evaluation.

1.8 Limitations

The key limitations centre with the availability of medical data, namely the data from the Ch. Pervaiz Elahi Institute of Cardiology. Furthermore, the study is bound by hardware specifications, emphasising the necessity for efficient algorithms that are appropriate for the computing resources available.

1.9 Dissertation Roadmap

The dissertation roadmap followed during the research work has been presented in the figure 1.1.

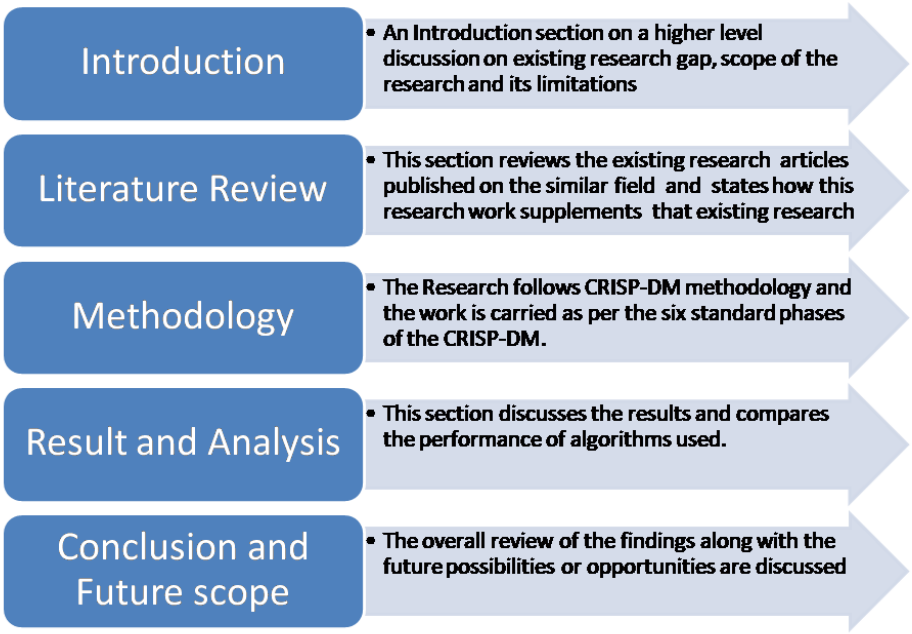

Figure 1.1 Dissertation Roadmap

2. Literature Review

The literature is organized in an orderly path, beginning with the critical need for accurate MI diagnosis (Section 2.1). Section 2.2 emphasizes the need for precise manual ECG interpretation, laying the groundwork for Section 2.3, which delves into early machine learning applications. Section 2.4 focuses on the emergence of deep learning; demonstrating the capabilities of CNNs and LSTMs. Section 2.5 provides dimensionality reduction using VAEs, while Section 2.6 examines current research, highlighting the efficacy of several deep learning models in MI detection. Section 2.7 acknowledges advances but highlights obstacles, emphasizing the necessity for seamless integration of deep learning into clinical practice.

2.1 Traditional ECG Interpretation and application of Machine Learning for interpretation of MI

Myocardial infarction (MI), a common kind of heart disease, continues to be a significant cause of death worldwide, highlighting the urgent need for reliable and efficient diagnostic methods (Labounty and Eagle, 2007). Long used as a primary diagnostic tool for MI detection, electrocardiogram (ECG) signals frequently require the interpretation skills of qualified specialists (Brady et al., 2001). New developments in machine learning, especially the rise of deep learning, have opened the door to improved medical interpretation, offering more automation, effectiveness, and accuracy in this crucial area (Rajpurkar et al., 2017, Ribeiro et al., 2016).

2.2 Importance of Accurate ECG interpretation and challenges in manual interpretation

The diagnosis of myocardial infarction (MI) depends heavily on the accurate interpretation of electrocardiograms (ECGs). To accurately recognize this process, one must possess specialized knowledge and skills (Kligfield et al., 2007). Electrocardiography (ECG) waveforms can exhibit distinct changes, such as ST-segment elevation and T-wave inversion, which often serve as important indicators of the potential for myocardial infarction (Michael et al., 2007). Accurately identifying and interpreting these minute changes is essential to quickly detecting and treating cardiovascular diseases. Electrocardiograms (ECGs) are therefore extremely important as a useful tool in modern cardiology. However, these methods usually need for manual feature extraction, requiring a large investment of knowledge and effort (Stracina et al., 2022). Moreover, bias may be introduced and intricate patterns in the data that machine learning models are able to recognize may not necessarily be captured by human feature extraction (Parvaneh and Rubin, 2018). As a result, there is increasing interest in techniques that might automate this process, cutting down on the time and knowledge needed while also perhaps enhancing the models' performance (Hemanth, 2021).

2.3 Machine learning applications in MI diagnosis

There is a lot of promise in using machine learning to ECG analysis at this early stage to aid in the diagnosis of myocardial infarction (MI). ECG data has been analyzed using a variety of methods, including logistic regression, support vector machines, and decision trees, all of which have helped to accurately identify MI patients. In a research, Osowski et al. (2004) presents an expert system utilising Support Vector Machine (SVM) classification for precise heartbeat recognition. Higher-order statistics (HOS) and Hermite characterisation of QRS complexes in electrocardiogram (ECG) waveforms are two different feature generating techniques used. An efficient expert system is produced by combining these SVM-based classifiers and optimising them using the least mean square technique. This approach's excellence and resilience in reliably recognising heartbeats from ECG waveforms were proved through numerical studies done on 13 distinct heart rhythm patterns. These machine learning techniques have demonstrated promise in helping medical personnel identify and diagnose this potentially fatal illness, offering crucial support during medical decision-making and enhancing patient outcomes.

2.4 Advent of Deep learning in ECG interpretation

Deep Learning's Arrival in ECG Analysis With the emergence of Deep Learning (DL), new opportunities for ECG interpretation have opened up, potentially leading to previously unimaginable powers. Acharya et al. (2017) emphasizes the issue of correctly diagnosing potentially fatal arrhythmias, such as ventricular fibrillation, atrial fibrillation, and atrial flutter, especially when it comes to the elderly. They suggest using a computer-aided diagnostic (CAD) system to evaluate ECG data, which are notoriously difficult and subjective to interpret by hand. This technology would offer an objective and accurate evaluation of the signals. Their eleven-layer deep convolutional neural network (CNN) technique produces amazing results. They get an accuracy of 92.50%, sensitivity of 98.09%, and specificity of 93.13% for two-second ECG segments. The system achieves 94.90% accuracy, 99.13% sensitivity, and 81.44% specificity for five-second segments. These results show that the suggested algorithm has the potential to be an effective tool that helps doctors diagnose patients.Deep learning algorithms are very good at automatically extracting complex characteristics from unprocessed data, which may allow them to outperform more conventional approaches. This paradigm change has the potential to drastically improve MI detection efficiency and accuracy, which might completely transform clinical practice.

ECG Analysis Using Convolutional Neural Networks (CNN) originally intended for image processing, convolutional neural networks (CNNs) has proven to be incredibly useful in ECG interpretation. Utilizing pooling and convolution, these network architectures demonstrate competence in identifying local characteristics. This might help in the effective identification of MI. Liu et al. (2017) presents a novel method for identifying myocardial infarction from multilead electrocardiogram (ECG) data by using a Convolutional Neural Network (CNN). Beat segmentation and fuzzy information granulation are included in the suggested method's

preprocessing. The multilead-CNN (ML-CNN) model is intended to collect multiscale characteristics from several leads. It consists of lead asymmetric pooling (LAP) layers and sub 2-dimensional convolutional layers. In tests utilising real ECG datasets, the system shows good sensitivity (95.40%), specificity (97.37%), and accuracy (96.00%). The approach also shows real-time processing capabilities on ARM Cortex-A9 and MATLAB platforms, suggesting that it may be used in lightweight mobile healthcare apps.

In order to analyse ECG signals, Chauhan and Vig (2015) presents a unique method that combines Long Short Term Memory (LSTM) units with a deep recurrent neural network. This methodology takes an anomaly detection viewpoint, in contrast to conventional approaches that depend on substantial preprocessing and understanding of various arrhythmia forms. With no complex preprocessing, the LSTM model interprets ECG signals directly, having only been trained on normal data. The findings are encouraging, indicating that Deep LSTM models may be useful in identifying irregularities in ECG data, even those related to arrhythmias that have never been identified before. Their ability to accurately detect MI situations is greatly promising due to their ability to capture temporal connections.

In order to predict myocardial infarction (MI) from ECG data, Hasbullah et al. (2023) suggests hybrid models that combine convolutional neural networks (CNN) with recurrent neural networks (RNN), namely CNN-LSTM and CNN-BILSTM. These models combine the advantages of RNNs for temporal information capture with CNNs for autonomous feature extraction. The models show great accuracy using the PTB XL dataset; CNN-BILSTM achieves 91% accuracy and CNN-LSTM achieves 89% accuracy. This shows that the suggested method holds promise for quick and accurate MI identification from ECG data, possibly leading to better patient outcomes through prompt intervention.

2.5 Role of dimensionality reduction in deep learning models

The role of dimensionality reduction through Variational Autoencoder (VAE) in deep learning models has been explored in several studies. Çöl and Ertekin (2021) demonstrated that VAE-based feature dimensionality reduction can improve the performance of Deep Bayesian Active Learning. Similarly, Wang et al. (2016) found that auto-encoder, a key component of VAE, has strong dimensionality reduction abilities, which can contribute to the success of deep learning. Dar and Palanivel (2020) further applied VAE for dimensionality reduction and denoising in the MNIST dataset, achieving better results. These studies collectively highlight the potential of VAE-based dimensionality reduction in enhancing the performance of deep learning models.

2.6 Current research

In the literature, there is substantial evidence suggesting the promise of deep learning techniques in the detection and localization of myocardial infarction (MI) using electrocardiogram (ECG) images. Xiong et al. (2022) provides a comprehensive review of 59 major deep learning studies,

emphasizing the widespread use of convolutional neural networks (CNN) in MI detection and localization, often achieving accuracies surpassing 97%. Ogrezeanu et al. (2020) evaluates various deep learning models for myocardial ischemia detection, demonstrating classification accuracies of up to 88.6%. Manimekalai and Kavitha (2020) propose an innovative deep learning model that combines CNN and long short-term memory (LSTM) networks, achieving an accuracy of 88.89% for MI classification. Sraitih and Jabrane (2022) reviewed various deep learning approaches for classifying ECG heartbeat arrhythmias and highlighted the potential for further improving classification performance. Collectively, these papers highlight the current research landscape regarding the application of deep learning techniques for MI detection using ECG images, emphasizing their capacity to enhance diagnostic accuracy in this field. A selection of the most important technical articles from 2015 to 2023 is displayed in table 2.1 under.

Table 2.1: Key technical papers

Model type	Year	Authors	Findings
Deep learning only	2023	Hadiyoso et al.	The author's research shows that a signal image-based classification method using transfer learning with VGG, AlexNet, and DenseNet can classify multi-class ECG signals with 92% accuracy and F1-score. DenseNet architecture proves particularly effective. The study underscores the importance of routine ECG assessments for heart disease prevention.
Deep learning only	2022	Rawi et al.	The findings show the rising popularity of Convolutional Neural Networks in biological signal recognition, as seen by the increase in related research since 2017. Direct CNN-based classification and function extraction utilizing optimum CNN features have emerged as the two major strategies. These developments highlight the potential of deep learning to improve ECG-based MI and arrhythmia diagnosis.
Deep learning only	2022	Hadiyoso et al.	The author's research shows that a VGG16-based convolutional neural network can detect ECG abnormalities with up to 95% accuracy. This method could assist medical professionals in diagnosing ECG irregularities, especially when commercial ECG equipment's raw data is inadequate.

Deep learning only	2021	Gupta et al.	The study introduces domain-inspired neural network models for myocardial infarction detection. It identifies v6, vz, and ii as key ECG leads and achieves high accuracy with the modified ConvNetQuake model. Remarkably, the multi-ECG-channel network matches a cardiologist's performance without needing human feature extraction or data pre-processing.
Deep learning only	2021	Ukil et al.	The research introduces an automated method for detecting myocardial infarction from single lead ECG data using a highly regularised deep neural network architecture, based on ResNet and both L2 and L1 regularisations. The method outperforms baseline techniques and current advanced algorithms on public ECG datasets, with its effectiveness further confirmed by ablation studies.
Deep learning only	2020	Haroon	The study shows that deep convolutional neural networks, specifically ResNet-50 and VGG-16, achieved up to 99% accuracy in classifying ECG arrhythmias. It highlights the use of Google Colab for complex neural network processing and the effectiveness of transfer learning, particularly with the VGG-16 model.
Deep learning only	2017	Acharya et al.	The author focuses on diagnosing fatal arrhythmias, particularly in the elderly, using a Computer-Aided diagnostic (CAD) system for ECG data interpretation. Their eleven-layer CNN technique achieves up to 94.90% accuracy, 99.13% sensitivity, and 81.44% specificity for five-second ECG segments, demonstrating its potential as a diagnostic tool.

Deep learning only	2017	Liu et al.	The study introduces a method for myocardial infarction detection from multilead ECG data using a CNN. The multilead-CNN (ML-CNN) model, with beat segmentation and fuzzy information granulation preprocessing, captures multiscale features from various leads. It shows good sensitivity (95.40%), specificity (97.37%), and accuracy (96.00%) in tests, and can process in real-time on ARM Cortex-A9 and MATLAB platforms, making it suitable for mobile healthcare apps.
Deep learning only	2015	Chauhan and Vig	The study introduces a unique Long Short Term Memory (LSTM) based method for ECG signal analysis. Unlike traditional methods, this approach requires no complex preprocessing and focuses on anomaly detection. The LSTM model, trained only on normal data, can directly interpret ECG signals. The results suggest that Deep LSTM models can effectively detect ECG irregularities, including previously unidentified arrhythmias.
Hybrid	2023	Hasbullah et al.	The study proposes hybrid models (CNN-LSTM and CNN-BILSTM) for predicting MI from ECG data. These models leverage Recurrent Neural Networks (RNNs) for temporal data and CNNs for feature extraction. With 91% accuracy for CNN-BILSTM and 89% for CNN-LSTM on the PTB XL dataset, these methods show potential for efficient and accurate MI detection, potentially improving patient outcomes.

Hybrid	2023	Ram et al.	The research addresses the high mortality rate of heart disease by employing three deep learning models (MLPs, DBNs, and RBMs) to detect heart disease through ECG signals. The proposed hybrid model outperforms existing models with accuracies of 98.6% and 97.1% on MIT-BIH and PTB-ECG datasets, respectively, demonstrating robustness with excellent F1-score and AUC values.
Hybrid	2022	Rai and Chatterjee	The paper proposes an automated MI detection system using ECG signals, employing CNN, hybrid CNN-LSTM, and an ensemble technique, achieving high accuracies up to 99.89% with a novel data balancing method, SMOTE-Tomek Link. The developed model is deemed ready for clinical applications for MI detection.

2.7 Research gap

Problems and Prospects for the Future Despite the impressive advancements in deep learning for ECG interpretation, a number of obstacles still exist. The most significant of them is that deep learning models require large amounts of carefully labelled data in order to be trained. There are significant obstacles that need to be overcome, including the spectre of overfitting, which is omnipresent, and the problem of interpretability in models (Hawkins, 2004). Looking ahead, a crucial objective that will need coordinated efforts and multidisciplinary collaboration is the smooth incorporation of DL-based ECG interpretation into clinical practice (Topol, 2019).

In summary, the thorough analysis has successfully traversed the terrain of deep learning's expanding impact on MI detection using ECG data. There has been a significant change from the early stages of classical machine learning approaches to the emergence of deep learning. Convolutional neural networks, recurrent neural networks, and hybrid architectures are being included into ECG analysis, which is a significant development that might lead to increased efficacy and accuracy. However, obtaining reliable data, preventing overfitting, and resolving interpretability issues are difficult jobs that require constant focus. The potential integration of deep learning into clinical practice highlights the possibilities of this cutting-edge intersection of technology and healthcare by offering revolutionary promise for improving MI diagnosis and treatment efficacy.

This work creatively blends Convolutional Variational Autoencoders (CVAE) with deep learning models—CNN, InceptionV3, VGG19, ResNet152, and ResNet50—to continue the trend of myocardial infarction (MI) prediction using ECG data. Unlike other hybrid techniques, our method takes advantage of the latent features in CVAE to achieve better temporal information collection and feature extraction. A special dataset from the Ch. Pervaiz Elahi Institute of Cardiology is used to test this novel hybrid model, which was obtained through the Mendeley website. The objective is to identify MIs with greater accuracy, which might lead to breakthroughs in prompt therapies and better patient outcomes.

3. Methodology

The accepted and methodical technique for developing data mining research is known as Cross-Industry Framework for Data Mining (also known as CRISP-DM). The steps involved in this approach is as shown in figure 3.1.

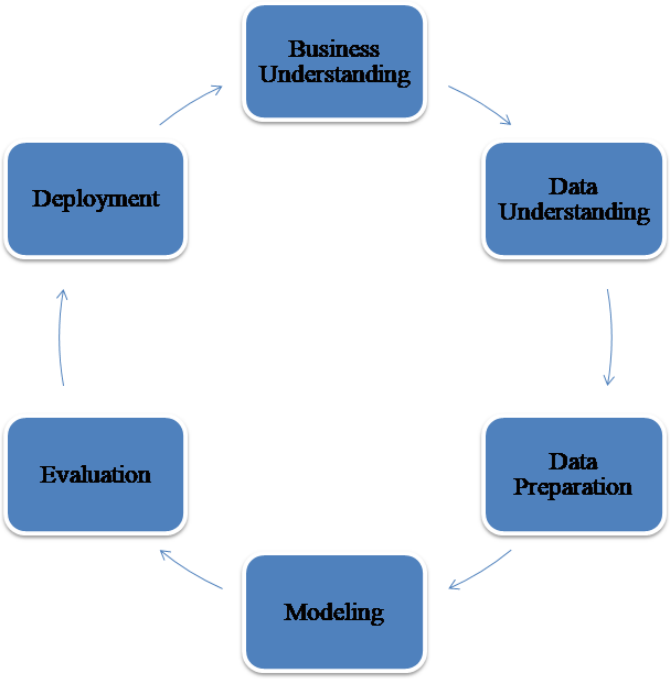

Figure 3.1 CRISP-DM cycle

Business understanding, data understanding, data preparation, modeling, evaluation, and deployment are the six processes that make up the CRISP DM framework (Schröer et al., 2021). Every phase has a significant importance in its own right. The phases of data understanding and business understanding are critical to the framework as a whole. Since the data constitutes the fundamental components of the project, research may be conducted according to the degree of comprehension of the business needs and the data at hand.

We will be changing a few steps in accordance with the requirements. First, we will work on business understanding. Next, we will understand and prepare data. After that, we will work on modeling, which will include parameter specification and implementation. Finally, we will move on to evaluation and deployment.

3.1 Business Understanding

The CRISP-DM process is built on the Business Understanding phase. It includes describing the problem that has to be solved, identifying the stakeholders, and comprehending the project's

commercial objectives. Ensuring that the data mining project is in line with the organization's objectives and that the findings can be put to use is dependent on this phase.

Myocardial infarction (MI), sometimes referred to as a heart attack, must be identified early and accurately in order to facilitate prompt medical attention and enhance patient outcomes. The current tools for diagnosing MI, such blood testing and electrocardiogram (ECG) analysis, can be laborious, subjective, and need specialized knowledge. Deep learning algorithms for automated ECG image categorization have emerged as a viable way to improve MI detection speed and accuracy. In order to meet the demand for automated ECG image classification for MI diagnosis, this research project will examine how well deep learning models and convolutional variational autoencoders (CVAEs) distinguish MI from non-MI ECG images.

3.2 Data Understanding

In the data understanding phase, the data that will be utilized for the data mining project is explored and understood. Data profiling, data quality evaluation, and data visualization are examples of such jobs. This stage aims to provide a comprehensive grasp of the features, constraints, and any problems with the data.

The dataset comprises electrocardiogram (ECG) images of cardiac patients. It was founded by the Ch. Pervaiz Elahi Institute of Cardiology in Multan, Pakistan, to assist cardiovascular disease research (Khan and Hussain, 2021). The example image of the dataset is as shown in the figure 3.2.

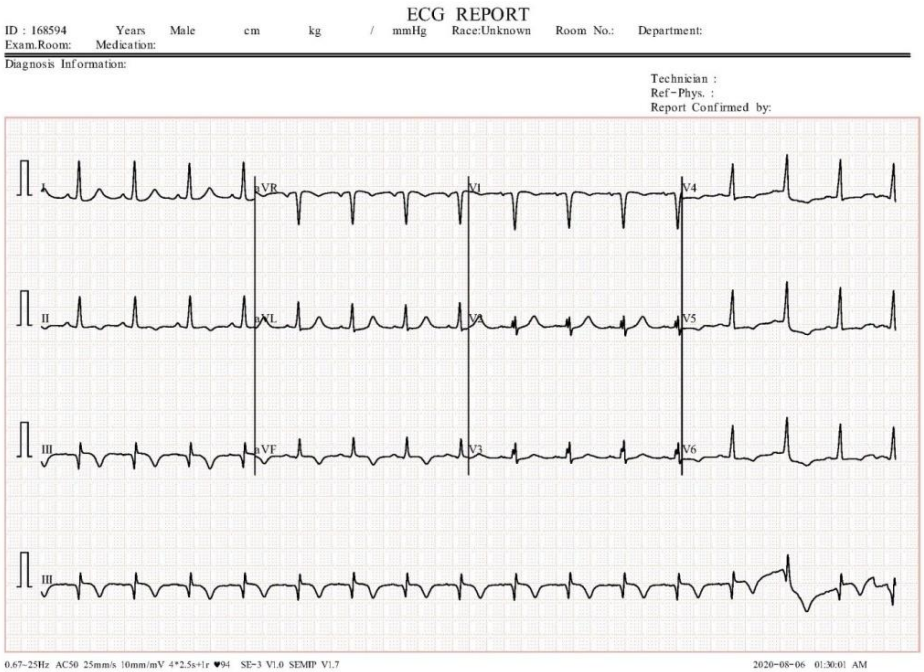

Figure 3.2 One of the images from the dataset

The information or data is divided into four groups or classes, each reflecting a different heart ailment or condition. The tabular form of the dataset is as shown in table 3.1.

Table 3.1: Categories in the dataset

Class	Description	Number of images	Resolution (Height x Width)
0	Myocardial Infarction patients	240	1572×2213
1	Patients with Abnormal heartbeat	233	1572×2213
2	Patients with History of Myocardial infarction	172	1572×2213
3	Normal person	284	1572×2213

3.3 Data Preparation

Cleaning, converting, and getting the data ready for modeling are all part of the data preparation step. This covers activities including treating outliers, addressing missing values, and normalizing data. Ensuring that the data is high-quality and appropriate for modeling is the aim of this step.

The following are the steps in preparing data for training and testing:

- Loading Images: In this stage, the algorithm reads the ECG images from the corresponding folders for each category. This is an important step since it enters the raw data into the programme for further processing. To facilitate processing, the images are imported in grayscale format and scaled to a uniform size of 90x120 pixels for the CNN and InceptionNet models and 360x480 for the CNN and InceptionNet models combined with Convolutional Variational Autoencoders' feature extraction.
- Labeling: Each image is labelled depending on its category. This is an important stage in supervised learning since it correlates a ground truth value (the category) with each input. In this scenario, the categories reflect various heart diseases, allowing the model to learn to appropriately categorise them.
- Train-test split: The dataset is split into two subsets in this step: a training set and a testing set. The testing set is a different collection of data used to assess the model's performance; the training set is used to train the neural network. Usually, the split is performed to make sure the model performs effectively when applied to new data. In the algorithms, a general 80-20 split rule has been taken into consideration.
- Combining data: After being labelled, all of the images' associated labels are combined into a single dataset. A thorough collection of labels and images is produced during this merging process, and this collection will be utilised as a whole to train and test the model.
- Converting into NumPy arrays: The data is transformed into NumPy arrays from a list format. A robust package for Python numerical calculations is called NumPy. During the

training phase, efficient and optimised operations are made possible by converting the data into arrays. Also, NumPy arrays are compatible with a lot of deep learning models.

- Data Normalization: In this stage, the image's pixel values are scaled. The range of values is changed from 0 to 1 by dividing the pixel values by 255. In order for the neural network to properly learn from the input data and provide more reliable and effective training, it is imperative that the data be normalized.

Together, these procedures prepare the data for neural network training, making sure the format is appropriate for the machine learning model to use for learning and producing precise predictions. The flowchart of preprocessing is as shown in figure 3.3.

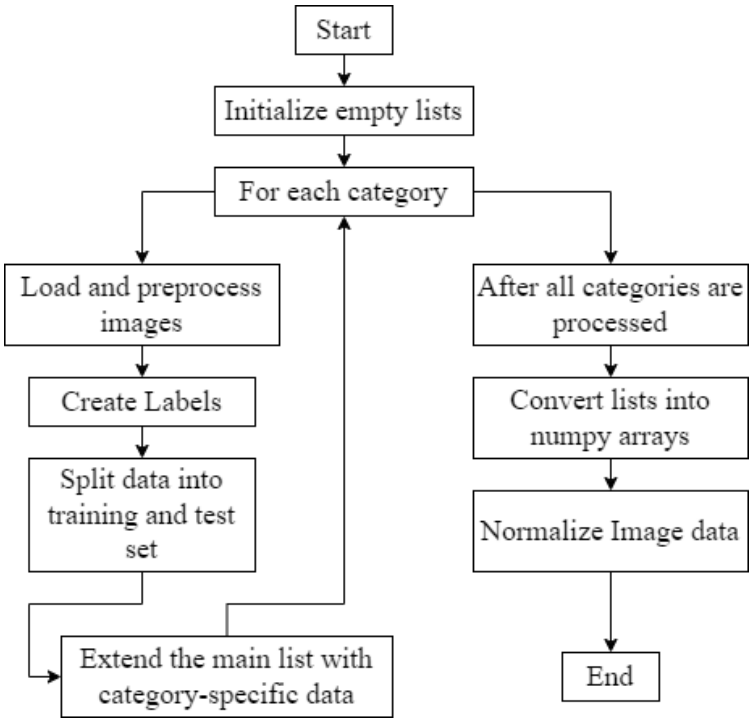

Figure 3.3 Data Preparation

3.4 Modeling

The selection and application of appropriate ideas during the training and testing of a model are crucial elements in the execution of the Deep Learning algorithm, as is the computing capacity and design. All of these modeling segments will be covered in this part.

3.4.1 Convolutional Variational Autoencoder

Feature extraction is the process of selecting and presenting the most significant facts or patterns from raw data (Dustakar et al., 2023). The code uses a convolutional neural network (CNN) based architecture as its autoencoder. One kind of neural network that is trained to discover effective representations (or features) of the input data is called an autoencoder.

An autoencoder consists of two main components: an encoder and a decoder. The overall goal of this design is to utilize the encoder to develop a concise and useful representation of the input data, which the decoder will subsequently use to recreate the original input. The convolutional architectures of encoder and decoder are as shown below in figure 3.4.

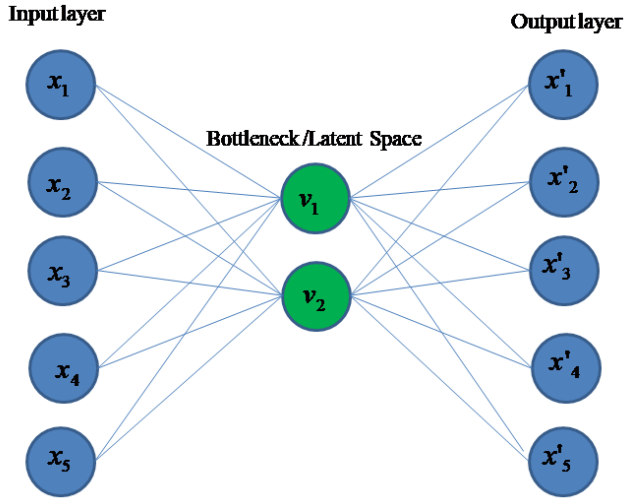

Figure 3.4 Basic Representation of Variational Autoencoder

3.4.1.1 Encoder

The encoder has the architecture as described below with its basic representation as shown in figure 3.5.

- The input data must be converted into a compressed representation by the encoder. This implementation consists of a sequence of max-pooling layers after a set of convolutional layers. From the incoming data, these layers learn to extract features.
- This first convolutional layer uses the ReLU activation function with'same' padding and contains sixteen filters, each of size (3, 3). This indicates that 16 distinct 3x3 filters are applied to the input, producing 16 feature maps as a consequence.
- By using a max-pooling layer with a pool size of (2, 2), the spatial dimensions are cut in half.
- To assist avoid overfitting, a dropout layer with a dropout rate of 0.2 is introduced to provide some regularisation.
- Eight filters with sizes of (3, 3) make up the second convolutional layer. A max-pooling layer and a dropout layer come next.
- A single filter with the ReLU activation function and a size of (3, 3) makes up the final convolutional layer. This generates the compressed representation, which is a single feature map.

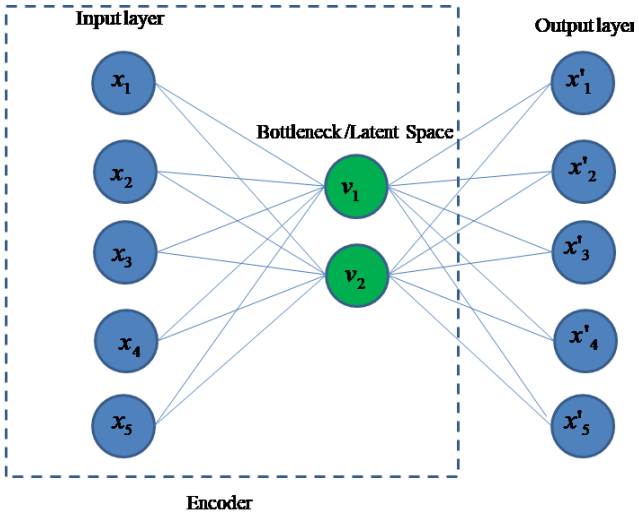

Figure 3.5 Encoder

3.4.1.2 Decoder

The decoder has the architecture as shown below with its basic representation as shown in figure 3.6.

- Reconstructing the original input data from the encoder's compressed form is the decoder's job. Additionally, a sequence of convolutional layers are used to implement it.
- Eight filters make up the first convolutional layer. An upsampling layer with a (2, 2) size follows, essentially doubling the spatial dimensions.
- For regularisation, there is an additional dropout layer added.
- The dropout and upsampling layers come after the second convolutional layer, which includes sixteen filters.
- The sigmoid activation function is used in the single filter of the final convolutional layer. By doing this, an output that is intended to resemble the original input is produced.

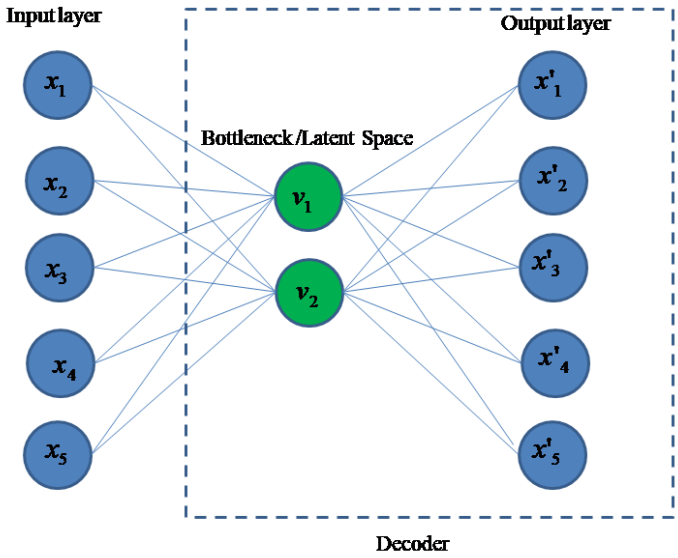

Figure 3.6 Decoder

3.4.1.3 Training the Autoencoder

During autoencoder training, it effectively processes input data, learns to encode and decode information. The encoded features that arise, which are kept in the variables *latent_features_train* and *latent_features_test*, can later be used for other analysis or applications. The loss function used in training is Mean Squared Error (MSE), which quantifies the average squared difference between the original input and the reconstructed output. This option is especially appropriate for reconstruction tasks. The Adam optimizer is used in the optimization process, with a learning rate of 0.0001. Adam's facilitation of learning rate modification assists in optimizing the model during training.

Every one of the autoencoder's 100 epochs represents a full iteration of the training dataset. Because of the longer training time, the model is able to identify complex correlations and patterns in the data. The training data is shuffled at the beginning of each epoch to improve generalization and stop the model from learning the sequence of the training samples. By encouraging a more robust learning process, this randomization helps to enhance model performance and lower the danger of overfitting.

3.4.1.4 Latent feature extraction and model integration

The original images are converted into a latent space of decreased dimensions using the autoencoder's encoder. These latent features extract the most important information from the input images and eliminate the rest. Then, for classification purposes, these latent characteristics are fed into three distinct models (CNN, InceptionV3, VGG19, ResNet50 and ResNet152). By

using a more condensed version of the input data, this method enables the models to operate more effectively and broadly.

3.4.2 Convolutional Neural Networks

A family of deep learning models called Convolutional Neural Networks (CNNs) is mostly employed for the analysis of visual data. CNNs are especially good at tasks like classifying images because of their ability to automatically and adaptively learn the spatial hierarchies of the characteristics present in the input data. The visual representation of CNN architectures is as shown in figure 3.7 and 3.8.

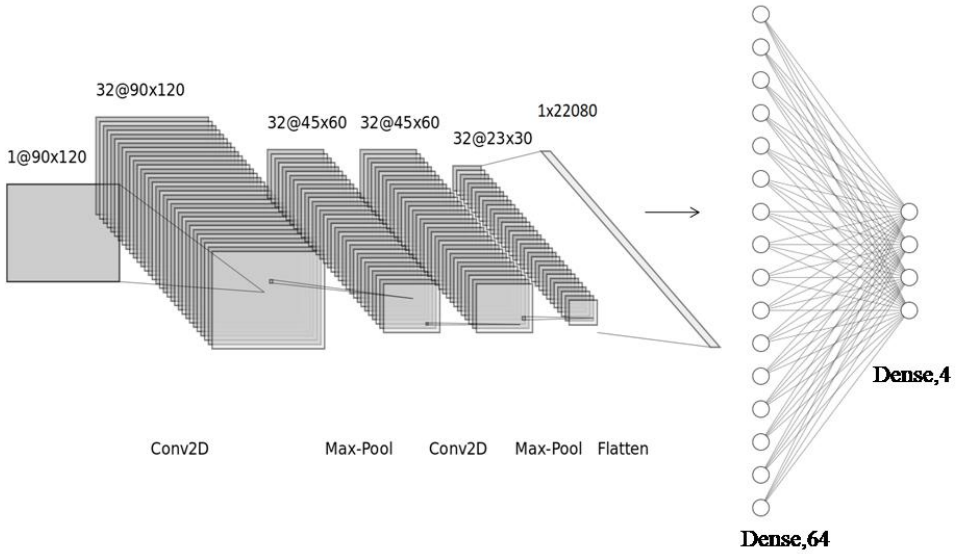

Figure 3.7 Architecture of CNN for 90 by 120 resolution (Produced using NN-SVG and PowerPoint)

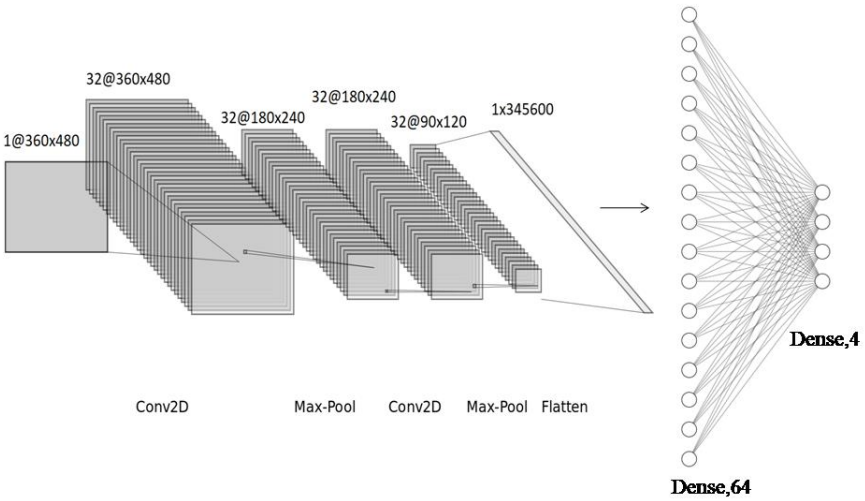

Figure 3.8 Architecture of CNN for 360 by 480 resolution (produced using NN-SVG and PowerPoint)

A CNN's architecture is made to capitalise on an input image's (2D) structure, as well as other 2D inputs like speech signals. Local connections and linked weights are used to do this, and then some kind of pooling produces characteristics that are translation invariant.

The architecture of CNN model used in this project is as described below.

- Input Layer: The input shape is (90, 120, 1) or (360, 480, 1) because experiments are performed on two different resolutions in grayscale format.
- Conv2D Layer: This layer uses 32 filters of size (3, 3) to conduct 2D convolution on the input data. The activation function is'relu,' and the padding is'same,' indicating that the output has the same width and height as the original input.
- MaxPooling2D Layer: This layer does max pooling with a 2x2 pool size, which cuts the spatial dimensions (height and breadth) of the input in half.
- Conv2D Layer: A second convolutional layer with the same settings as the first.
- MaxPooling2D Layer: This is a second max pooling layer with the same settings as the first.
- Flatten Layer: This layer converts 3D inputs to 1D, which may then be utilized as input to fully linked layers.
- Dense Layer: This layer is completely linked and has 64 nodes with 'relu' activation.
- Dense Layer: This is the network's output layer. It is a fully connected layer with four nodes (corresponding to four classes) and 'softmax' activation, which generates a probability distribution over the four classes.

The reasoning for utilizing this CNN for image classification is that it can learn hierarchical representations of the input data automatically, which is necessary for recognizing complicated patterns in ECG images. The addition of 'relu' activation to the model increases non-linearity, helping it to learn more complicated patterns. The activation of 'softmax' in the output layer guarantees that the output values total to 1, allowing them to be read as probabilities. During training, the model learns to change the weights of the connections between nodes to minimize the discrepancy between its predictions and the actual values.

3.4.3 InceptionV3

With over a million images from 1,000 distinct categories, the ImageNet dataset served as the pre-training dataset for the deep convolutional neural network (CNN) that powers the InceptionV3 base model. Because it has already been taught to extract high-level characteristics from images, this pre-trained model is a great place to start when fine-tuning to a particular purpose (Sivakumar, 2023). The architecture of inceptionV3 used in the project is as shown in figure 3.9.

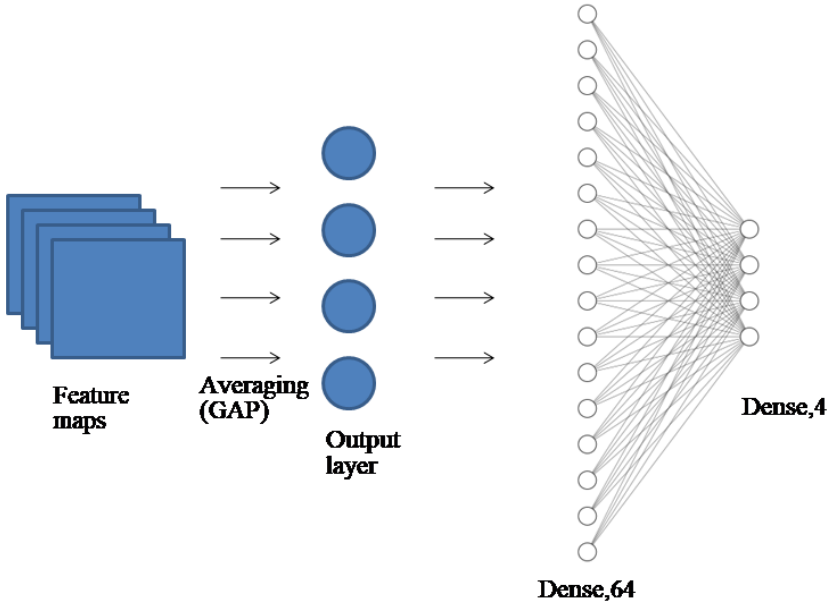

Figure 3.9 Architecture of InceptionV3 (Produced using PowerPoint and NN-SVG)

In order to customize this model for the particular classification task, the following architecture has been used:

- Base Model: The base model extracts features from the input data. These features are known as feature maps.
- Global Average Pooling2D: With this layer, the underlying model's retrieved feature maps' spatial dimensions are condensed into a single output vector.

- Dense (64 neurons, ReLU): Using the ReLU activation function, this layer introduces non-linearity by transforming the output vector from the Global Average Pooling2D layer into a higher-dimensional representation.
- Dense (4 neurons, Softmax): The final output is generated by this layer, which shows the probabilities for the four classes the model has been taught to identify. These probabilities must add up to one, which is ensured by the Softmax activation function.

3.4.4 VGG19

VGG is another transfer learning model. A 19-layer version of the VGG model, called VGG19, has 16 convolution layers, 3 fully linked layers, 5 MaxPool layers, and 1 SoftMax layer (Humayun et al., 2022). The architecture of VGG19 is as shown in figure 3.10.

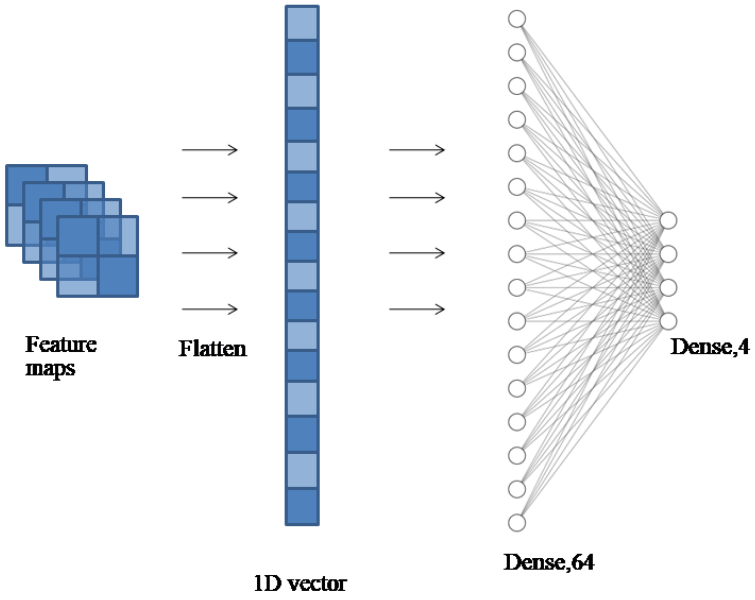

Figure 3.10 Architecture of VGG19 (Produced using PowerPoint and NN-SVG)

The VGG19 model serves as the foundational model for transfer learning in the architecture employed for this project. Since the VGG19 model's weights are configured to be non-trainable, these layers' weights won't change while the model is trained. As a result, the VGG19 model's potent feature extraction capabilities may be utilised by the model. A flatten layer, which converts the base model's 3D output to 1D, comes after the base model. This is required since the base model's output is a 3D tensor, while the subsequent Dense layers require 1D input.There are two Dense layers that follow the Flatten layer. 64 neurons make up the first Dense layer, which activates using the Rectified Linear Unit (ReLU) activation function. The model's output layer, the second Dense layer, includes four neurons—one for each class—and outputs the probabilities for each class using the Softmax activation function. The architecture therefore takes an input

image, flattens the 3D feature maps to 1D, extracts features using the VGG19 base model, then utilises the Dense layers to generate the final predictions based on these features.

3.4.5 ResNet152 and ResNet50

The Residual Network, or ResNet, model has several variations, including ResNet50 and ResNet152. Applied to image classification tasks, they are deep convolutional neural networks, or CNNs. Each of their names has a number that corresponds to the number of network tiers. ResNet50 and ResNet152 offer 50 and 152 layers, respectively. Although ResNet50 is well-liked due to its low processing resource requirements, ResNet152 has more layers and is utilised to extract intricate patterns from images (Narin et al., 2021). The architecture of ResNet models is as shown in figure 3.11.

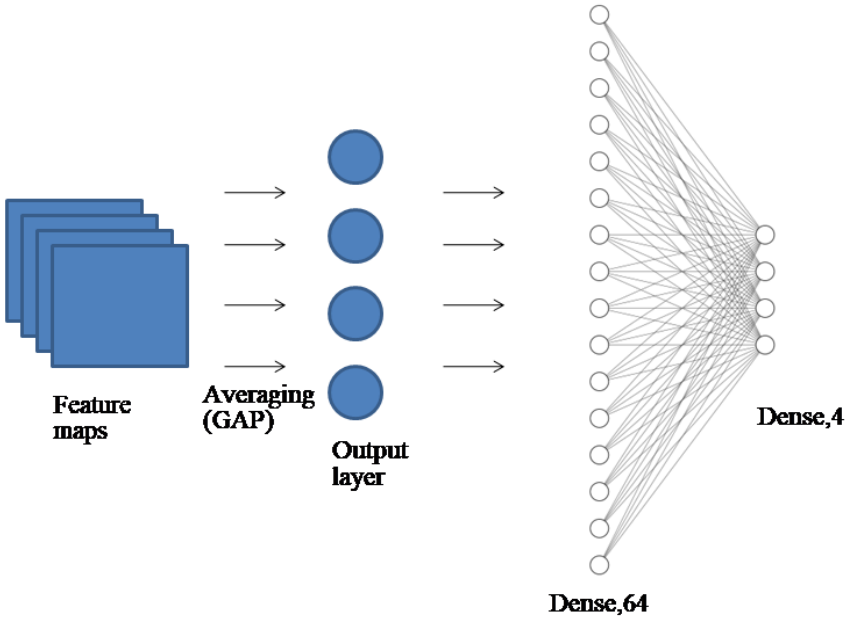

Figure 3.11 Architecture of ResNet50 and ResNet152 models (Produced using PowerPoint and NN-SVG)

ResNet50 or ResNet152 can be used as the basic model for transfer learning in the architecture specified. Since the base model's weights are configured to be non-trainable, these layers' weights won't change while the model is trained. This enables the model to take use of the ResNet model's potent feature extraction capabilities. A Global Average Pooling (GAP) layer that decreases the spatial dimensions of the 3D output of the base model to 1D comes after the base model. This drastically lowers the model's parameter count, which may aid in avoiding overfitting. There are two Dense layers that follow the GAP layer. 64 neurons make up the first Dense layer, which activates using the Rectified Linear Unit (ReLU) activation function. The model's output layer, the second Dense layer, includes four neurons—one for each class—and outputs the probabilities for each class using the Softmax activation function.

Thus, the architecture receives an input image, utilizes the ResNet base model to extract features, reduces the dimensionality of the feature maps by using global average pooling, and employs the Dense layers to provide final predictions based on the features.

3.5 Evaluation

Finally, performance indicators are employed to evaluate the performance of the model that has been trained. They offer a numerical evaluation of the model's prediction performance on the data. For classification tasks, common performance measurements include accuracy, precision, recall, the F1 score, and Specificity.

- **Accuracy:** The ratio of accurately predicted observations to total observations is known as accuracy. The formula of the accuracy is as shown below in equation 3.1.

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total Number of predictions}} \qquad \text{Eq. 3.1}$$

  However if there is an imbalance in the classes, accuracy might be deceiving. In these situations, a model may still achieve high accuracy while predicting the majority class the majority of the time. Therefore, we need to take into consideration other performance metrics as well. Merely relying on accuracy might be deceiving because one of the class in our 4 classes is in minority as compared to other classes.
- **Precision:** The precision measures how many positive observations were successfully predicted out of all the positive observations that were expected. The accuracy for class is computed in multi-class classification. It's computed as shown in equation 3.2.

$$\text{Precision}_i = \frac{(\text{True Positives})_i}{(\text{True Positives})_i + (\text{False Positives})_i} \qquad \text{Eq. 3.2}$$

  Here i denotes a class.
- **Recall (Sensitivity):** Recall is defined as the proportion of accurately predicted positive observations to all observations inside the actual class. Recall is computed for each class in multi-class classification. It's computed as shown in equation 3.3.

$$\text{Precision}_i = \frac{(\text{True Positives})_i}{(\text{True Positives})_i + (\text{False Negatives})_i} \qquad \text{Eq. 3.3}$$

- **F1-score:** Precision and Recall are weighted averages that make up the F1 Score. As such, this score accounts for both false positives and false negatives. Usually, it is more helpful than accuracy, particularly in cases when the distribution of classes is not uniform. One may compute the F1 score as shown in the equation 3.4.

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \qquad \text{Eq. 3.4}$$

- **Specificity:** True negatives divided by the total of true negatives and false positives is the measure of specificity. The ratio of accurately recognized real negatives is measured.

Each class's specificity is determined and then averaged in multi-class categorization. It may be computed as shown in equation 3.5.

$$\text{Specificity}_i = \frac{(\text{True Negatives})_i}{(\text{True Negatives})_i + (\text{False Positives})_i} \qquad \text{Eq. 3.5}$$

4. Result and Analysis

The purpose of this work is to examine how well deep learning models perform in the categorization of ECG images, with a focus on their ability to identify myocardial infarction. Two types of resolutions of image—one with a resolution of 90 by 120 pixels and another with a greater resolution of 360 by 480 pixels—are used for the evaluation. Further investigated is the effect of downsizing the higher-resolution images to 90 by 120 pixels using latent feature extraction from an autoencoder. The confusion matrices and other performance measures are thoroughly examined in the ensuing study, which sheds light on the advantages and disadvantages of each approach. An extensive summary of the outcomes of classifying ECG (Electrocardiogram) images using different resolutions and deep learning models is shown in the table 4.1. Every row denotes a unique configuration for the experiment, including details on the model, the ECG image resolution, and important performance metrics.

Table 4.1: Performance matrix of all models across all resolutions

Resolution	Model	Accuracy	Precision (Avg)	Recall (Avg)	F1 Score (Avg)	Specificity (Avg)	Training Time (s)
90x120	CNN	87.70%	0.87	0.87	0.87	0.96	30.76
90x120	InceptionV3	84.49%	0.85	0.83	0.83	0.95	96.34
90x120	VGG19	77.01%	0.78	0.77	0.76	0.93	65.22
90x120	ResNet152	68.98%	0.74	0.69	0.68	0.9	286.48
90x120	ResNet50	59.36%	0.61	0.57	0.56	0.86	141.21
360x480	CNN	91.98%	0.96	0.92	0.92	0.97	149.6
360x480	InceptionV3	97.33%	0.97	0.97	0.97	0.99	408.13
360x480	VGG19	99.47%	0.99	0.99	0.99	1	581.38
360x480	ResNet152	78.07%	0.85	0.73	0.7	0.92	1757.25
360x480	ResNet50	86.10%	0.86	0.85	0.85	0.95	888.94
Latent 90x120	CNN	94.65%	0.94	0.94	0.94	0.98	44.02
Latent 90x120	InceptionV3	93.05%	0.94	0.93	0.93	0.98	82.5
Latent 90x120	VGG19	94.65%	0.96	0.94	0.95	0.98	88.77
Latent 90x120	ResNet152	79.14%	0.79	0.78	0.78	0.93	293.23
Latent 90x120	ResNet50	82.35%	0.82	0.82	0.82	0.94	148.77

The performance metrics of individual algorithms is discussed in the following sections.

4.1 Resolution 90 by 120 pixel images

4.1.1 CNN performance

This model was trained over 100 epochs using a dataset with 24 batches per epoch. The training process aimed to minimize the categorical crossentropy loss, resulting in an overall classification accuracy of 87.7% on test dataset and it took 30.76 seconds to complete the training process.

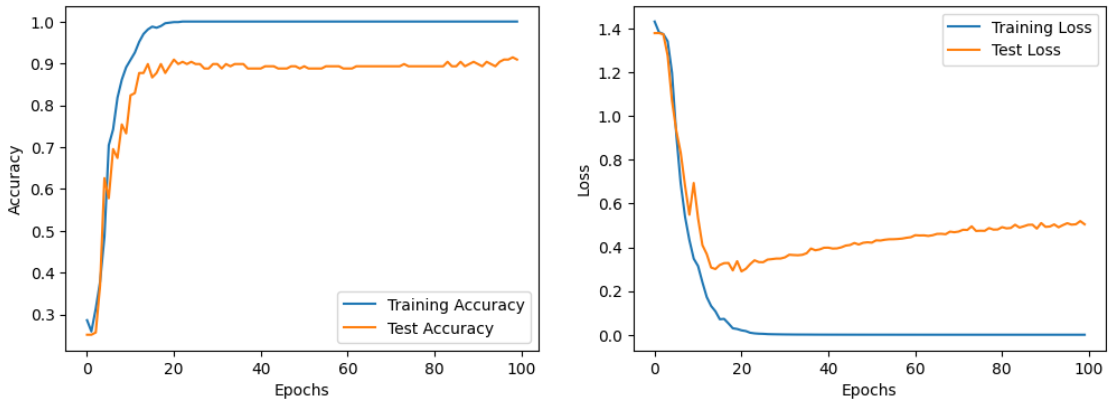

Figure 4.1 Performance curves for CNN at 90 by 120

The model demonstrates its capacity to accurately categorize occurrences. The figure 4.2 provides a visual representation of various measures including recall, accuracy, F1 Score, and specificity, which provide a thorough knowledge of the model's overall performance.

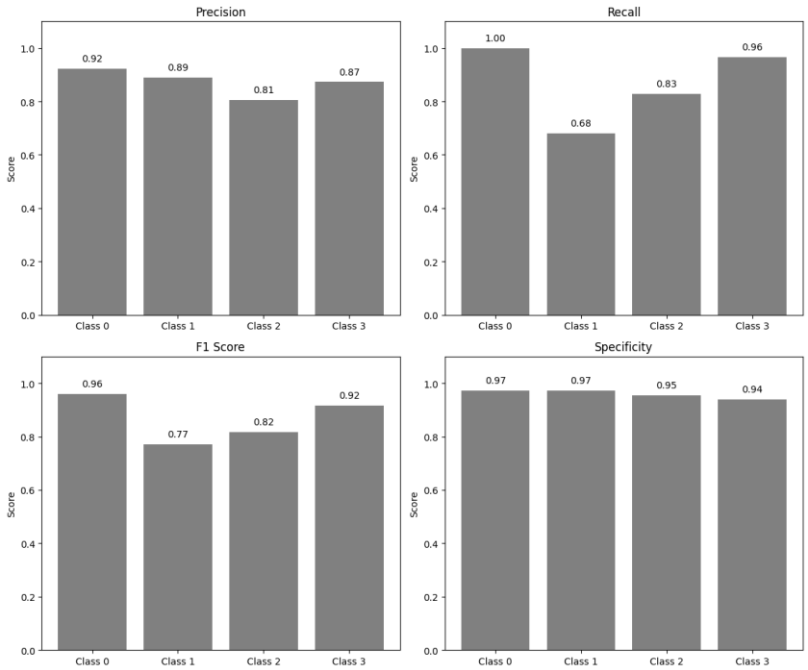

Figure 4.2 Performance metrics of CNN at 90 by 120

4.1.2 InceptionV3 Performance

This model was trained over 100 epochs using a dataset with 24 batches per epoch. The training process aimed to minimize the categorical crossentropy loss, resulting in an overall classification accuracy of 84.49% on test dataset and it took 96.34 seconds to complete the training process.

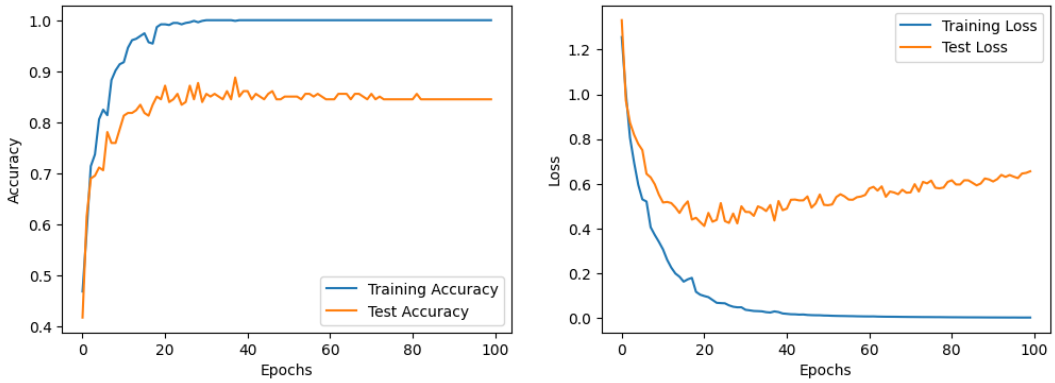

Figure 4.3 Performance curves of InceptionV3 at 90 by 120

The figure 4.4 provides a visual representation of various measures including recall, accuracy, F1 Score, and specificity, which provide a thorough knowledge of the model's overall performance.

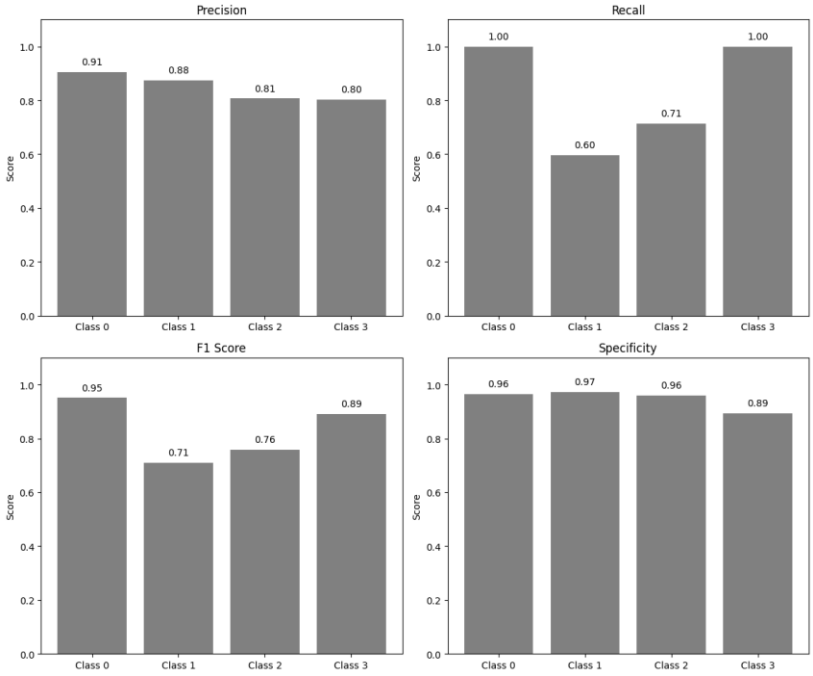

Figure 4.4 Performance metrics of inceptionV3 at 90 by 120

4.1.3 VGG19 Performance

This model was trained over 100 epochs using a dataset with 24 batches per epoch. The training process aimed to minimize the categorical crossentropy loss, resulting in an overall classification accuracy of 77.01% on test dataset and it took 65.22 seconds to complete the training process.

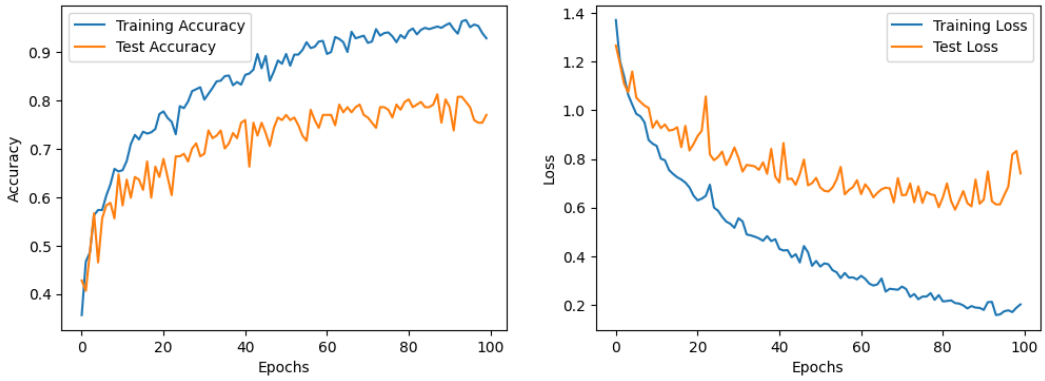

Figure 4.5 Performance curves of VGG19 at 90 by 120

The figure 4.6 provides a visual representation of various measures including recall, accuracy, F1 Score, and specificity, which provide a thorough knowledge of the model's overall performance.

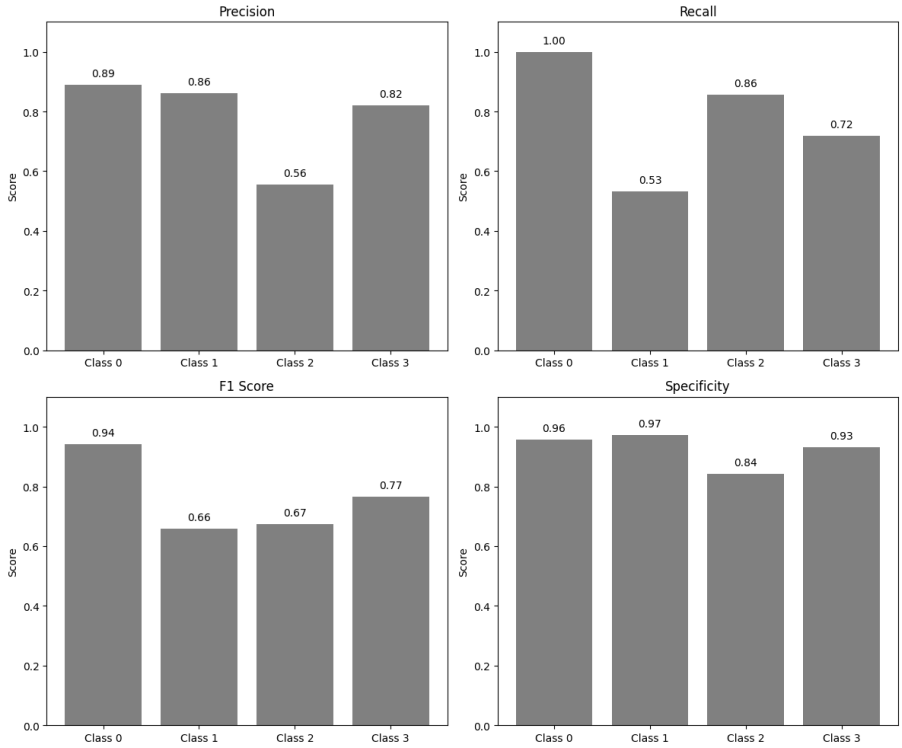

Figure 4.6 Performance metrics of VGG19 at 90 by 120

4.1.4 ResNet152 Performance

This model was trained over 200 epochs using a dataset with 24 batches per epoch. The training process aimed to minimize the categorical crossentropy loss, resulting in an overall classification accuracy of 68.98% on test dataset and it took 286.48 seconds to complete the training process.

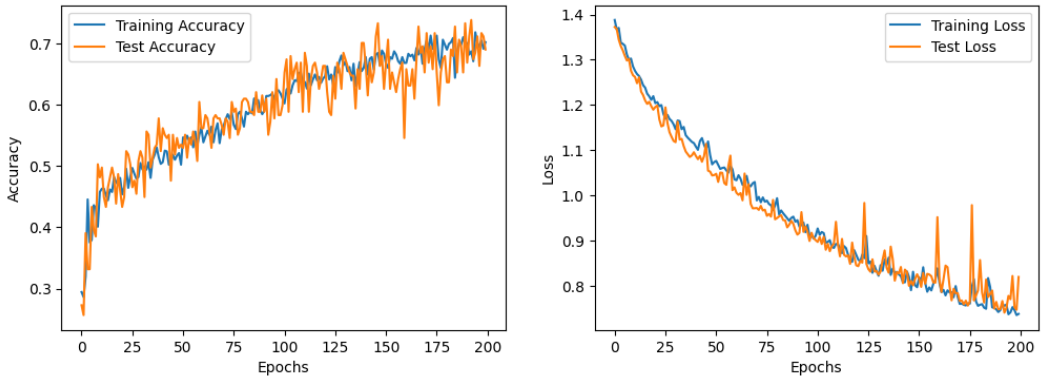

Figure 4.7 Performance curves of ResNet152 at 90 by 120

The figure 4.8 provides a visual representation of various measures including recall, accuracy, F1 Score, and specificity, which provide a thorough knowledge of the model's overall performance.

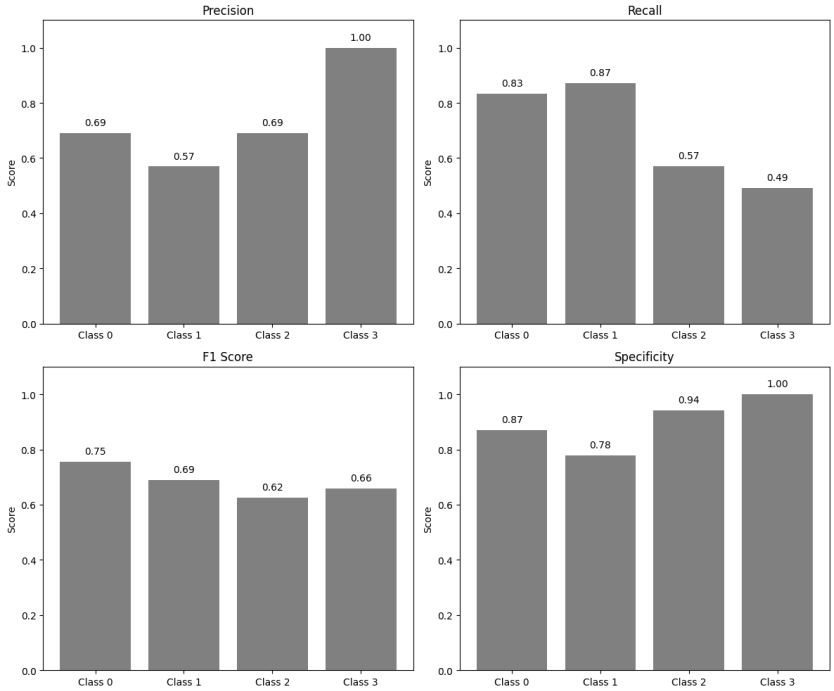

Figure 4.8 Performance metrics of ResNet152 at 90 by 120

4.1.5 ResNet50 Performance

This model was trained over 200 epochs using a dataset with 24 batches per epoch. The training process aimed to minimize the categorical crossentropy loss, resulting in an overall classification accuracy of 59.36% on test dataset and it took 141.21 seconds to complete the training process.

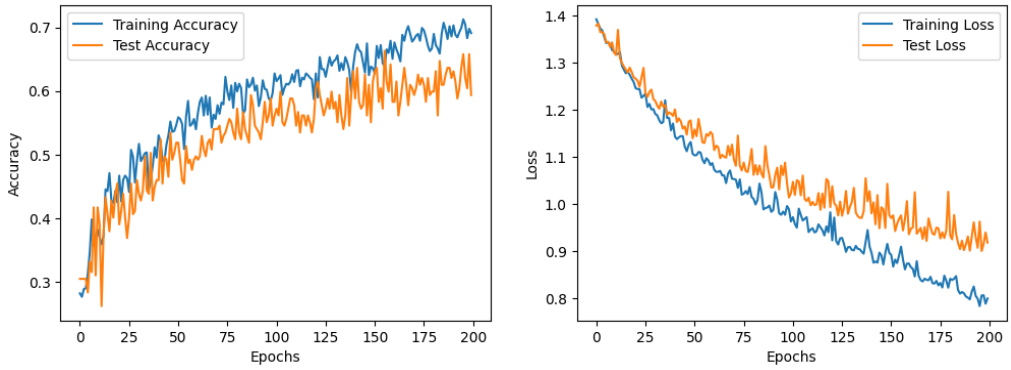

Figure 4.9 Performance curves of ResNet50 at 90 by 120

The figure 4.10 provides a visual representation of various measures including recall, accuracy, F1 Score, and specificity, which provide a thorough knowledge of the model's overall performance.

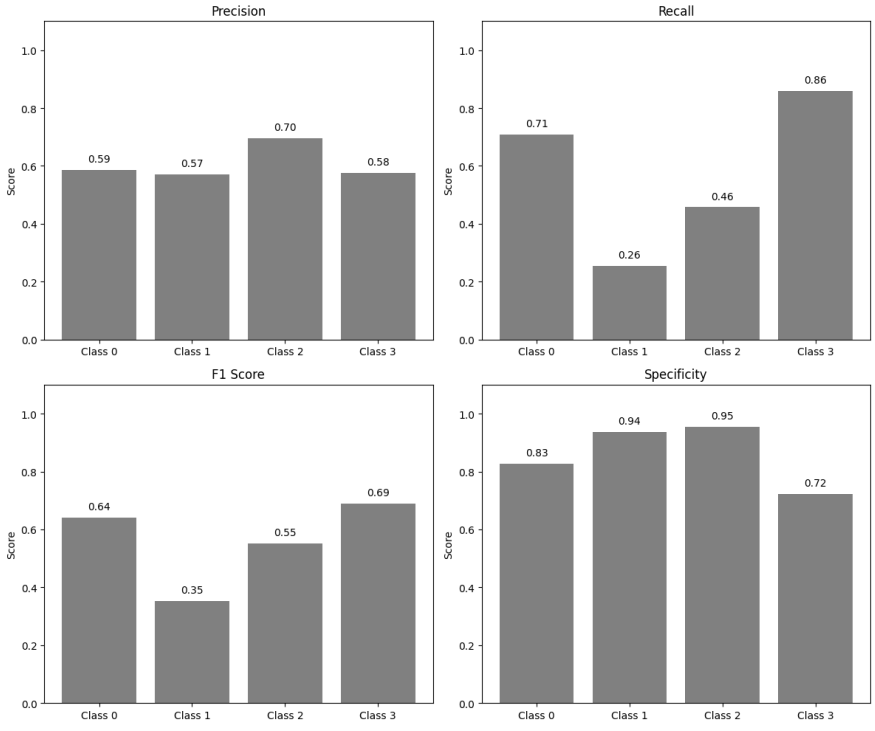

Figure 4.10 Performance metrics of ResNet50 at 90 by 120

4.2 Resolution 360 by 480 pixels images

4.2.1 CNN Performance

This model was trained over 100 epochs using a dataset with 24 batches per epoch. The training process aimed to minimize the categorical crossentropy loss, resulting in an overall classification accuracy of 91.98% on test dataset and it took 149.6 seconds to complete the training process.

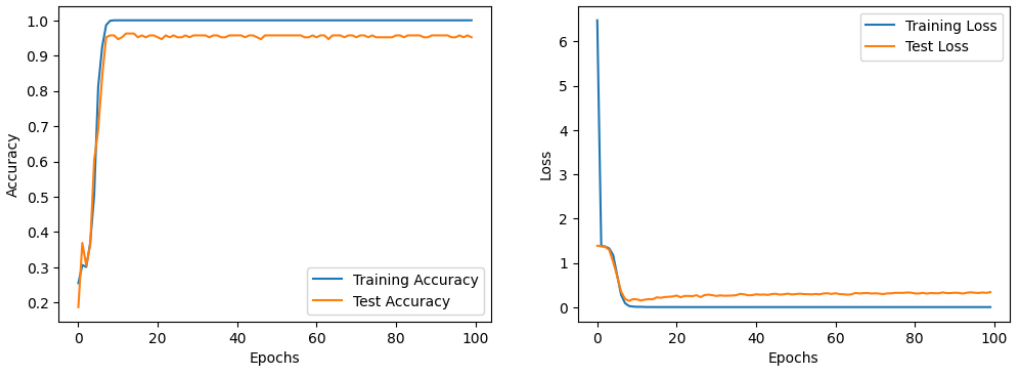

Figure 4.11 Performance curves of CNN at 360 by 480

The figure 4.12 provides a visual representation of various measures including recall, accuracy, F1 Score, and specificity, which provide a thorough knowledge of the model's overall performance.

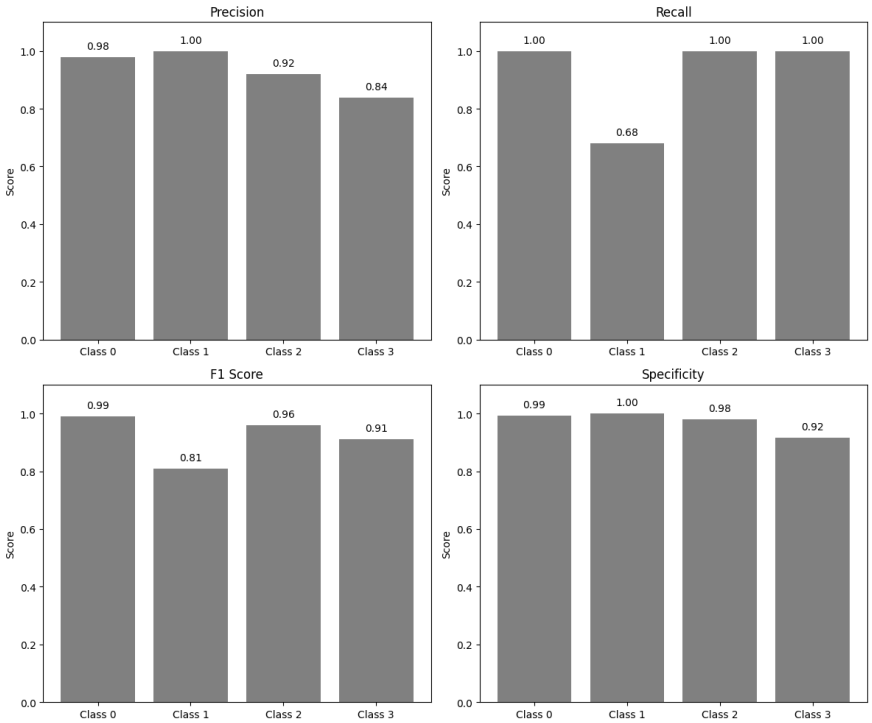

Figure 4.12 Performance metrics of CNN at 360 by 480

4.2.2 InceptionV3 Performance

This model was trained over 100 epochs using a dataset with 24 batches per epoch. The training process aimed to minimize the categorical crossentropy loss, resulting in an overall classification accuracy of 97.33% on test dataset and it took 408.13 seconds to complete the training process.

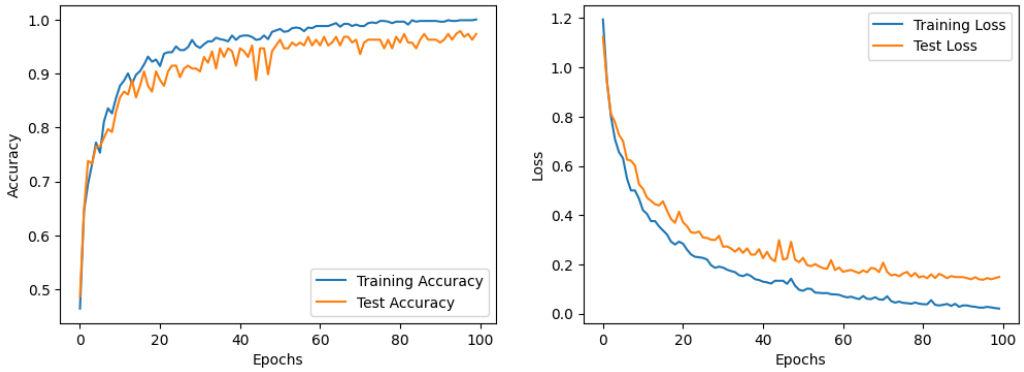

Figure 4.13 Performance curves of InceptionV3 at 360 by 480

The figure 4.14 provides a visual representation of various measures including recall, accuracy, F1 Score, and specificity, which provide a thorough knowledge of the model's overall performance.

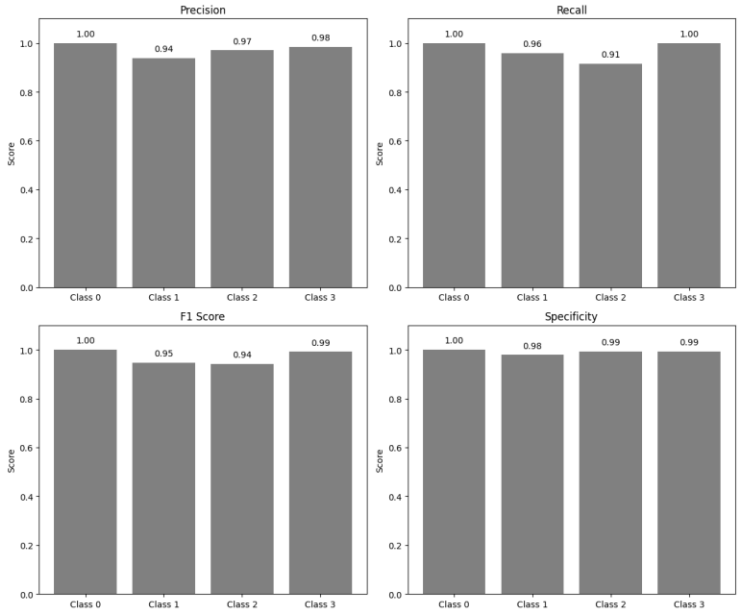

Figure 4.14 Performance metrics of InceptionV3 at 360 by 480

4.2.3 VGG19 Performance

This model was trained over 100 epochs using a dataset with 24 batches per epoch. The training process aimed to minimize the categorical crossentropy loss, resulting in an overall classification accuracy of 99.47% on test dataset and it took 581.38 seconds to complete the training process.

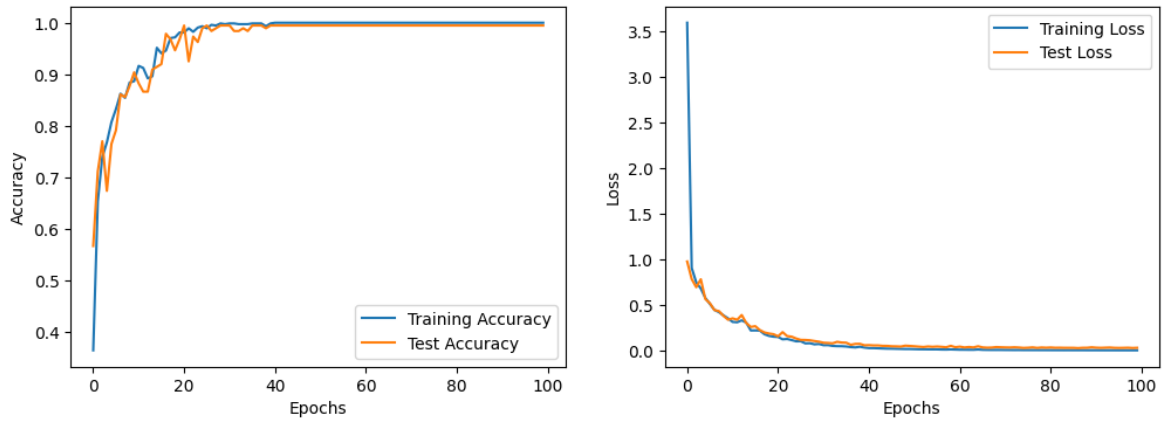

Figure 4.15 Performance curves of VGG19 at 360 by 480

The figure 4.16 provides a visual representation of various measures including recall, accuracy, F1 Score, and specificity, which provide a thorough knowledge of the model's overall performance.

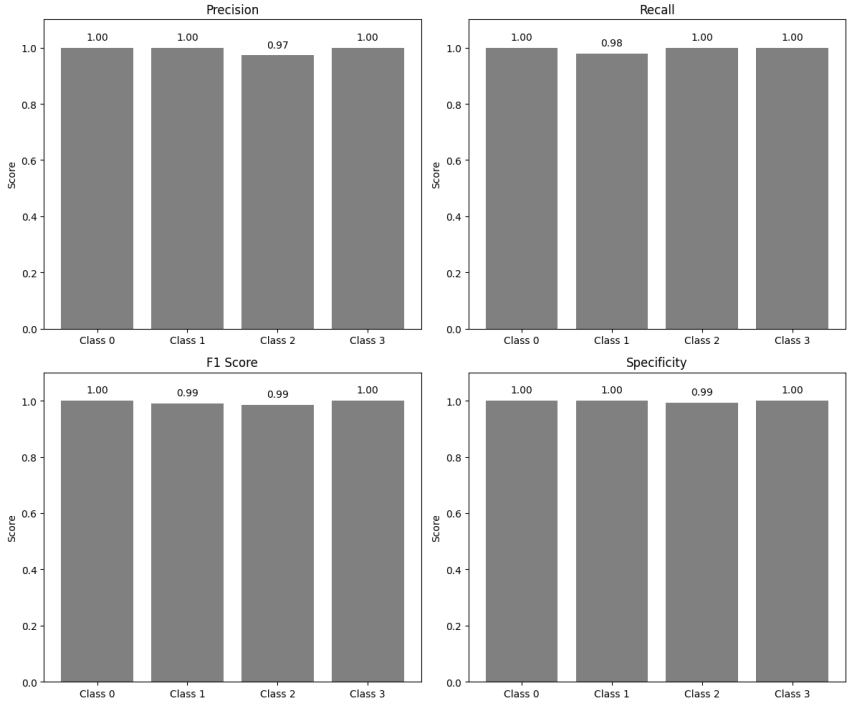

Figure 4.16 Performance metrics of VGG19 at 360 by 480

4.2.4 ResNet152 performance

This model was trained over 200 epochs using a dataset with 24 batches per epoch. The training process aimed to minimize the categorical crossentropy loss, resulting in an overall classification accuracy of 78.07% on test dataset and it took 1757.25 seconds to complete the training process.

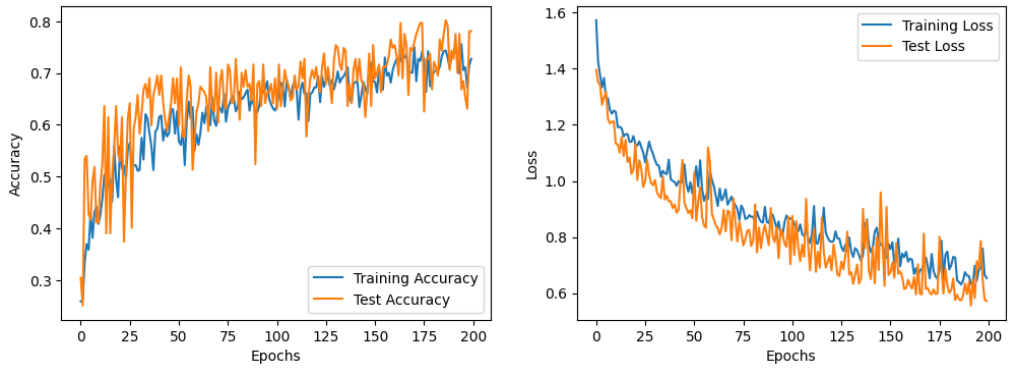

Figure 4.17 Performance curves of ResNet152 at 360 by 480

The figure 4.18 provides a visual representation of various measures including recall, accuracy, F1 Score, and specificity, which provide a thorough knowledge of the model's overall performance.

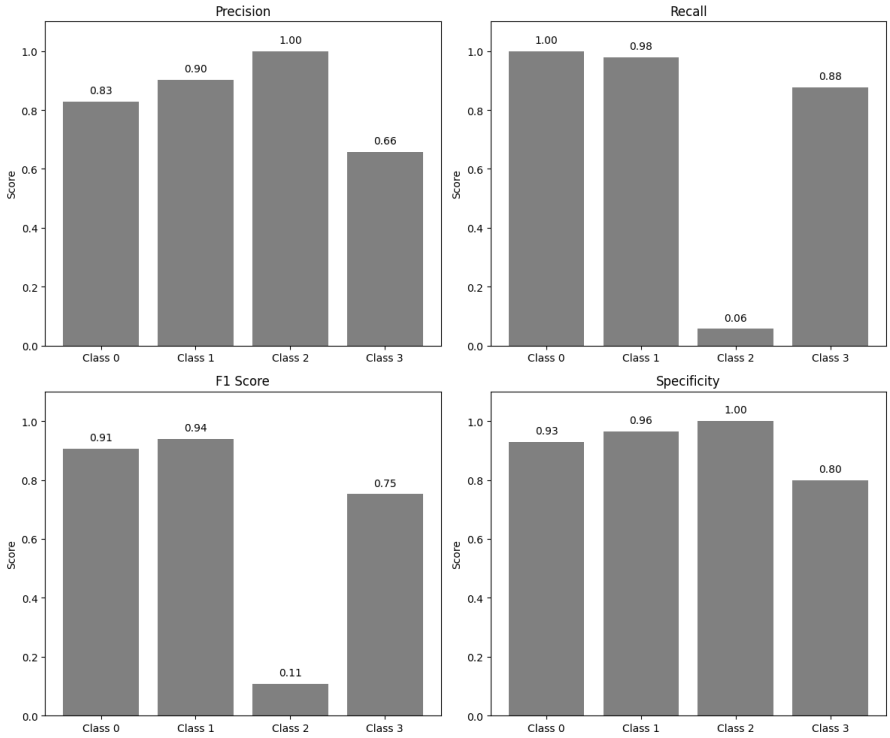

Figure 4.18 Performance metrics of ResNet152 at 360 by 480

4.2.5 ResNet50 Performance

This model was trained over 200 epochs using a dataset with 24 batches per epoch. The training process aimed to minimize the categorical crossentropy loss, resulting in an overall classification accuracy of 86.10% on test dataset and it took 888.94 seconds to complete the training process.

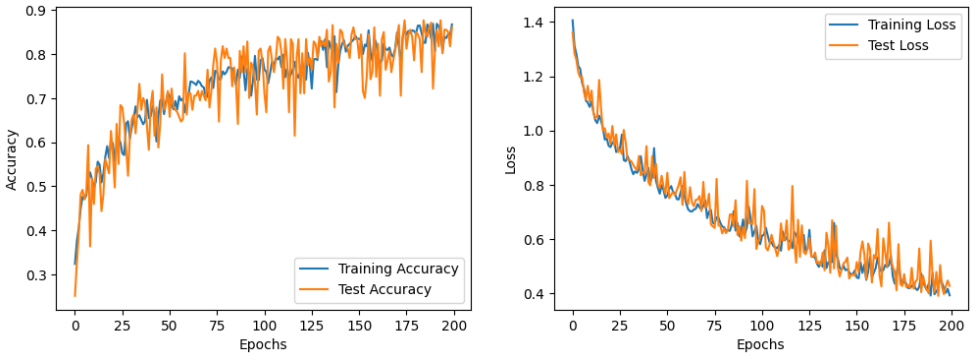

Figure 4.19 Performance curves of ResNet50 at 360 by 480

The figure 4.20 provides a visual representation of various measures including recall, accuracy, F1 Score, and specificity, which provide a thorough knowledge of the model's overall performance.

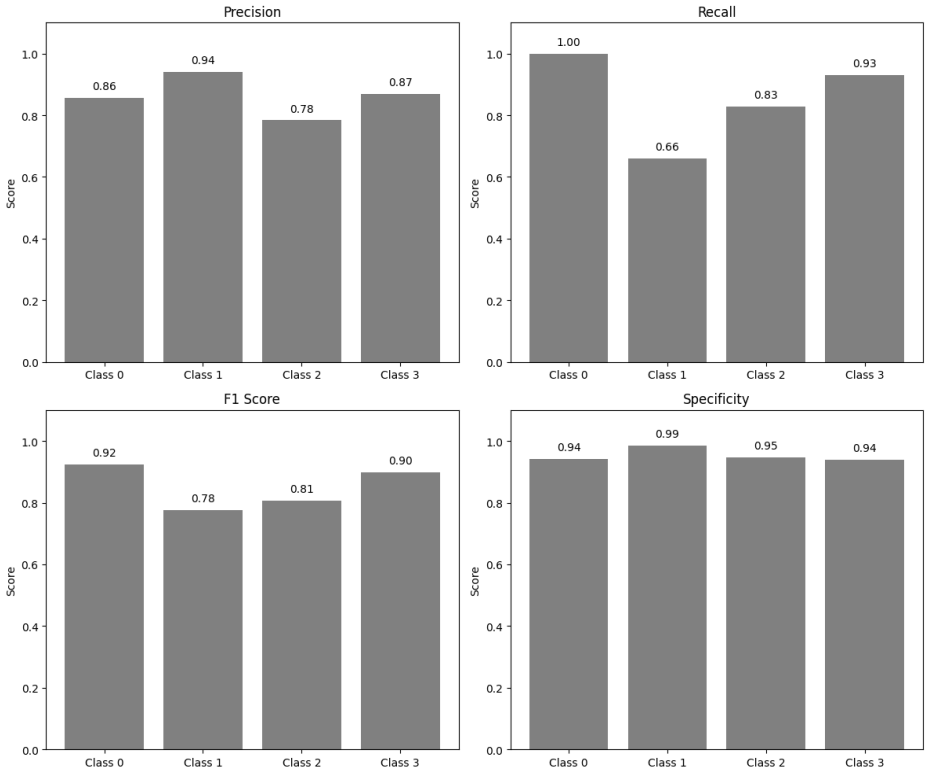

Figure 4.20 Performance metrics of ResNet50 at 360 by 480

4.3 Latent Feature Extraction (360 by 480 Pixels Images Reduced to 90 by 120 Pixels)

4.3.1 CNN Performance

This model was trained over 100 epochs using a dataset with 24 batches per epoch. The training process aimed to minimize the categorical crossentropy loss, resulting in an overall classification accuracy of 94.65% on test dataset and it took 44.02 seconds to complete the training process on latent features.

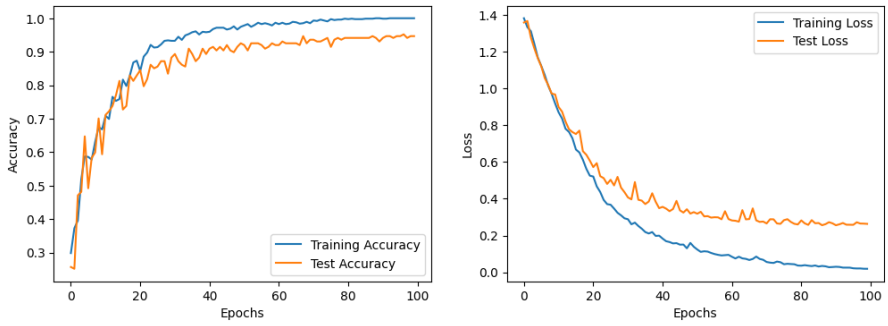

Figure 4.21 Performance curves of CNN (latent)

The figure 4.22 provides a visual representation of various measures including recall, accuracy, F1 Score, and specificity, which provide a thorough knowledge of the model's overall performance.

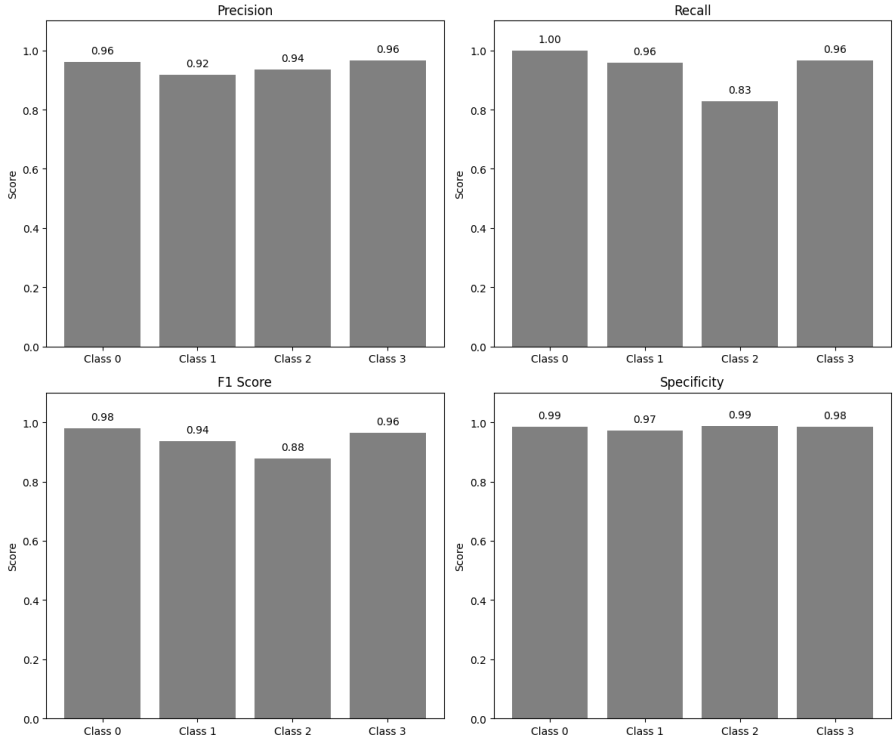

Figure 4.22 Performance metrics of CNN (latent)

4.3.2 InceptionV3 Performance

This model was trained over 100 epochs using a dataset with 24 batches per epoch. The training process aimed to minimize the categorical crossentropy loss, resulting in an overall classification accuracy of 93.05% on test dataset and it took 82.5 seconds to complete the training process on latent features.

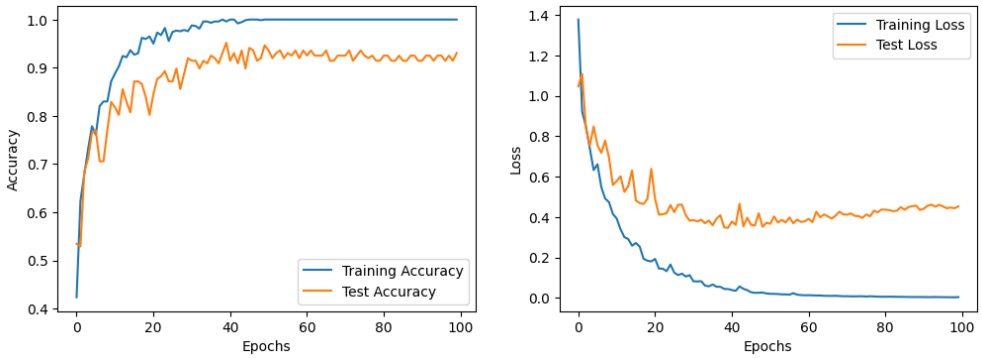

Figure 4.23 Performance curves of InceptionV3 (latent)

The figure 4.24 provides a visual representation of various measures including recall, accuracy, F1 Score, and specificity, which provide a thorough knowledge of the model's overall performance.

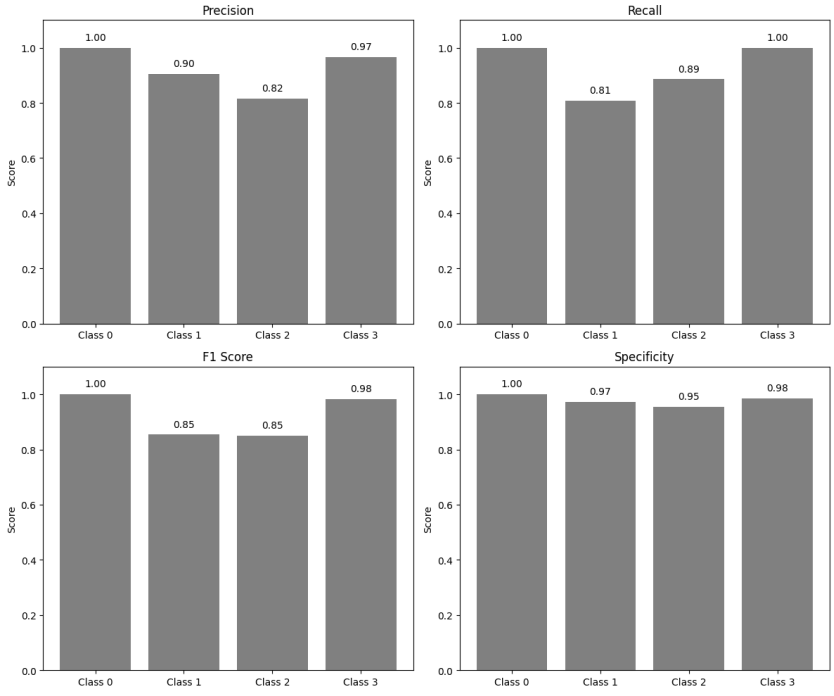

Figure 4.24 Performance metrics of InceptionV3 (latent)

4.3.3 VGG19 Performance

This model was trained over 100 epochs using a dataset with 24 batches per epoch. The training process aimed to minimize the categorical crossentropy loss, resulting in an overall classification accuracy of 94.65% on test dataset and it took 88.77 seconds to complete the training process on latent features.

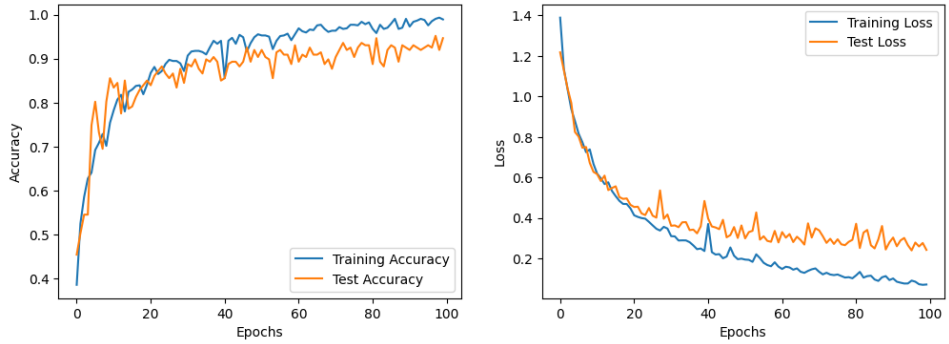

Figure 4.25 Performance curves of VGG19 (latent)

The figure 4.26 provides a visual representation of various measures including recall, accuracy, F1 Score, and specificity, which provide a thorough knowledge of the model's overall performance.

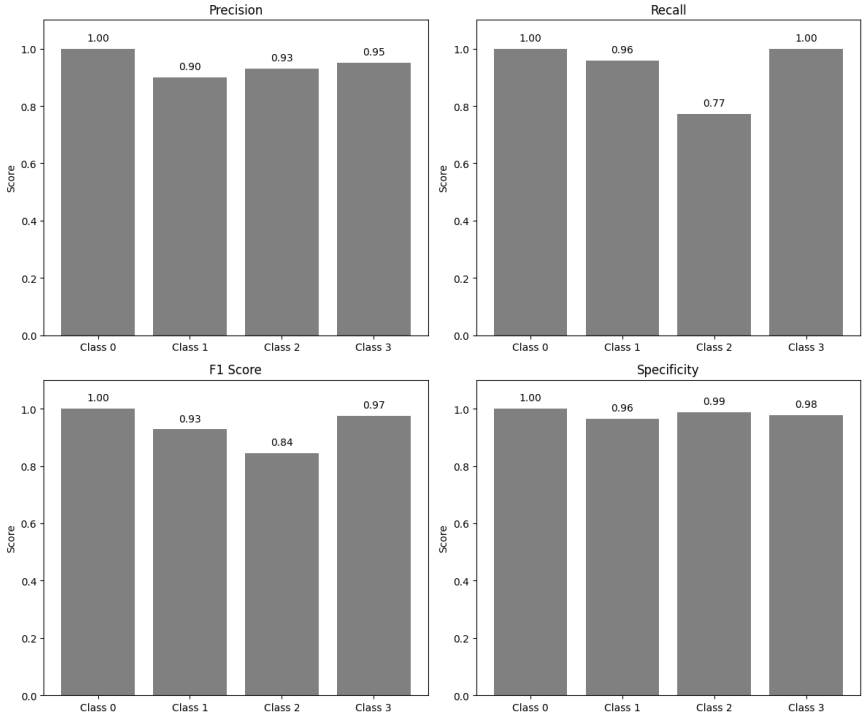

Figure 4.26 Performance metrics of VGG19 (latent)

4.3.4 ResNet152 Performance

This model was trained over 200 epochs using a dataset with 24 batches per epoch. The training process aimed to minimize the categorical crossentropy loss, resulting in an overall classification accuracy of 79.14% on test dataset and it took 293.23 seconds to complete the training process on latent features.

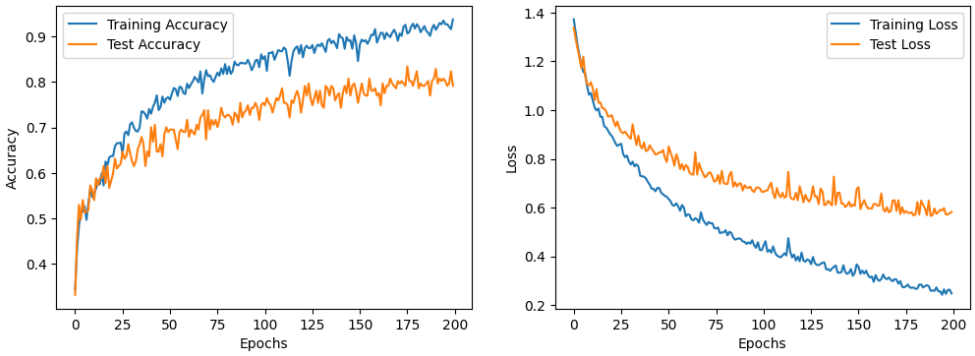

Figure 4.27 Performance curves of ResNet152 (latent)

The figure 4.28 provides a visual representation of various measures including recall, accuracy, F1 Score, and specificity, which provide a thorough knowledge of the model's overall performance.

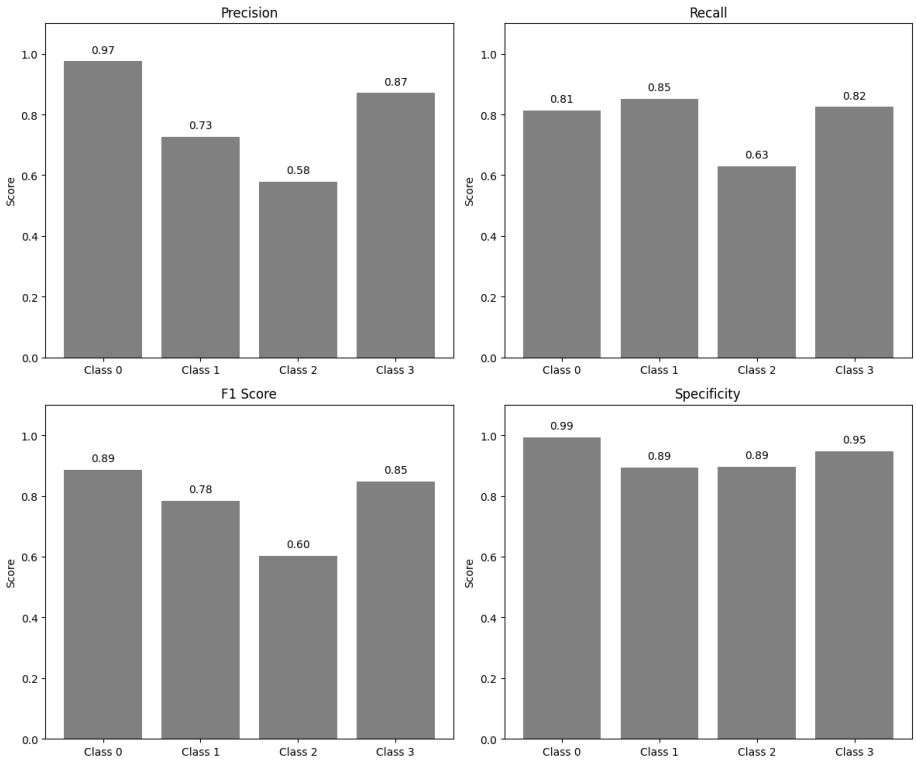

Figure 4.28 Performance metrics of ResNet152 (latent)

4.3.5 ResNet50 Performance

This model was trained over 200 epochs using a dataset with 24 batches per epoch. The training process aimed to minimize the categorical crossentropy loss, resulting in an overall classification accuracy of 82.35% on test dataset and it took 148.77 seconds to complete the training process on latent features.

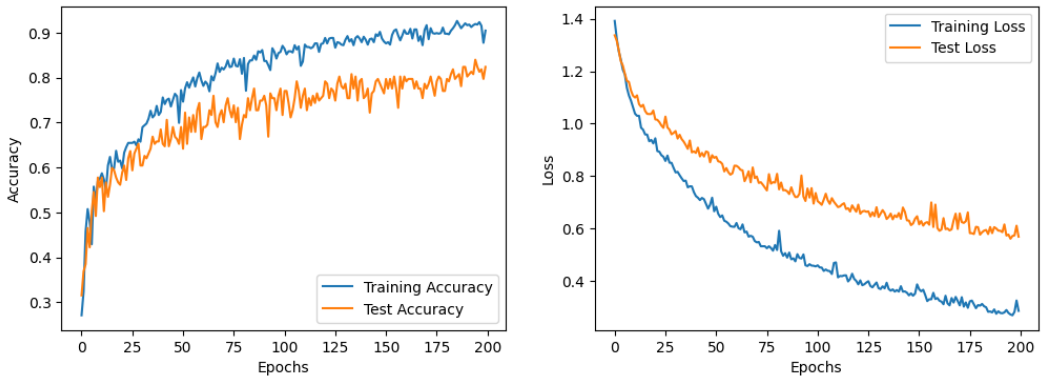

Figure 4.29 Performance curves of ResNet50 (latent)

The figure 4.30 provides a visual representation of various measures including recall, accuracy, F1 Score, and specificity, which provide a thorough knowledge of the model's overall performance.

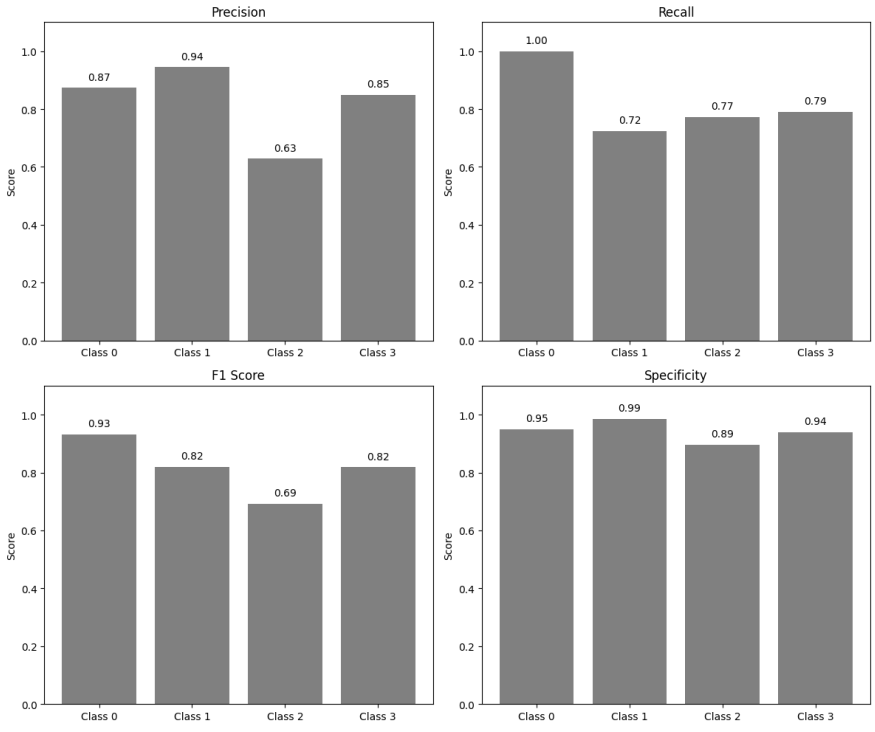

Figure 4.30 Performance metrics of ResNet50 (latent)

4.4 Comparison

The comparison of algorithms is made on the basis of accuracy, specificity and training time. This is as discussed in the following sections.

4.4.1 Accuracy

Accuracy is a key performance indicator in ECG image classification as it provides a broad sense of the model's overall performance across all classes. A high accuracy level indicates that most of the time the model is correctly predicting the situation. The accuracy comparison of several models for three resolutions is shown in the figure 4.31.

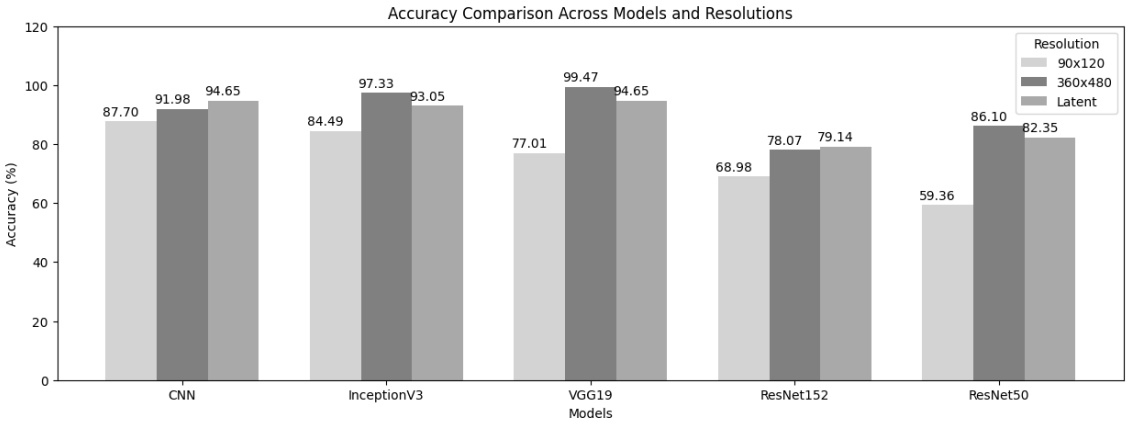

Figure 4.31 Accuracy comparison across models and resolutions

The resolution of the input images has a considerable impact on how well the models function.With an accuracy of 87.70%, the CNN model outperforms the other models at 90x120 resolution, while InceptionV3 comes in second with 84.49% accuracy. At this resolution, the ResNet50 model has the lowest accuracy, at 59.36%.

The VGG19 model performs better than all others at 360x480 resolution, with an astounding accuracy of 99.47%. At an accuracy of 86.10%, the ResNet50 model demonstrates a significant increase over its performance at the lower resolution.

Both the CNN and VGG19 models attain the maximum accuracy of 94.65% for the Latent 90x120 resolution. At 82.35% accuracy, the ResNet50 model likewise does rather well at this resolution.

In conclusion, the ResNet50 model exhibits notable performance variance based on the input image resolution, However the CNN and VGG19 models consistently perform well across all resolutions. It's also important to note that, at 360x480 resolution, the VGG19 model delivers the best overall accuracy. Nevertheless, additional aspects like training time and specificity should also be taken into account when selecting a model.

4.4.2 Specificity

Since specificity sheds light on the model's capacity to accurately choose the genuine negatives from among all the actual negatives, it is seen a crucial parameter, particularly in medical and diagnostic applications (Zhu et al., 2010). The specificity comparison of several models for three resolutions is shown in the figure 4.32.

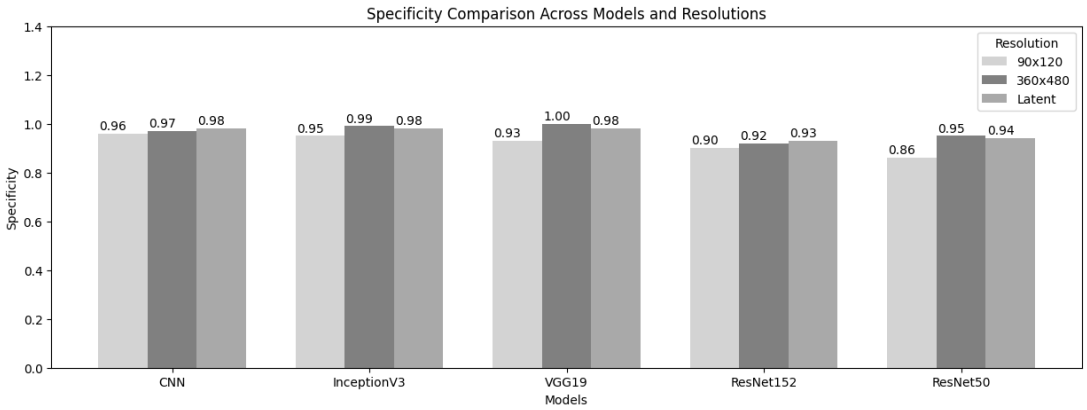

Figure 4.32 Specificity comparison across models and resolutions

The CNN model has the greatest average specificity of 0.96 for the 90x120 resolution, which suggests a high true negative rate. At this resolution, the ResNet50 model has the lowest average specificity of 0.86.

The VGG19 model gets the maximum average specificity of 1, indicating an exceptional true negative rate, when the resolution is raised to 360x480. At this resolution, the ResNet152 model has the lowest average specificity of 0.92.

The CNN and VGG19 models both obtain the maximum average specificity of 0.98 for the Latent 90x120 resolution. At this resolution, the ResNet152 model has the lowest average specificity of 0.93.

In conclusion, the ResNet50 and ResNet152 models exhibit notable performance variance based on the input image resolution, However the CNN and VGG19 models consistently perform well in terms of specificity across all resolutions. It's also important to note that, at 360x480 resolution, the VGG19 model delivers the best overall specificity.

4.4.3 Training time

The training time of all of the models across various resolutions is as shown in the following figure 4.33.

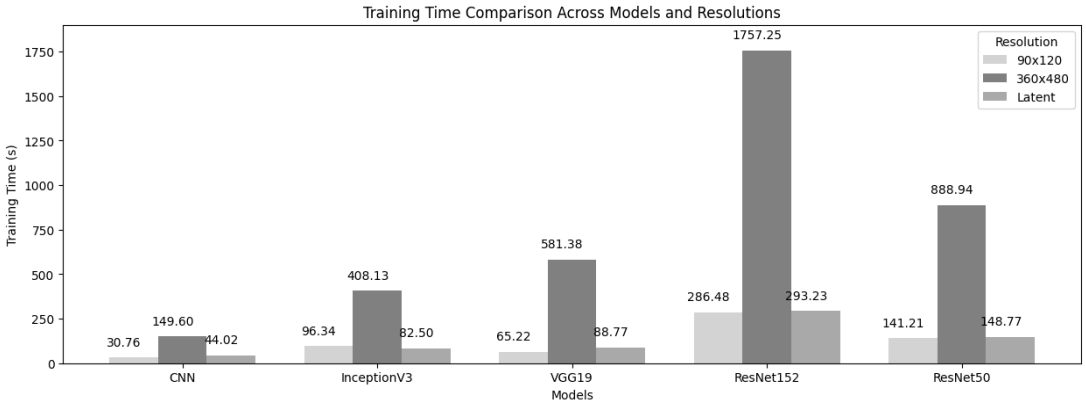

Figure 4.33 Training time comparison across models and resolutions

The CNN model trains in the quickest amount of time—30.76 seconds—for the 90x120 resolution, while the ResNet152 model takes the longest—286.48 seconds. At this resolution, the training time of the ResNet50 model is 141.21 seconds.

The CNN model still takes the least amount of time to train—149.6 seconds—when the resolution is raised to 360x480, while the ResNet152 model takes the longest—1757.25 seconds—among all the models and resolutions. At this resolution, the training time of the ResNet50 model is also longer—888.94 seconds.

The CNN model continues to have the quickest training time, at 44.02 seconds, for the Latent 90x120 resolution. Although the training duration of the ResNet152 model is significantly shorter at 293.23 seconds, it remains the longest of all the models at this resolution. At this resolution, the training time of the ResNet50 model is 148.77 seconds.

In conclusion, the ResNet152 model has the greatest training time, whereas the CNN model consistently has the smallest training time across all resolutions. The training time of the ResNet50 model is in the medium range.

5. Conclusion and Future work

With a particular focus on myocardial infarction diagnosis, this work explores the performance assessment of several deep learning models for automated electrocardiogram (ECG) image classification. The study evaluates important parameters at different resolutions, including training duration, specificity, and accuracy. The study clarifies the consistent performance of VGG19, the resolution-dependent behaviour of ResNet50, and the effectiveness of convolutional neural networks (CNNs), highlighting the complex relationship between image quality and model efficacy. The findings highlight the complex relationship that exists between model parameters, image resolution, and the possible advantages of using autoencoders to reduce dimensionality.

5.1 Summary of findings

The following are the main conclusions drawn from the study:

- Impact of Resolution: The models' performance is greatly affected by the resolution of the input images. When compared to lower quality images (90x120), higher resolution images (360x480) often result in improved accuracy and specificity across all models.
- Model Performance: Across all resolutions, VGG19 consistently exhibits good accuracy and specificity across the models. However it also takes the longest to train, particularly for images with greater quality.
- ResNet50 Insights: Depending on the resolution, the ResNet50 model's performance varies significantly. When compared to latent images and higher quality images (360x480), it performs quite badly on low-resolution images (90x120).
- Conclusions on Latent Resolution: Using latent dimensions or autoencoder features offers a decent balance between training time and accuracy. The models perform comparably to high-resolution images and often outperform low-resolution images when applied to latent images, which are 360x480 images reduced to 90x120 by the use of a variational autoencoder. When compared to high-resolution images, the training period is considerably shorter.
- CNN Effectiveness: With consistently the quickest training time across all resolutions, the CNN model is very efficient. It also performs admirably in terms of specificity and accuracy, particularly on latent images.

These findings highlight the importance of considering both the resolution of the input images and the specific characteristics of the models when conducting image classification tasks. They also demonstrate the potential benefits of using autoencoders to reduce the dimensionality of the input images.

5.2 Future work

There are a number of directions that this field of automated ECG image classification for myocardial infarction (MI) detection might go in the future. Investigating more sophisticated dimensionality reduction methods like Principal Component Analysis (PCA) and t-Distributed Stochastic Neighbor Embedding (t-SNE) outside of convolutional variational autoencoders (CVAEs) may improve the models' effectiveness. Preprocessing might be made even more efficient by looking at hybrid approaches or the possible inclusion of additional unsupervised learning techniques.

Second, it could be worthwhile to look at how model performance is affected by hyperparameter adjustment in the future. Training time and accuracy may be increased by fine-tuning the deep learning models' parameters, such as changing learning rates or optimizing model topologies. This may be especially important for models such as VGG19, which although regularly delivering good results, may benefit from optimizations to shorten training times without sacrificing accuracy.

Furthermore, expanding the study to include a wider variety of ECG data—including a range of patient demographics and clinical conditions—would improve the suggested system's generalizability. This extension could offer a more thorough comprehension of the models' performance in various settings, enhancing the automated MI detection system's robustness and dependability. To sum up, in order to conduct a more thorough assessment of automated ECG classification systems, future research should focus on improving dimensionality reduction methods, optimising model parameters, and broadening dataset diversity.

References

Acharya, U. R., Fujita, H., Lih, O. S., Hagiwara, Y., Tan, J. H. and Adam, M. (2017) 'Automated detection of arrhythmias using different intervals of tachycardia ECG segments with convolutional neural network', *Information sciences,* 405, pp. 81-90.

Brady, W. J., Aufderheide, T. P., Chan, T. and Perron, A. D. (2001) 'Electrocardiographic diagnosis of acute myocardial infarction', *Emergency Medicine Clinics,* 19(2), pp. 295-320.

Chauhan, S. and Vig, L. (2015) 'Anomaly detection in ECG time signals via deep long short-term memory networks'. *2015 IEEE international conference on data science and advanced analytics (DSAA)*: IEEE, 1-7.

Çöl, P. E. and Ertekin, Ş. (2021) 'Feature dimensionality reduction with variational autoencoders in deep bayesian active learning'. *2021 29th Signal Processing and Communications Applications Conference (SIU)*: IEEE, 1-4.

Dar, S. A. and Palanivel, S. (2020) 'Deep variational auto encoder for dimensionality reduction, denoising in MNIST datasets using TensorFlow and keras', *Tech Trends 2021: Issues and Emerging Challenges and Changes in the Student-Centric Learning and Best Innovative Practices for Quality Enhancement in Education*, pp. 218.

Dustakar, S. R., Rao, L. K. and Vipparthi, B. (2023) 'An Automated Medical Image Segmentation Framework using Deep Learning and Variational Autoencoders with Conditional Neural Networks', *International Journal of Advanced Computer Science and Applications,* 14(8).

Gupta, A., Huerta, E., Zhao, Z. and Moussa, I. (2021) 'Deep learning for cardiologist-level myocardial infarction detection in electrocardiograms'. *8th European Medical and Biological Engineering Conference: Proceedings of the EMBEC 2020, November 29–December 3, 2020 Portorož, Slovenia*: Springer, 341-355.

Hadiyoso, S., Aulia, S. and Irawati, I. D. (2023) 'Multi-Class Heart Abnormalities Detection Based on ECG Graph Using Transfer Learning Method', *Jurnal Rekayasa Elektrika,* 19(1).

Hadiyoso, S., Fahrozi, F., Hariyani, Y. S. and Sulistiyo, M. D. (2022) 'Image Based ECG Signal Classification Using Convolutional Neural Network', *International Journal of Online & Biomedical Engineering,* 16(4).

Haroon, M. A. (2020) 'ECG arrhythmia classification Using deep convolution neural networks in transfer learning'.

Hasbullah, S., Mohd Zahid, M. S. and Mandala, S. (2023) 'Detection of Myocardial Infarction Using Hybrid Models of Convolutional Neural Network and Recurrent Neural Network', *BioMedInformatics,* 3(2), pp. 478-492.

Hawkins, D. M. (2004) 'The problem of overfitting', *Journal of chemical information and computer sciences,* 44(1), pp. 1-12.

Hemanth, D. J. (2021) 'Automated feature extraction in deep learning models: A boon or a bane?'. *2021 8th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI)*: IEEE, 3-3.

Humayun, M., Sujatha, R., Almuayqil, S. N. and Jhanjhi, N. (2022) 'A transfer learning approach with a convolutional neural network for the classification of lung carcinoma'. *Healthcare*: MDPI, 1058.

Khan, A. H. and Hussain, M. (2021) 'ECG images dataset of cardiac patients', *Mendeley Data*, 2, pp. 2021.

Kligfield, P., Gettes, L. S., Bailey, J. J., Childers, R., Deal, B. J., Hancock, E. W., Van Herpen, G., Kors, J. A., Macfarlane, P. and Mirvis, D. M. (2007) 'Recommendations for the standardization and interpretation of the electrocardiogram: part I: the electrocardiogram and its technology: a scientific statement from the American Heart Association Electrocardiography and Arrhythmias Committee, Council on Clinical Cardiology; the American College of Cardiology Foundation; and the Heart Rhythm Society endorsed by the International Society for Computerized Electrocardiology', *Circulation,* 115(10), pp. 1306-1324.

Labounty, T. and Eagle, K. A. (2007) 'The nature of the problem: an overview of acute coronary syndromes and myocardial infarction', *Biological Rhythm Research,* 38(3), pp. 143-153.

Liu, W., Zhang, M., Zhang, Y., Liao, Y., Huang, Q., Chang, S., Wang, H. and He, J. (2017) 'Real-time multilead convolutional neural network for myocardial infarction detection', *IEEE journal of biomedical and health informatics,* 22(5), pp. 1434-1444.

Manimekalai, K. and Kavitha, D. (2020) 'Deep learning methods in classification of myocardial infarction by employing ECG signals', *Indian J Sci Technol,* 13, pp. 2823-32.

Michael, M. A., El Masry, H., Khan, B. R. and Das, M. K. (2007) 'Electrocardiographic signs of remote myocardial infarction', *Progress in cardiovascular diseases,* 50(3), pp. 198-208.

Narin, A., Kaya, C. and Pamuk, Z. (2021) 'Automatic detection of coronavirus disease (covid-19) using x-ray images and deep convolutional neural networks', *Pattern Analysis and Applications,* 24, pp. 1207-1220.

Ogrezeanu, I., Stoian, D., Turcea, A. and Itu, L. M. (2020) 'Deep learning based myocardial ischemia detection in ECG signals'. *2020 24th International Conference on System Theory, Control and Computing (ICSTCC)*: IEEE, 250-253.

Osowski, S., Hoai, L. T. and Markiewicz, T. (2004) 'Support vector machine-based expert system for reliable heartbeat recognition', *IEEE transactions on biomedical engineering,* 51(4), pp. 582-589.

Parvaneh, S. and Rubin, J. (2018) 'Electrocardiogram monitoring and interpretation: from traditional machine learning to deep learning, and their combination'. *2018 Computing in Cardiology Conference (CinC)*: IEEE, 1-4.

Rai, H. M. and Chatterjee, K. (2022) 'Hybrid CNN-LSTM deep learning model and ensemble technique for automatic detection of myocardial infarction using big ECG data', *Applied Intelligence*, 52(5), pp. 5366-5384.

Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D., Bagul, A., Langlotz, C. and Shpanskaya, K. (2017) 'Chexnet: Radiologist-level pneumonia detection on chest x-rays with deep learning', *arXiv preprint arXiv:1711.05225*.

Ram, R. S., Akilandeswari, J. and Kumar, M. V. (2023) 'HybDeepNet: A Hybrid Deep Learning Model for Detecting Cardiac Arrhythmia from ECG Signals', *Information Technology & Control*, 52(2).

Rawi, A. A., Albashir, M. K. and Ahmed, A. M. (2022) 'Classification and detection of ECG arrhythmia and myocardial infarction using deep learning: a review', *Webology*, 19(1), pp. 1151-70.

Ribeiro, M. T., Singh, S. and Guestrin, C. (2016) '" Why should i trust you?" Explaining the predictions of any classifier'. *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, 1135-1144.

Schröer, C., Kruse, F. and Gómez, J. M. (2021) 'A systematic literature review on applying CRISP-DM process model', *Procedia Computer Science*, 181, pp. 526-534.

Sivakumar, D. (2023) *Introduction to InceptionNet*. Available at: https://www.scaler.com/topics/inception-network/ (Accessed: 09-November-2023 2023).

Sraitih, M. and Jabrane, Y. (2022) 'A survey of deep learning approaches for classifying ECG heartbeat arrhythmias'. *2022 International Conference on Intelligent Systems and Computer Vision (ISCV)*: IEEE, 1-8.

Stracina, T., Ronzhina, M., Redina, R. and Novakova, M. (2022) 'Golden standard or obsolete method? Review of ECG applications in clinical and experimental context', *Frontiers in Physiology*, 13, pp. 867033.

Topol, E. J. (2019) 'High-performance medicine: the convergence of human and artificial intelligence', *Nature medicine*, 25(1), pp. 44-56.

Ukil, A., Marín, L. and Jara, A. J. (2021) 'L1 and L2 Regularized Deep Residual Network Model for Automated Detection of Myocardial Infarction (Heart Attack) Using Electrocardiogram Signals'. *CIKM Workshops*.

Wang, Y., Yao, H. and Zhao, S. (2016) 'Auto-encoder based dimensionality reduction', *Neurocomputing*, 184, pp. 232-242.

Xiong, P., Lee, S. M.-Y. and Chan, G. (2022) 'Deep learning for detecting and locating myocardial infarction by electrocardiogram: A literature review', *Frontiers in cardiovascular medicine*, 9, pp. 860032.

Zhu, W., Zeng, N. and Wang, N. (2010) 'Sensitivity, specificity, accuracy, associated confidence interval and ROC analysis with practical SAS implementations', *NESUG proceedings: health care and life sciences, Baltimore, Maryland*, 19, pp. 67.

Appendices

In support of this research, the following appendices provide additional details and resources for comprehensive understanding.

Dataset Information

The dataset images utilized in this study are sourced from the Ch. Pervaiz Elahi Institute of Cardiology which is available at Mendeley data website. The dataset is under Creative Commons 4.0 license and the link to the dataset is as shown below.

https://data.mendeley.com/datasets/gwbz3fsgp8/1

Python implementation files

A series of Jupyter Notebook files (.ipynb) are included in the artifacts to offer a comprehensive view of the code implementation process throughout the research. The code has been saved in this format so that all of the outputs can be viewed. The list of code implementation files is as shown below.

1. All models_with_VAE_DimensionalityReduction.ipynb
2. CNN(90by120).ipynb
3. CNN(360by480).ipynb
4. Inception(90by120).ipynb
5. Inception(360by480).ipynb
6. RESTNET50(90by120).ipynb
7. RESTNET50(360by480).ipynb
8. RESTNET152(90by120).ipynb
9. RESTNET152(360by480).ipynb
10. VGG(90by120).ipynb
11. VGG(360by480).ipynb
12. Performance.ipynb