ECG Interpretation with Machine Learning

Submitted by: Diya Patel
Student ID: 20115612

Supervisor: Andrew Lowe

School of Engineering, Computer and Mathematical Science

A final year project report presented to the Auckland University of Technology in partial fulfilment of the requirements of the degree of Bachelor of Engineering
2023

Acknowledgements

I want to thank my supervisor, Professor Andrew Lowe, for the constant guidance, encouragement, and support that you have provided me throughout this project.

I want to acknowledge my friends for the help and encouragement you provided me while working on this project.

Lastly, I would like to thank my family for supporting me through the process of this project and my degree.

Statement of Originality

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgements), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.

Diya Patel

......01/08/2024...... ....

Date Diya Patel

Abstract

The electrocardiogram (ECG) is a crucial diagnostic tool that records the heart's electrical activity through waveforms and pulses, analyzed by medical experts to assess cardiac health. Recent advancements in healthcare, driven by technological innovation, have enabled the automation and enhancement of ECG monitoring, analysis, and interpretation. To contribute to this progress, an extensive review of state-of-the-art techniques in ECG analysis was conducted, focusing on machine learning and deep learning approaches for automating ECG signal interpretation and anomaly detection. Based on these insights, a Convolutional Neural Network (CNN) model was developed, achieving 100% accuracy in classifying both normal and arrhythmic ECG signals. This model demonstrates the significant potential of deep learning in revolutionizing ECG analysis, facilitating more accurate and timely diagnoses by medical professionals, which can ultimately improve patient outcomes and reduce the burden on healthcare systems. The study also highlights the importance of further validation to ensure the robustness and generalizability of the model in clinical settings.

Keywords: Electrocardiograms (ECGs), Convolutional Neural Network (CNN), cardiac health, deep learning, ECG signal analysis

Acronyms

Afib	Atrial Fibrillation
ECG	Electrocardiogram
CNN	Convolution Neural Network
SAN	Sinoatrial Node
AVN	Atrioventricular Node
STFT	Short-Time Fourier Transform
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
MLP	Multilayer Perceptron
AI	Artificial Intelligence

Table of Contents

Acknowledgements 2

Statement of Originality 3

Abstract 4

Acronyms 5

List of Figures 8

Chapter 1 Introduction 9

1.1 Introduction 9

1.2 Scope and Limitations 9

Chapter 2 Theory 10

2.1 The Heart and Conductive System 10

2.1.1 Introduction to ECG (Electrocardiography): 10

2.1.2 Cardiac Conduction 10

2.1.3 ECG Waves 11

2.1.4 Arrhythmia 12

2.2 ECG Instrumentation 13

2.2.1 ECG instruments 13

2.2.2 Challenges in Manual ECG Interpretation 13

2.3 Machine Learning 14

2.3.1 Introduction to Machine Learning: 14

2.3.2 Machine Learning in Healthcare: 14

2.3.3 Deep Learning with Neural Networks 14

2.3.4 Training Neural Networks 14

2.3.5 Time series to image 15

Chapter 3 Literature Review 16

3.1 Background 16

3.1.1 Atrial Fibrillation 16

3.1.2 Machine learning 16

3.2 Research Papers 16

3.2.1 Deep Learning for ECG Interpretation 17

3.2.2 Convolution Neural Networks 17

3.2.3 Time Series to Images 18

3.2.4 Convolutional Neural Networks (CNNs) for image-based classification........ 18

3.3 Case Study 19

3.3.1 Convolutional Neural Network........ 19

3.3.2 Image-based time series forecasting: A deep convolutional neural network approach 23

Chapter 4 Methodology........ 27

4.1 Project Execution Plan 27

4.2 Resource Analysis 29

4.3 Risk Assessment 30

Chapter 5 Results 31

5.1 1D CNN with Time Series ECG and Arrhythmia Data 31

5.2 1D CNN with Spectrogram ECG and Arrhythmia Data 32

5.3 Code and Model Limitations 33

5.4 Evaluation of Results in Relation to Project Goals 34

Chapter 6 Conclusion and Future Work........ 34

References 36

List of Figures

Figure 1 Parts of the ECG 11
Figure 2 Arrhythmia 12
Figure 3 Example of image-based CNN algorithm 18
Figure 4 Receptive field of particular neuron in the next layer 20
Figure 5 Pooling operation performed by choosing a 2 x 2 window 20
Figure 6 Architecture of LeNet5, a CNN where each box represents a different feature map.... 21
Figure 7 AlexNet architecture 21
Figure 8 Overview of the proposed image-based time series forecasting method, ForCNN, consisting of an encoder and a regressor module. 24
Figure 9 ForCNN-SD architecture. Top: Stacks create latent representations; FC layers make h-step-ahead forecasts. Middle: Stacks have convolutional blocks ending with Conv2D. Bottom: Blocks include Conv2D, Batch Norm, ReLU, and shortcuts. 25
Figure 10 CNN methodology 28
Figure 11 Gantt chart Part A 29
Figure 12 Gantt chart Part B 29
Figure 13 Risk assessment table 30
Figure 14 Time series results 31
Figure 15 Time series confusion matrix 32
Figure 16 Spectrogram results 33
Figure 17 Spectrogram confusion matrix 33

Chapter 1 Introduction

1.1 Introduction

Electrocardiogram (ECG) interpretation is a critical component of cardiac healthcare, serving as a fundamental diagnostic tool for assessing the heart's electrical activity. The wave patterns and intervals observed in ECG signals provide valuable insights into various cardiac conditions, such as arrhythmias, myocardial infarction, and other heart disorders. However, the manual interpretation of ECGs presents several challenges, including interobserver variability, human error, and the need for specialized expertise. These limitations can lead to diagnostic inconsistencies and delays in treatment.

Recent advancements in healthcare technology, particularly in machine learning, have introduced new possibilities for automating and improving ECG analysis. Machine learning algorithms, especially deep learning models, offer the ability to learn from vast amounts of data, enabling them to identify patterns and abnormalities that might be overlooked by human experts. This fusion of medical expertise with computational intelligence has the potential to revolutionize ECG interpretation, reducing the time and effort required for diagnosis, while simultaneously improving accuracy and patient outcomes.

This project aims to explore the integration of machine learning with ECG interpretation, specifically focusing on the use of Convolutional Neural Networks (CNNs). By investigating the feasibility and effectiveness of CNNs in automating ECG classification, this research contributes to the ongoing transformation of cardiac healthcare. Through a comprehensive examination of the underlying theory, a review of relevant literature, and the development of CNN-based models, this project bridges the gap between traditional diagnostic practices and cutting-edge technological innovations, providing insights into the future of automated ECG analysis and its impact on patient care.

1.2 Scope and Limitations

This project focuses on key areas of ECG monitoring and interpretation, while acknowledging that the field is vast and multifaceted. The following objectives define the project's scope:

1. Feature Extraction: The initial focus is on identifying and extracting key features from ECG signals that are crucial for subsequent classification tasks. The project will prioritize the extraction of clinically significant features that can aid in accurate ECG interpretation.
2. Denoising Techniques: The quality of ECG data is paramount to achieving accurate analysis. The project will explore and implement techniques to remove noise artifacts from ECG signals, while preserving the integrity of the underlying waveform. Improved signal quality is expected to enhance the overall accuracy of classification models.
3. Transformation of ECG Time Series into Image Representations: An essential aspect of this project involves transforming ECG time-series data into image representations, such as spectrograms, which can be processed by CNNs. This transformation is intended to enhance the algorithm's ability to discern subtle patterns in the data, potentially improving classification accuracy and model interpretability.
4. Machine Learning Algorithm Implementation: CNNs will be developed and implemented using Python for robust ECG classification. The selection, tuning, and optimization of CNN models are critical to achieving high classification performance. The project will

focus on selecting appropriate architectures and fine-tuning hyperparameters to maximize the effectiveness of these models in ECG analysis.

While the field of "ECG Interpretation" encompasses a wide range of topics, this project focuses on key foundational elements and offers a high-level overview of what can be explored. In-depth investigation into specialized topics such as real-time ECG monitoring, multi-class classification, and domain adaptation is beyond the scope of this project. Additionally, factors such as data quality, sample size, and model complexity may pose limitations on the performance of the machine learning algorithms. These constraints should be considered when interpreting the results.

Chapter 2 Theory

2.1 The Heart and Conductive System

2.1.1 Introduction to ECG (Electrocardiography):

[1] Electrocardiography (ECG) is a non-invasive diagnostic tool used to assess the electrical activity of the heart. It involves recording the electrical impulses generated by the heart as it contracts and relaxes. ECG is essential in clinical practice for diagnosing various cardiac abnormalities, including arrhythmias, ischemia, myocardial infarction, and conduction disorders. By analysing the patterns and intervals of the ECG waveform, healthcare professionals gain valuable insights into the heart's health and function.

2.1.2 Cardiac Conduction

[2] The cardiac conduction system is a network of specialized cells that coordinates electrical impulses responsible for regulating the heartbeat. It ensures synchronized contraction and relaxation of the heart chambers, facilitating efficient blood pumping throughout the body. The key components of the cardiac conduction system include:

1. Sinoatrial Node (SAN): The heart's natural pacemaker, generating electrical impulses to initiate atrial contraction.
2. Atrioventricular Node (AVN): Transmits impulses from the atria to the ventricles, facilitating coordinated contraction.
3. Bundle of His: Conducts impulses from the AVN to the ventricles through bundle branches, ensuring efficient signal transmission.
4. Purkinje Fibers: Rapidly transmit electrical signals for synchronized ventricular contraction and effective blood pumping.

2.1.3 ECG Waves

[3] In terms of clinical significance and diagnostic value, as shown in figure 1, the most important waves in an electrocardiogram (ECG) are typically considered to be:

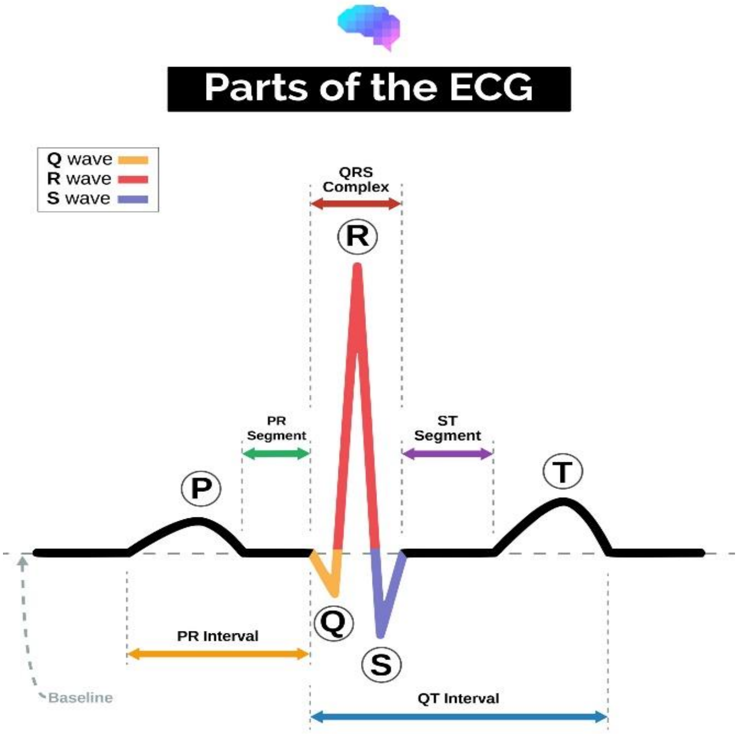


Figure 1 Parts of the ECG

1. The P wave, the first waveform in an ECG, depicts the initial electrical pulse from the SA node propagating across the atria to depolarize myocardial cells for contraction and blood influx. Although smaller in amplitude compared to the QRS complex and T wave, abnormalities in the P wave can indicate atrial enlargement, atrial fibrillation, or other atrial arrhythmias. The end of the P wave marks the point where the pulse traverses through the Atrioventricular Node (AV Node), signifying the beginning of the diastole phase.

2. QRS Complex: The QRS complex represents ventricular depolarization, initiating ventricular contraction and serving as a vital indicator for diagnosing abnormalities in ventricular conduction, such as bundle branch blocks, ventricular hypertrophy, and myocardial infarction (heart attack). This waveform starts after the PR segment, signifying the departure of the pulse from the AVN and the commencement of ventricular depolarization through the His-Purkinje system. With greater magnitude on ECGs due to the larger size of the ventricles compared to the atria, the QRS complex marks the systole phase, essential for the effective pumping of blood out of the heart.

3. The ST segment and T wave together signify the repolarization phase of both the ventricles and atria in the cardiac cycle. The ST segment is crucial for identifying myocardial ischemia or injury, with elevation or depression indicating conditions like myocardial infarction, ischemia, or pericarditis. Similarly, the T wave represents ventricular repolarization, and changes in its morphology or amplitude can signal electrolyte imbalances, myocardial ischemia, infarction, or other cardiac

abnormalities. Together, these components provide critical information for diagnosing and managing various cardiac conditions.

2.1.4 Arrhythmia

The combination of all these waves makes up one heartbeat and repeats in a consistent fashion to make what is called Normal Sinus Rhythm. Arrhythmias are heartbeat rhythms that are different to the normal sinus rhythm we expect to see, as shown in figure 2. There are several different arrhythmias categorized by their cause, place of origin and phenomena, not all of which are medically concerning.

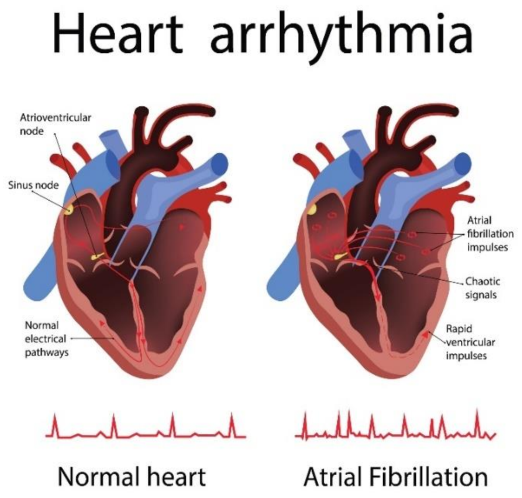


Figure 2 Arrhythmia

Places of Origin:
- SA Node
- Atria
- AV Junction
- Ventricles

Phenomena:
- Tachycardia (>100 bpm)
- Bradycardia (<60 bpm)
- Premature Contractions
- Flutter
- Fibrillation

Atrial Fibrillation (Afib):
- Common type of arrhythmia characterized by asynchronous and erratic atrial contractions.
- Often caused by reentry mechanism.
- ECG shows undiscernible P waves (F-waves) and irregular R-R intervals.
- Associated with risks such as blood clots leading to strokes, ischemic heart disease, hypertension, pulmonary disease, and rheumatic heart disease.
- Symptoms include heart palpitations, shortness of breath, weakness, but can also be asymptomatic.
- Early detection is crucial due to associated risks.

2.2 ECG Instrumentation

2.2.1 ECG instruments

Electrodes: Contacts made between the skin and the electrical readout circuit.
Wet Electrodes: Wet electrodes are the most common kind of electrodes used in hospitals. These gel patches are applied to the skin and contain a tiny metal contact covered with electrolytic paste. To achieve galvanic coupling, this forms an electrolyte-electrode contact that transforms ionic current on the skin's surface into electrical current. After that, the patches are connected to conductive copper wires leading to the rest of the circuit. Its key benefits are its low electrode impedance and easier disposal and use for hygienic and financial reasons.
Dry Electrodes: Instead of using a gel, dry electrodes rely on capacitive coupling to isolate the metal contact from the skin. This is achieved by coating the metal contact with a thin layer of insulating oxide or a film with a high dielectric constant. Since there is no need for skin preparation, this is more practical for a personal biopotential collecting device that is portable.

Leads: The location of the electrodes on the body gives different signals.
12-Leads: From 10 electrodes, we can derive 12 signals characterizing the heart's activity. On the left arm, left leg, and right arm, three electrodes are positioned. Lead I, II, and III are the three bipolar leads, and the three unipolar leads (lead aVR, aVL, and aVF) that result from this create six limb leads. The remaining six leads (V1–V6) are unipolar and come from six electrodes that are positioned on the chest.
Single-Lead: Two electrodes, one on the left and one on the right arm, can be used to generate a single lead, lead I in a 12-lead configuration. For convenience, handheld ECG monitors frequently employ this configuration.

Amplifier: Used to amplify the biopotential signal from the electrodes.
Instrumentation Amplifiers: The most used type of amplifier for testing and measurement is an instrumentation amplifier. The gain, which is typically controlled by a single external resistor, is the amount by which the signal is amplified. High gain has the drawback of magnifying even noise, but it can provide us a larger picture to work with. The Common-Mode Rejection Ratio, which reduces Power Line Interference, is another crucial feature.

ADC: Digitizes the analogue signal from the amplifier for processing.
Microcontrollers: Due to their on-board ADC and transmitter for transmitting the digital signal to a computer, where software may be used to process and graphically display the ECG signal, microcontrollers/processors are frequently utilized. The quality of the conversion and the Voltage Reference are determined by the resolution of an ADC. To achieve maximum accuracy and detail in the digitized signal, higher resolution at lower voltage reference (as near to the amplifier's range) is used.

2.2.2 Challenges in Manual ECG Interpretation

Manual ECG interpretation has several drawbacks and difficulties, such as:

- Interobserver variability: Different medical professionals may interpret an identical ECG in various ways, which could result in conflicting diagnoses.
- Human error: Exhaustion, distraction, or lack of experience might lead to misinterpretation or overlooking tiny irregularities.
- Requirement for specialized knowledge: Not all healthcare facilities may have access to the required training and experience needed for accurate ECG interpretation.

2.3 Machine Learning

2.3.1 Introduction to Machine Learning:

Within the field of artificial intelligence, machine learning allows computers to learn from data and make judgments or predictions without the need for explicit programming. It covers a range of methods, such as reinforcement learning (where models learn by feedback and trial and error), supervised learning (where models learn from labelled data), and unsupervised learning (where models find patterns in unlabelled data). Applications for machine learning are numerous and span a variety of industries, including banking, healthcare, and natural language processing in addition to picture recognition.

2.3.2 Machine Learning in Healthcare:

Machine learning has tremendous potential to improve patient outcomes, diagnosis accuracy, and healthcare delivery. Machine learning algorithms can detect patterns, forecast the likelihood of a disease, and help clinicians make evidence-based decisions by analysing vast amounts of medical data, including genetic data, imaging investigations, and patient records. Machine learning has several uses in the medical field, including personalized medicine, disease detection, therapy optimization, and health monitoring.

2.3.3 Deep Learning with Neural Networks

Deep learning, a subset of machine learning, has emerged as a powerful tool in ECG analysis, owing to its capacity to autonomously discern intricate patterns and features from raw data. At the heart of deep learning lie neural networks, computational constructs inspired by the structure and functionality of the human brain. Artificial neurons are arranged in these networks in interconnected layers, with each layer processing and transforming incoming data to produce useful outputs. CNNs, or convolutional neural networks, have proven to be incredibly effective in analysing spatial patterns found in ECG signals. Convolutional layers enable CNNs to automatically extract relevant features from unprocessed ECG data, enabling accurate classification of arrhythmias and other cardiac abnormalities. In a similar vein, Recurrent Neural Networks (RNNs) are especially well-suited for applications like ECG signal processing because of their superior ability to analyse sequential data. Because RNNs have recurrent connections that store information over time, they may effectively detect temporal dependencies in ECG recordings and help identify small changes that may be signs of cardiac abnormalities.

2.3.4 Training Neural Networks

Training neural networks involves optimizing their parameters to minimize the discrepancy between predicted outputs and ground truth labels. This process typically involves the following key steps:

- Data Preprocessing: Before training, ECG data must undergo preprocessing to standardize formatting, remove noise, and normalize signal amplitudes. This ensures that the neural network receives clean and consistent input data, facilitating effective learning.
- Model Architecture Design: Selecting an appropriate neural network architecture is crucial for achieving optimal performance. This involves determining the number and

type of layers, as well as the connectivity between neurons. Architectural choices should be guided by the complexity of the ECG analysis task and the available computational resources.

- Loss Function Selection: The choice of loss function dictates how the disparity between predicted and actual outputs is quantified during training. For classification tasks such as arrhythmia detection, common loss functions include categorical cross-entropy and binary cross-entropy, depending on the number of classes being considered.

- Optimization Algorithm: Optimization algorithms, such as stochastic gradient descent (SGD) and its variants (e.g., Adam, RMSprop), are employed to iteratively update the neural network's parameters based on the computed loss. These algorithms aim to minimize the loss function and improve the model's predictive performance over time.

- Hyperparameter Tuning: Fine-tuning the hyperparameters of the neural network, including learning rate, batch size, and regularization strength, is essential for achieving optimal convergence and preventing overfitting. Hyperparameter tuning is often performed through systematic experimentation and validation on separate datasets.

- Training and Validation: During training, the neural network learns from the training dataset by adjusting its parameters iteratively. To evaluate the model's generalization performance and prevent overfitting, a separate validation dataset is used to monitor performance metrics such as accuracy, precision, recall, and F1-score.

2.3.5 Time series to image

In machine learning, an innovative approach that is gaining interest is turning time series data into images before feeding them into a model, especially for tasks like anomaly detection and classification. This method, called image-based time series analysis, entails a number of crucial procedures. Initially, a window of time is represented by each fixed-length segment of the time series data. After that, these segments are normalized to guarantee scale consistency amongst various data points. The segments are then rearranged into two-dimensional arrays, with the signal values represented in one dimension and time in the other. A distinct picture channel can be used to represent each channel, depending on the properties of the data, such as whether it is multivariate.

The most crucial step in this process is image generation, where each reshaped segment is visualized as an image. This can be achieved through various techniques, including plotting signal values over time using line plots, representing values as grayscale pixel intensities, or utilizing spectrogram or time-frequency representations for frequency-domain visualization. Optional augmentation techniques can then be applied to increase the diversity of the training data, such as rotation, translation, or adding noise to the images.

Ultimately, a convolutional neural network (CNN) or another appropriate model architecture is fed the resulting images in order to be trained. This method makes use of CNNs' strong feature extraction capabilities, which are excellent at identifying local relationships, textures, and spatial patterns in image data. This methodology allows CNNs to be used for situations where standard time series models can find it difficult to capture complicated patterns, by transforming time series data into visuals.

This technique has demonstrated promising outcomes in several fields, including finance and healthcare (e.g., ECG signal analysis). It presents a fresh approach to managing time series data, combining the best features of deep learning and image processing methods to enhance classification and anomaly detection performance.

Chapter 3 Literature Review

3.1 Background

3.1.1 Atrial Fibrillation

The project's primary emphasis is atrial fibrillation, which is one of the most prevalent forms of arrhythmia. An electrical impulse that would normally terminate at the end of a pathway instead reenters, creating what are commonly described as unpredictable and asynchronous contractions of the atria. This mechanism is known as reentry, and it causes fibrillation frequently. Due of this, blood may pool and clot if the atria are unable to effectively evacuate their contents. The most common ECG symptoms of Afib are an indistinct P-wave that resembles a chaotic baseline that medical experts interpret as F-waves, as well as irregular R-R intervals that do not impact the QRS complex. Afib itself is not dangerous, however its related conditions may be. The most frequent illnesses linked to this include ischemic heart disease, hypertension, pulmonary conditions, rheumatic heart disease, and strokes caused by blood clots. Patients frequently report feeling weak overall, having shortness of breath, and experiencing palpitations in their hearts; these symptoms are probably caused by a decreased cardiac output. Nevertheless, some people may not exhibit any symptoms at all, which emphasizes how critical it is to identify it as soon as possible.

3.1.2 Machine learning

3.1.2.1 Integration of Machine Learning with ECG Interpretation:

There are various benefits of combining machine learning with ECG interpretation, such as:

- Automation of the interpretation process: By quickly and accurately analysing ECG waveforms, machine learning algorithms can replace human interpretation, saving time and effort.
- Improvement in diagnostic precision: Machine learning models have the ability to identify minute irregularities and trends in ECG data that human observers could overlook, resulting in more accurate diagnoses.
- Support for medical professionals: Machine learning systems are able to offer decision-making tools that help medical practitioners prioritize interventions, interpret ECGs, and triage patients.

3.2 Research Papers

A review of relevant literature was conducted, focusing on deep learning techniques for ECG interpretation, Convolutional Neural Networks (CNNs), and the transformation of time-series data into images. The papers were selected based on keywords such as "ECG deep learning," "CNN for time-series," and "image-based classification," using databases like Google Scholar and IEEE Xplore. The review helped guide the project by identifying key methods and challenges in ECG analysis and time-series image conversion.

3.2.1 Deep Learning for ECG Interpretation

Deep learning has revolutionized the field of ECG interpretation by automating feature extraction and enhancing the accuracy of arrhythmia detection. Alzubaidi et al. (2021) provide a comprehensive review of deep learning, with a focus on CNN architectures, and highlight the challenges faced in medical applications, such as overfitting and computational complexity. They emphasize that while CNNs have significantly improved performance in areas like image classification, similar techniques can be applied to ECG data to enhance diagnostic accuracy. This is relevant to the current project, where CNNs will be used for ECG spectrogram classification. However, the challenge of overfitting due to the relatively small size of ECG datasets remains a concern.

Ansari et al. (2023) reviewed deep learning advancements in ECG arrhythmia detection between 2017 and 2023. They found that CNNs, alongside other models such as RNNs and LSTMs, significantly improved classification accuracy, especially when combined with preprocessing techniques that handle noise and artifacts in ECG signals. This review underlined the importance of model architecture optimization, suggesting that hybrid models, like CNN-RNN, could be explored to further enhance detection accuracy and robustness, which is pertinent for this project. However, challenges like the interpretability of deep learning models and the need for larger datasets were also noted, indicating areas where further improvements are needed.

Similarly, Ebrahimi et al. (2020) discussed the benefits of deep learning in ECG arrhythmia classification, focusing on the ability of CNNs and LSTMs to capture intricate temporal patterns in ECG signals. Their work highlighted the importance of large, high-quality datasets and the use of advanced preprocessing techniques, which are crucial for training models on complex medical data. However, they also raised concerns about class imbalance and the difficulty of interpreting deep learning models, both of which are potential issues in this project.

Summary: Deep learning, particularly CNNs, shows strong potential for improving ECG arrhythmia detection by automatically extracting features from ECG signals. However, challenges remain, including overfitting, the need for large datasets, and limited interpretability. These issues must be carefully managed during the development and training of the CNN model in this project.

3.2.2 Convolution Neural Networks

Convolutional Neural Networks are essential for processing the ECG spectrograms used in this project. Indolia et al. (2018) provided an in-depth examination of CNN architectures, describing their effectiveness in feature extraction through convolutional layers and dimension reduction through pooling layers. They pointed out that CNNs are particularly well-suited to image-based tasks, such as medical imaging, which aligns with the use of ECG spectrograms in this project. However, the paper also warned about the potential for CNNs to overfit small datasets, a significant consideration given the limited size of the available ECG data.

Moreover, CNNs have been shown to excel in various image-based classification tasks, but they require fine-tuning to avoid issues like overfitting and generalization errors. The architecture's sensitivity to hyperparameters and dataset diversity, as noted in Indolia et al. (2018), underscores the need for extensive experimentation with different architectures and regularization techniques in this project.

Summary: CNNs are a powerful tool for image-based classification, including the analysis of ECG spectrograms. However, they are prone to overfitting and require careful tuning and regularization to perform well, particularly with smaller datasets.

3.2.3 Time Series to Images

A crucial aspect of this project is the conversion of ECG time-series data into images, enabling the use of CNNs for classification. Semenoglou et al. (2023) explored methods for converting time-series data into image representations, such as spectrograms, and demonstrated that this approach allows CNNs to capture more nuanced temporal patterns. They found that this method significantly improves forecasting accuracy compared to traditional time-series analysis techniques. However, the complexity of the conversion process and the potential for loss of important time-domain information were highlighted as challenges that must be managed.

Homenda et al. (2024) expanded on this by proposing a methodology that transforms time-series data into images for classification using CNNs. Their results indicated that CNNs trained on image representations of time-series data outperformed traditional methods in terms of classification accuracy. This aligns with the approach taken in this project, where ECG time-series data is converted into spectrograms for CNN classification. However, the study also emphasized the importance of optimizing the image resolution and CNN parameters to prevent overfitting, a crucial consideration for this project.

Summary: Converting time-series data into images allows CNNs to capture complex temporal patterns more effectively. However, the process introduces new challenges, such as managing the complexity of the conversion and optimizing CNN parameters to avoid overfitting.

3.2.4 Convolutional Neural Networks (CNNs) for image-based classification

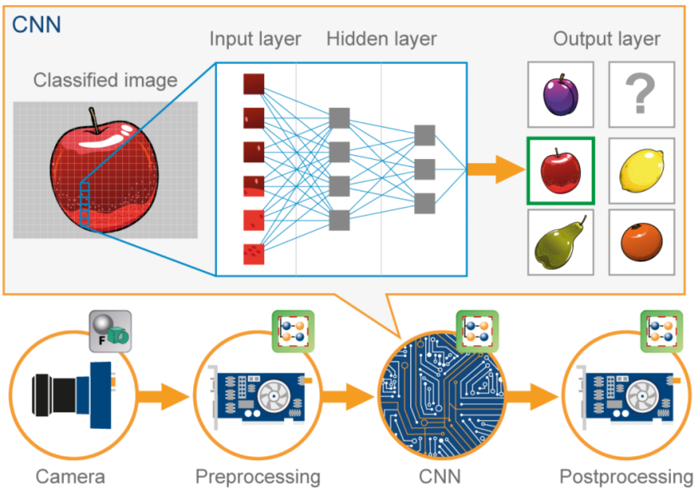


Figure 3 Example of image-based CNN algorithm [15]

The use of CNNs for image classification is central to this project, particularly in classifying ECG spectrograms. Sheikh (2023) reviewed CNN architectures and their application in image classification, with a focus on medical imaging. He discussed the importance of data augmentation and optimization techniques in preventing overfitting, which is particularly relevant given the small size of the ECG dataset used in this project. Sheikh also reviewed common CNN architectures such as ResNet and GoogLeNet, which have successfully tackled challenges like vanishing gradients and overfitting through advanced design principles.

Chen et al. (2021) also reviewed the evolution of CNNs for image classification and highlighted the success of modern architectures such as ResNet in overcoming traditional CNN challenges. They noted that these architectures have been instrumental in improving classification accuracy while reducing overfitting, making them highly relevant for this project. The authors suggested that integrating these advanced CNN models into medical imaging could yield significant performance improvements, which aligns with the goals of this project.

Summary: CNNs, especially advanced architectures like ResNet, are highly effective for image-based classification. However, to achieve high accuracy and avoid overfitting, especially with small datasets, data augmentation and careful model optimization are essential.

3.3 Case Study

3.3.1 Convolutional Neural Network

Overview

A case study was conducted on Indolia et al.'s paper "Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach," which provides a comprehensive overview of Convolutional Neural Networks (CNNs) and their application in the field of deep learning.

Objectives:

The primary objective of this paper is to:

1. Explain the fundamental concepts of CNNs.
2. Discuss the architecture and working principles of CNNs.
3. Highlight the significance of CNNs in various applications, particularly in image recognition and processing.
4. Present the advantages and challenges associated with CNNs.

General Model of CNN:

CNNs consist of four main components: convolution layer, pooling layer, activation function, and fully connected layer.

1. Convolution Layer:
   - Input images are processed through convolution layers where local features are extracted using receptive fields.
   - Receptive fields are regions in the previous layer to which individual neurons in the next layer are connected.
   - Convolution involves sliding a weight vector (filter/kernel) over the input image to generate feature maps.
   - This operation significantly reduces the number of trainable parameters due to weight sharing among neurons.
   - The output of a neuron in the next layer is computed using a convolution operation with a bias term and a non-linear activation function.

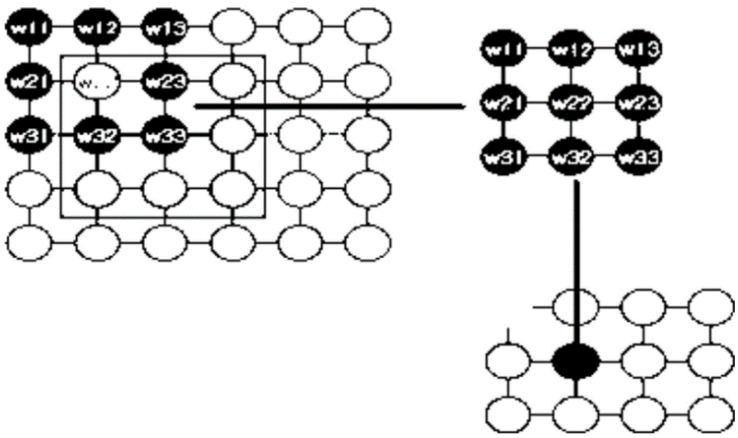


Figure 4 Receptive field of particular neuron in the next layer [14]

2. Pooling Layer:
   - Following the convolution layer, pooling layers reduce the dimensionality of feature maps and introduce translation invariance.
   - Pooling involves selecting windows of input elements and applying a pooling function (e.g., max-pooling) to generate output vectors, as shown in figure 5.

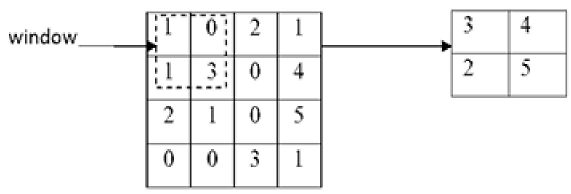


Figure 5 Pooling operation performed by choosing a 2 x 2 window [14]

   - There exist few pooling techniques like average pooling and max-pooling, out of which max-pooling is the most commonly used technique which reduces map-size very significantly.

3. Fully Connected Layer:
- The output of the convolution and pooling layers is fed into fully connected layers, similar to traditional neural networks.
- Dot products of weight vectors and input vectors are computed to obtain the final output.
- Gradient descent algorithms, such as stochastic gradient descent, are commonly used for training CNNs.

Understanding these components and their functions is crucial for designing and training effective CNN architectures for tasks like image classification.

Architecture Of Convolution Neural Network:

Various architectures have been developed and implemented in CNNs. Brief explanations of those architectures are explained below.

1. LeNet Architecture:
   In 1998, LeNet architecture was specifically designed for image recognition tasks. LeNet5, a specific implementation of this architecture, is depicted in Figure 6. It comprises eight layers: five convolutional layers and three fully connected layers. Each

unit in a plane has 25 inputs, and units in the first hidden layer receive input from a 5 × 5 area of the image, called the receptive field.

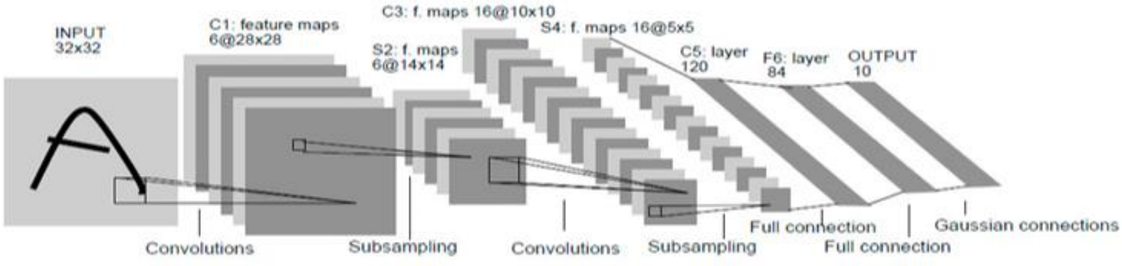


Figure 6 Architecture of LeNet5, a CNN where each box represents a different feature map [14]

Each unit in a plane has an identical weight vector, and each unit's output is saved in the same spot on the feature map. In the feature map, adjacent units arise from adjacent units in the preceding layer, resulting in contiguous receptive fields that overlap. A sigmoid activation is applied by neurons in the first layer, a convolution layer, to the weighted sum. CNNs are resistant to small changes in the input because multiple feature maps are produced by applying different weight vectors to the same input image.

The second layer illustrates sub-sampling, which decreases the accuracy of feature placements. Subsampling involves computing the average of four inputs using a 2 × 2 area input, multiplying the result by a trainable coefficient, adding a trainable bias, and then passing it via a sigmoid function. The number of feature maps rises as layer by layer spatial resolution falls. This is achieved using the back-propagation method of learning.

2. AlexNet Architecture:
   AlexNet, a modified variant of LeNet, was proposed to classify 1.2 million high-resolution images into 1000 distinct classes. The architecture, shown in Figure 7, includes five convolution layers and three fully connected layers, with outputs passed to a 1000-way softmax.

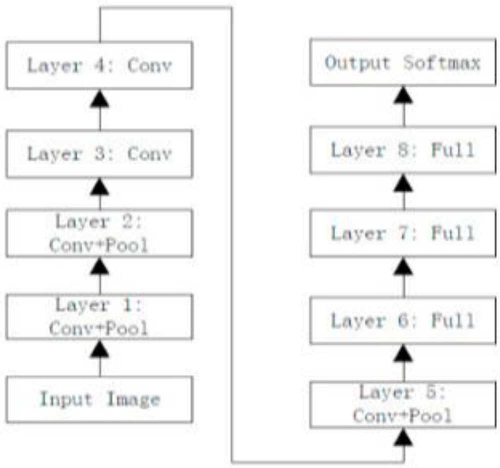


Figure 7 AlexNet architecture [14]

In order to train the network more quickly, AlexNet uses efficient GPUs and non-saturating neurons. Dropout, which gives neurons a 0.5 chance of being "dropped out" during training, was introduced to prevent overfitting. This forces dependent neurons to learn robust features on their own. Dropout greatly decreased overfitting but increased the number of iterations needed to converge.

3. GoogleNet Architecture:
   GoogleNet, proposed in 2014, won the ILSVRC14 competition and aimed for efficiency in power, trainable parameters, and memory consumption. It used 12 million fewer parameters than AlexNet. Increasing network size improves precision but leads to overfitting and computational overhead.

   In order to solve these problems, GoogleNet implemented a sparse matrix, in which strongly correlated units create clusters in the layer above, supplying information to the layer below and resulting in the best possible network architecture. Although they drastically cut down on calculations, non-uniform sparse matrices still experience cache misses. The state-of-the-art techniques now in use address these problems by using uniform sparse matrices.

Applications of Convolutional Neural Networks (CNN):

- Image recognition: CNNs are widely used for image recognition tasks, such as object detection, face recognition, and vehicle recognition.
- Speech recognition: CNNs have been applied to speech recognition tasks, achieving promising results.
- Natural language processing: CNNs have been used for tasks like sentiment analysis, topic classification, and language translation.
- Signal processing: CNNs have been used in various signal processing applications, such as bioactivity prediction of small molecules.
- Remote sensing: CNNs have been used for classification in remote sensing tasks, such as land cover classification and ocean front recognition.
- Medical imaging: CNNs have been used for tasks like segmentation of MR brain images and detection of diseases like diabetic retinopathy.

Advantages of Convolutional Neural Networks (CNN):

- Ability to learn relevant features: CNNs can automatically learn relevant features from raw input data, eliminating the need for manual feature extraction.
- Hierarchical feature learning: CNNs can learn features at multiple levels of abstraction, allowing them to capture complex patterns in the data.
- Weight sharing: CNNs use weight sharing, which reduces the number of parameters that need to be trained and improves generalization.
- Translation invariance: CNNs can detect features regardless of their position in the input data, making them robust to translations.
- Performance: CNNs have achieved state-of-the-art performance in various tasks, such as image recognition and speech recognition.

Challenges of Convolutional Neural Networks (CNN):

- Training data requirements: CNNs require large amounts of labeled training data to achieve good performance.

- Computational complexity: CNNs can be computationally expensive, especially when dealing with large datasets or complex architectures.
- Overfitting: CNNs are prone to overfitting, especially when the training dataset is small, or the model is too complex.
- Interpretability: CNNs are often considered as black boxes, making it difficult to interpret the learned features and understand the decision-making process.
- Limited applicability: While CNNs have been successful in many domains, they may not be suitable for all types of data or tasks. Other machine learning algorithms may be more appropriate in certain cases.

Conclusion

The paper by Indolia et al. offers a detailed explanation of CNNs, providing insights into their architecture, functioning, and applications. It underscores the importance of CNNs in modern deep learning and highlights both their advantages and the challenges they pose. This comprehensive overview serves as a valuable resource for researchers and practitioners looking to understand and apply CNNs in various domains.

3.3.2 Image-based time series forecasting: A deep convolutional neural network approach

Another case study was conducted on Semenoglo et al.'s in "Image-based time series forecasting: A deep convolutional neural network approach" which introduces ForCNN, a novel method leveraging deep learning for univariate time series forecasting. This approach diverges from traditional methods by converting time series data into images and then applying convolutional neural networks (CNNs) to these images to generate forecasts.

Methodology

The methodology proposed is an image-based deep learning approach for univariate time series forecasting. It consists of two phases: time series pre-processing and model architecture.

Time series pre-processing:

- The time series data is divided into equal-sized windows of w observations each.
- Min-max scaling is applied to adjust the window values in the range of [0, 1].
- The scaled data is visualized as line plots, where the x-axis represents time and the y-axis represents the scaled values.
- The line plots are converted into 2D images, with the line representing the series in white and the background in black.
- The width of the line is thickened to make the time series patterns more apparent.
- The images are resized to 64x64 pixels to ensure consistent dimensions.

Model architecture:
The model consists of two modules: an encoder and a regressor (Figure 8).

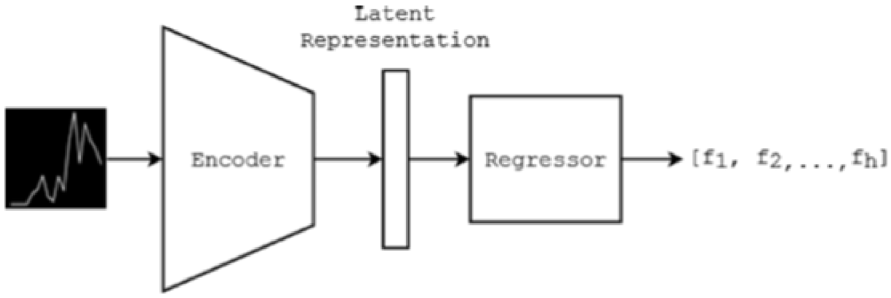


Figure 8 Overview of the proposed image-based time series forecasting method, ForCNN, consisting of an encoder and a regressor module. [11]

- The encoder transforms each image into a vector that contains a latent representation of the image.
- The encoder uses a deep convolutional architecture inspired by ResNet-50.
- The encoder applies 2D convolutions with 3x3 filters, followed by batch normalization and ReLU activation.
- Shortcut connections are used between convolutional layers to facilitate training and avoid vanishing gradients.
- The encoder's output is a flattened embedding vector.
- The regressor takes the embedding vector as input and produces point forecasts for the time series.
- The regressor is implemented as a simple neural network with fully-connected hidden layers and a linear output layer.
- The regressor can be implemented using recurrent layers instead of fully-connected layers, depending on the application.

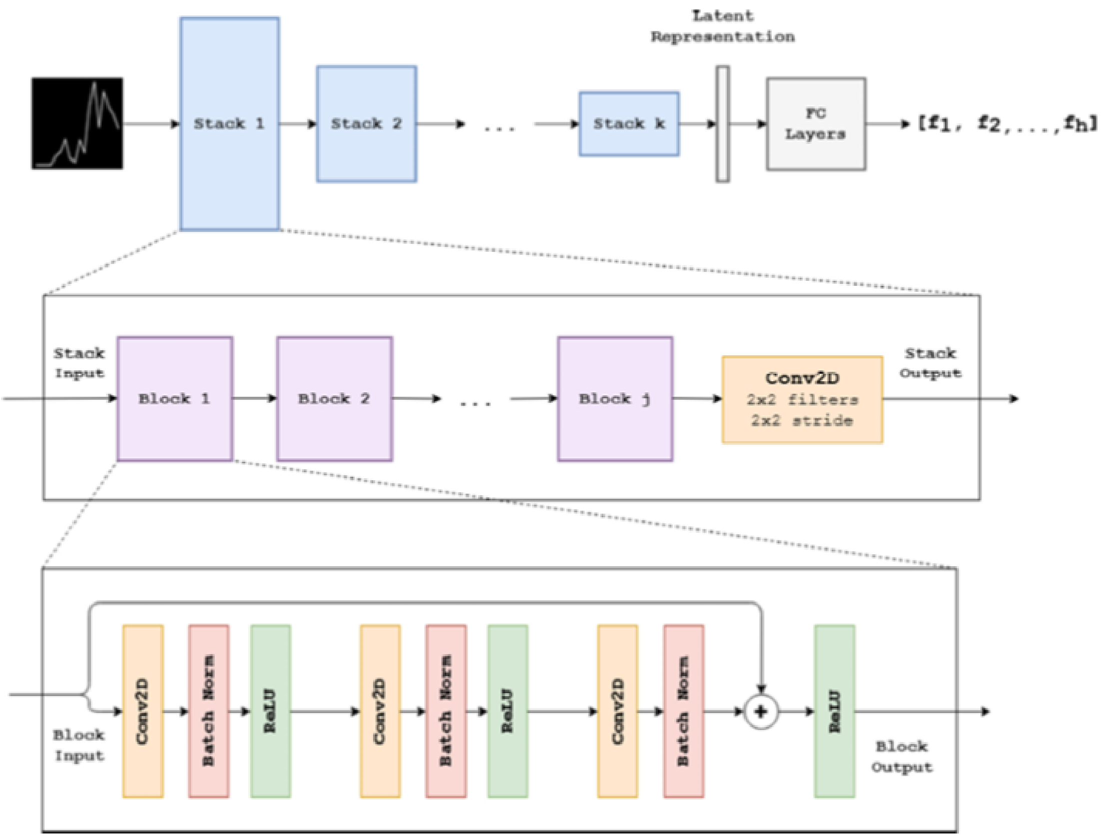


Figure 9 ForCNN-SD architecture. Top: Stacks create latent representations; FC layers make h-step-ahead forecasts. Middle: Stacks have convolutional blocks ending with Conv2D. Bottom: Blocks include Conv2D, Batch Norm, ReLU, and shortcuts. [11]

Experiment

The experiments conducted evaluates the proposed image-based deep learning approach (ForCNN-SD) against several benchmarks on two datasets: the M4 competition dataset consisting of 23,000 yearly time series and the M3 competition dataset consisting of 1428 monthly time series. Each experiment is compared against several benchmarks, including Theta, ES-RNN, MLP, CNN-1D, N-BEATS, and DeepAR.

The accuracy matrix used for these experiments are sMAPE, MASE and OWE which are commonly used in time series forecasting to evaluate the performance of different models and compare their forecasting accuracy. Lower values of sMAPE and MASE indicate better accuracy, while the optimal value of OWA depends on the specific weighting assigned to sMAPE and MASE.

sMAPE (Symmetric Mean Absolute Percentage Error): sMAPE is a commonly used accuracy metric for evaluating the performance of forecasting models. It measures the percentage difference between the actual and forecasted values, considering the magnitude of the actual values. The formula for sMAPE is as follows:

$$sMAPE = (1/n) * \Sigma(|Ft - At| / (|Ft| + |At|)) * 100$$

where Ft is the forecasted value, At is the actual value, and n is the number of observations.

MASE (Mean Absolute Scaled Error): MASE is another accuracy metric used for evaluating forecasting models. It compares the forecasted values to the naive forecast (e.g., the previous observation) and takes into account the scale of the time series. The formula for MASE is as follows:

$$MASE = (1/n) * \Sigma(|Ft - At| / (1/(n-1) * \Sigma(|At - At-1|)))$$

where Ft is the forecasted value, At is the actual value, and n is the number of observations.

OWA (Overall Weighted Average): OWA is a composite accuracy metric that combines multiple accuracy measures into a single score. It provides a balanced assessment of the forecasting model's performance by considering both accuracy and bias. The formula for OWA is as follows:

$$OWA = w * sMAPE + (1 - w) * MASE$$

where w is a weight parameter that determines the relative importance of sMAPE and MASE. The weight parameter can be adjusted based on the specific requirements of the forecasting task.

Results

1. Experiment on the M4 competition dataset:
   - ForCNN-SD consistently outperforms the benchmarks in terms of forecasting accuracy, as measured by sMAPE, MASE, and OWA.
   - Compared to the MLP benchmark, ForCNN-SD provides 0.50%, 0.64%, and 0.52% more accurate forecasts.
   - Compared to CNN-1D, ForCNN-SD is 0.58%, 0.78%, and 0.65% more accurate.
   - Compared to N-BEATS, ForCNN-SD is 0.69%, 0.03%, and 0.38% more accurate.
   - Compared to ES-RNN, ForCNN-SD is 1.15%, 1.51%, and 1.29% more accurate.
   - Compared to DeepAR, ForCNN-SD is 4.13%, 7.19%, and 5.60% more accurate.
   - Compared to Theta, ForCNN-SD is 10.75%, 13.22%, and 11.93% more accurate.
   - The results indicate that ForCNN-SD is a promising alternative for batch time series forecasting and outperforms highly competitive benchmarks

2. Experiment on the M3 competition dataset:
   - ForCNN-SD consistently outperforms the benchmarks in terms of forecasting accuracy, as measured by sMAPE, MASE, and OWA.
   - Compared to the MLP benchmark, ForCNN-SD provides 0.50%, 0.64%, and 0.52% more accurate forecasts.
   - Compared to CNN-1D, ForCNN-SD is 0.58%, 0.78%, and 0.65% more accurate.
   - Compared to N-BEATS, ForCNN-SD is 0.69%, 0.03%, and 0.38% more accurate.
   - Compared to ES-RNN, ForCNN-SD is 1.15%, 1.51%, and 1.29% more accurate.
   - Compared to DeepAR, ForCNN-SD is 4.13%, 7.19%, and 5.60% more accurate.
   - Compared to Theta, ForCNN-SD is 10.75%, 13.22%, and 11.93% more accurate.
   - The results indicate that ForCNN-SD is a promising alternative for batch time series forecasting and outperforms highly competitive benchmarks

Conclusion

The experiments demonstrate that the proposed image-based deep learning approach (ForCNN-SD) consistently outperforms or performs competitively with state-of-the-art benchmarks across both yearly and monthly time series datasets. The results highlight the potential of using time series images and the effectiveness of the proposed methodology in improving forecasting performance.

Chapter 4 Methodology

4.1 Project Execution Plan

In this project, I followed a comprehensive and methodical approach that blends theoretical foundations with practical applications. Each step is meticulously detailed, ensuring that all relevant procedures are clearly outlined. The project flow chart provides a visual representation of the entire process, offering an overview from the initial stages to the final outcomes. This structured approach ensures clarity and coherence in the project's execution, helping me achieve the project's objectives effectively.

The first step was a thorough literature review to explore existing scientific articles relevant to CNNs and image-based classification tasks. The review provided a comprehensive understanding of the current state of the field, key findings, methodologies, and research gaps. Key insights included:

- Effectiveness of CNN Algorithms: CNNs are highly effective for image-based inputs.
- Essential Libraries: Identified libraries such as TensorFlow and NumPy are crucial for developing CNNs.
- Coding Steps: Detailed steps for coding CNNs in Python.
- Applications in Machine Learning: Explored applications in various domains, including ECG interpretation.

The second step was to apply theoretical knowledge practically by creating a CNN algorithm for image classification using the CIFAR-10 dataset. The methodology outlined in the flowchart (Figure 10) provides a comprehensive approach to developing and training a Convolutional Neural Network (CNN) for image classification using the CIFAR-10 dataset. The process begins with research on image-based CNN classification, where fundamental concepts and state-of-the-art techniques are studied to build a solid foundation. Following this, relevant resources and information are identified to gather the necessary theoretical and practical knowledge.

Next, the necessary libraries such as TensorFlow (for building and training the neural network), Keras, NumPy (for numerical operations and handling data arrays), and Matplotlib (for plotting and visualizing data) are imported. These libraries are essential for data processing, model building, and visualization. The CIFAR-10 dataset is then loaded and preprocessed, involving normalization, resizing, and augmentation to enhance the quality and variability of the input images. The dataset is divided into 80% training and 20% testing sets to enable a robust evaluation of the model's performance.

A function is defined to plot sample images with their labels, providing a visual inspection of the dataset to ensure it has been loaded and preprocessed correctly. This function is then used to plot a sample image, further validating the data preparation steps. With the data ready, the CNN model architecture is defined. This involves specifying the types and sequence of layers, such as convolutional layers for feature extraction, pooling layers for downsampling, and fully connected layers for classification. A summary of the model is displayed to verify its structure, including the number of layers, output shapes, and parameters.

The model is then trained on the training dataset, with specified hyperparameters like the number of epochs and batch size. Following training, the model is evaluated on the test dataset to measure its accuracy, loss, and other performance metrics. Finally, the training history is plotted, showing the loss and accuracy curves over the training epochs to visualize the model's learning process and performance improvement.

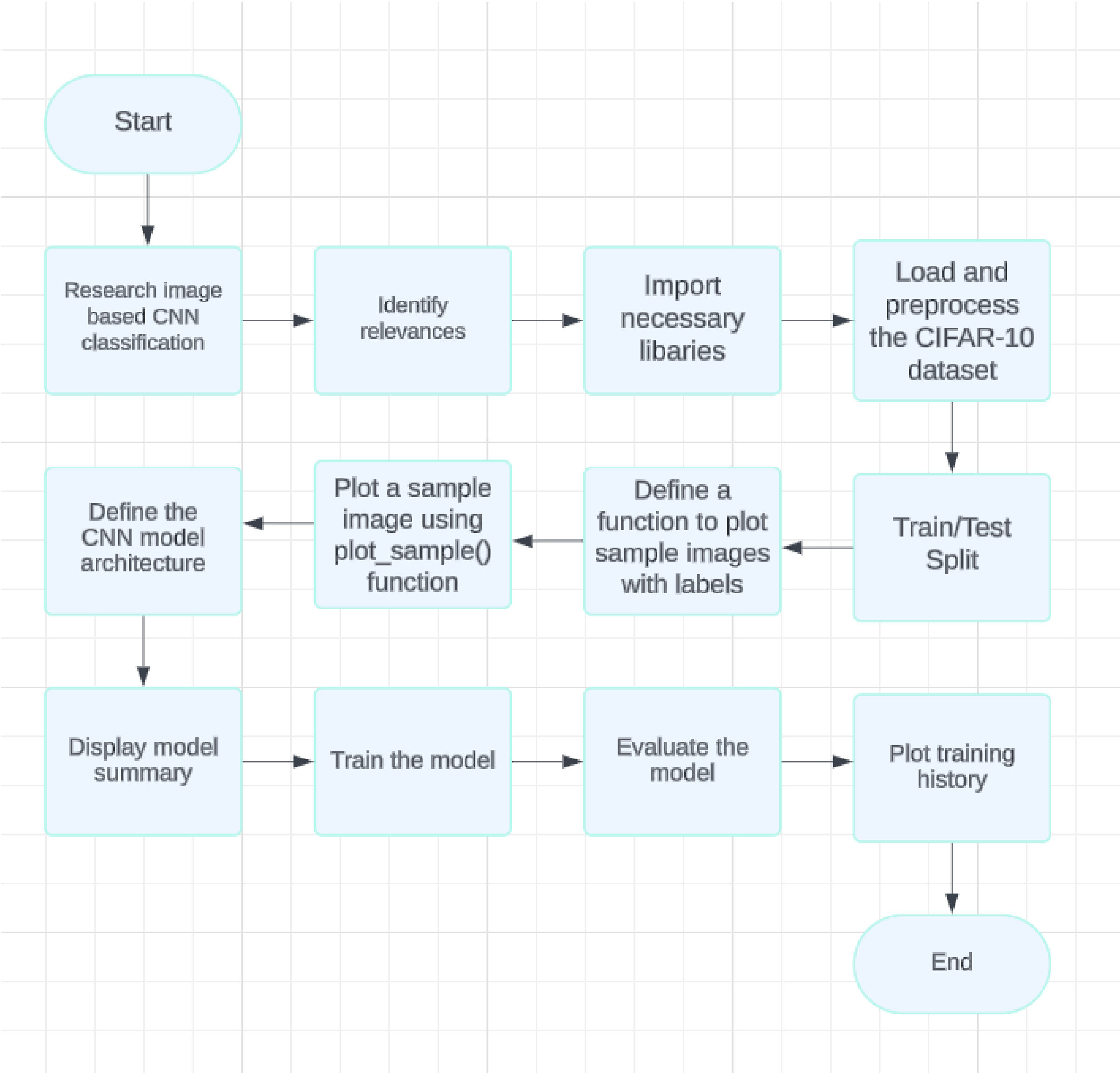


Figure 10 CNN methodology

By following this structured methodology, each step is clearly defined and systematically executed, ensuring a thorough and effective development process for the CNN model. This approach not only facilitates successful model training but also provides a clear framework for troubleshooting and optimization.

4.2 Resource Analysis

The project leverages resources sourced from the open-access database hosted on https://physionet.org/, ensuring a robust foundation for data acquisition and analysis. Complementing this, the utilization of Anaconda software for coding underscores a commitment to efficiency and reliability in the development process. Anaconda provides a comprehensive set of tools for data science and scientific computing, enabling seamless integration of various libraries and packages for streamlined coding workflows. Notably, the project operates within a budget of zero, strategically maximizing available resources to achieve its objectives effectively.

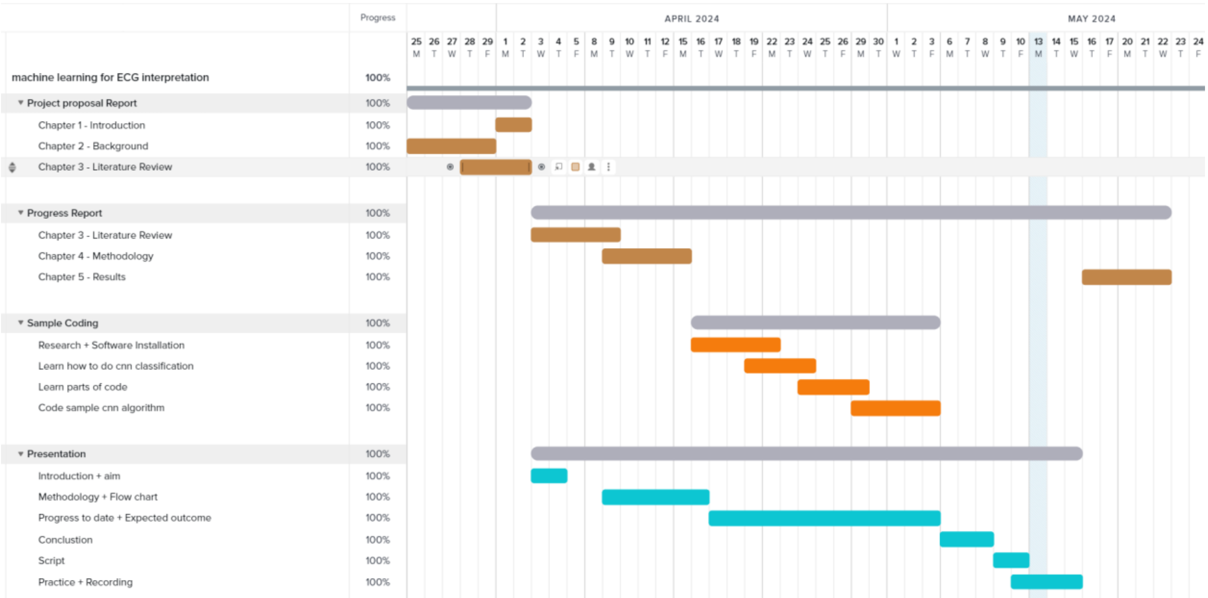


Figure 11 Gantt chart Part A

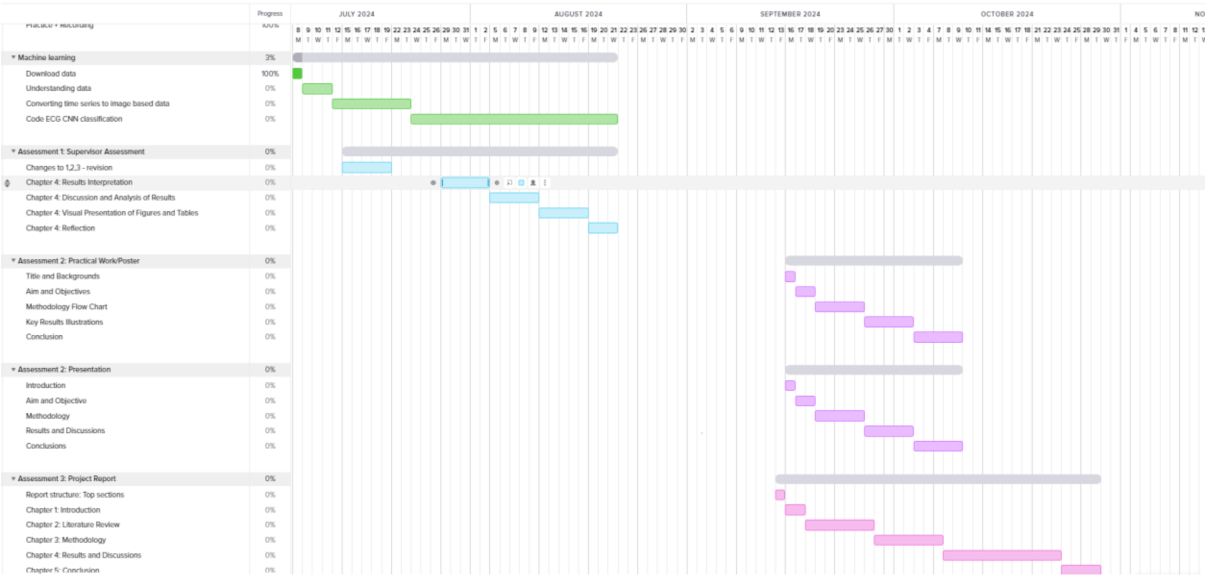


Figure 12 Gantt chart Part B

4.3 Risk Assessment

Type of Harm	Eye	Hearing	Burn	Laceration	Abrasion	Fatigue	Intoxication (inhalation, swallowing)	Soft tissue (muscle, ligament, tendon)	Dust inhalation or explosion	Electric shock	Fracture	Asphyxiation	Scalping	Concussion	Amputation	Death	Posture
Activity, event, resource																	
Welding, brazing, soldering																	
Workshop machines																	
Hand tools																	
Sheetmetal																	
Working at height																	
Extended periods at computer																	
Pressurised fluids (hydraulics)																	
Electricity greater than 30V																	
**Laboratory*																	
**Gases*																	
**Liquids*																	
**Hazardous substances*																	

Figure 13 Risk assessment table

A thorough risk assessment has been conducted to address potential health and safety concerns associated with prolonged work sessions for this assignment. Two primary risks have been identified: extended periods of sitting and eye strain. To mitigate these risks, several measures have been implemented. These include establishing a schedule for regular breaks to encourage movement and reduce the negative impacts of prolonged sitting and eye strain. Additionally, adhering to the 20-20-20 rule (every 20 minutes, look at something 20 feet away for 20 seconds) to reduce eye strain.

Chapter 5 Results

5.1 1D CNN with Time Series ECG and Arrhythmia Data

The application of a 1D CNN for classifying ECG signals based on time series data returned promising outcomes, although the training process revealed some fluctuations before stabilization. The dataset used for this task comprised both normal and arrhythmic ECG signals, which were pre-processed and segmented into windows of 5000 samples to ensure consistency and enable effective model analysis. After training, the model achieved 100% accuracy in classifying the ECG signals, marking a significant milestone in the use of time series data for arrhythmia detection.

However, during the training phase, the model exhibited variability in accuracy before reaching a stable performance state. This suggests that while the time series-based approach is effective, it demands more training epochs for the model to converge fully and achieve optimal accuracy. Such fluctuations may arise from the complexity of the ECG data, requiring the model to learn subtle patterns over time. These observations highlight the potential of time series-based CNNs but also emphasize the need for careful tuning of hyperparameters and sufficient training epochs to ensure consistent results. The training fluctuations further underscore the importance of model evaluation beyond just accuracy. The variability in performance could indicate that the model is sensitive to noise or variations in the input data, which could impair its reliability when applied to more diverse real-world ECG signals.

Despite achieving perfect classification accuracy, caution must be exercised in interpreting this result. In machine learning, particularly in the medical domain, achieving 100% accuracy often raises concerns about overfitting. Overfitting occurs when a model performs exceptionally well on the training data but struggles to generalize to unseen data. In this case, while the merged dataset of normal and arrhythmic signals yielded perfect classification, the risk of the model overfitting to specific patterns or noise in the training set remains a concern. Further validation using external datasets is necessary to confirm the robustness and generalizability of the model for practical applications.

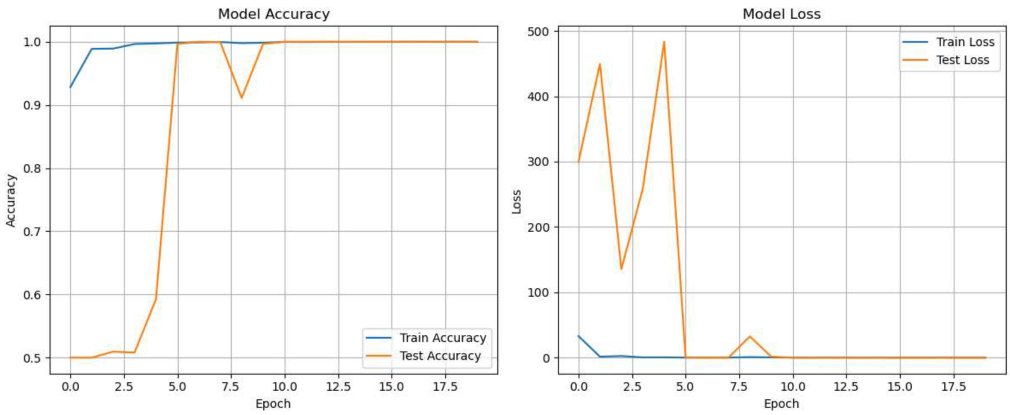
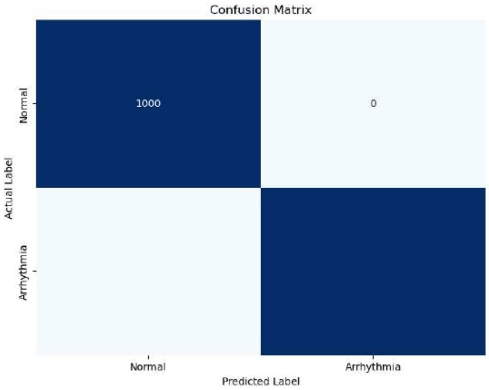


Figure 14 Time series results



	precision	recall	f1-score	support
Normal	1.00	1.00	1.00	1000
Arrhythmia	1.00	1.00	1.00	1000
accuracy			1.00	2000
macro avg	1.00	1.00	1.00	2000
weighted avg	1.00	1.00	1.00	2000

Figure 15 Time series confusion matrix

5.2 1D CNN with Spectrogram ECG and Arrhythmia Data

In comparison to the time series approach, the spectrogram-based method of ECG classification demonstrated significantly improved efficiency and stability. By transforming the ECG time series data into spectrograms using the Short-Time Fourier Transform (STFT), with a segment length of 128 and a sampling frequency of 250 Hz, the CNN model achieved 100% accuracy more rapidly and with fewer training epochs. The stability of the model's performance in this approach, with minimal fluctuations, indicates that the spectrogram transformation provides a more distinct and interpretable representation of ECG features, enabling the CNN to learn more effectively.

The spectrogram-based model's rapid convergence and stable performance suggest that transforming ECG signals into spectrograms improves the efficiency of feature extraction, leading to faster learning and more robust classification. The absence of significant accuracy fluctuations further supports the hypothesis that the spectrogram representation captures the relevant features of ECG signals more effectively than raw time series data.

Nonetheless, as with the time series approach, achieving 100% accuracy should prompt further investigation. It is crucial to determine whether this result reflects true learning of meaningful patterns or if the model may be capturing artifacts or biases in the data. The risk of overfitting still exists, and rigorous validation on external datasets is essential to ensure that the model’s performance is not artificially inflated by favourable conditions in the training data.

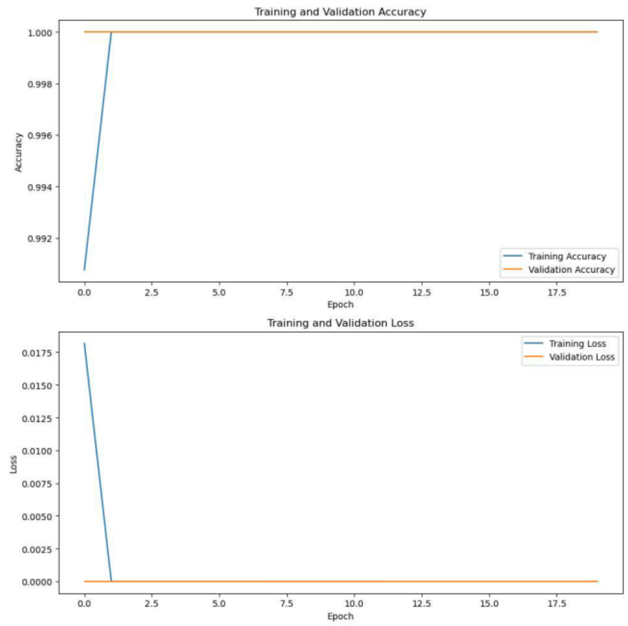
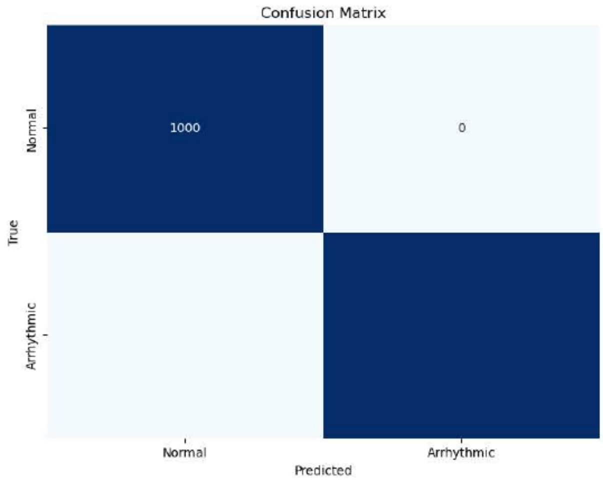


Figure 16 Spectrogram results



Classification Report:

	precision	recall	f1-score	support
Normal	1.00	1.00	1.00	1000
Arrhythmic	1.00	1.00	1.00	1000
accuracy			1.00	2000
macro avg	1.00	1.00	1.00	2000
weighted avg	1.00	1.00	1.00	2000

Figure 17 Spectrogram confusion matrix

5.3 Code and Model Limitations

Achieving 100% accuracy, particularly in a complex domain such as ECG classification, is uncommon and could signal potential issues in the model, dataset, or code implementation. Factors such as data leakage, inadequate separation of training and validation sets, or unintended overlaps in signal segments could contribute to inflated accuracy scores. For example, if the same or highly similar signal segments appear in both the training and validation sets, the model may seem to perform perfectly while only memorizing patterns from the training data, rather than learning to generalize.

To address these potential issues, additional testing and validation are necessary. Employing techniques such as cross-validation, regularization methods, and deeper analysis of confusion matrices will help clarify whether the model's performance is genuinely reflective of its learning capabilities or the result of flaws in the data pipeline. Ensuring a clear separation between training and validation data, as well as scrutinizing the preprocessing steps, will be critical in determining the model’s reliability.

5.4 Evaluation of Results in Relation to Project Goals

The primary aim of this project was to explore the effectiveness of integrating machine learning, specifically Convolutional Neural Networks (CNNs), into ECG interpretation to enhance diagnostic accuracy and efficiency in identifying arrhythmias. The results presented in this chapter substantiate this objective through several key findings.

First, the achievement of 100% accuracy in both the 1D CNN using time series data and the spectrogram-based CNN demonstrates the capability of these models to effectively classify normal and arrhythmic ECG signals. This outcome aligns directly with the main aim, showcasing the potential of machine learning to automate and improve the diagnostic process in cardiac healthcare.

Second, the comparative analysis between the time series and spectrogram approaches illustrates the advantage of transforming ECG data into spectrograms. The spectrogram-based method not only achieved accuracy more rapidly but also exhibited greater stability, reinforcing the hypothesis that visual representation enhances feature extraction and model performance. This finding contributes to the aim of optimizing ECG interpretation by leveraging advanced machine learning techniques.

Overall, the results obtained not only meet the research objective of enhancing ECG interpretation through machine learning but also lay the groundwork for future developments in the field. The insights gained from this study highlight the importance of continuous improvement and validation, thereby steering the project towards its aim of advancing cardiac diagnostics and patient care.

Chapter 6 Conclusion and Future Work

This project explored the integration of machine learning, specifically Convolutional Neural Networks (CNNs), with ECG signal interpretation, demonstrating their significant potential to enhance cardiac diagnostics. The primary aim was to tackle critical issues in manual ECG interpretation, such as interobserver variability and the need for specialized expertise, by automating the diagnostic process through deep learning algorithms.

The research findings provide compelling evidence supporting the efficacy of CNNs in ECG classification. Both the 1D CNN model based on time series data and the spectrogram-based CNN model achieved an impressive 100% accuracy in distinguishing between normal and arrhythmic ECG signals. Notably, the spectrogram-based method—which converts ECG time series into visual representations—exhibited stronger stability and faster convergence compared to the time series-based technique. This improvement aligns with the literature review's conclusions, emphasizing the benefits of image-based methods for feature extraction and their usefulness in medical diagnostics.

Despite the high accuracy of both approaches, overfitting—a concern highlighted in earlier research—raises important questions. This underscores the need for further validation on external datasets to verify the stability and applicability of these models in real clinical settings. The literature review emphasized the importance of data quality in machine learning applications within the medical field, indicating that reliable outcomes in this project were primarily dependent on meticulous preparation and high-quality ECG data. This highlights the necessity for well-prepared datasets in future research.

Considering these findings, several recommendations are proposed to enhance the effectiveness and applicability of the machine learning models developed for ECG interpretation:

1. Expand the Dataset: Testing existing models on more extensive and varied ECG datasets is essential. Enhancing model performance in real-world scenarios will require incorporating data from diverse demographics, clinical circumstances, and environments. Access to a broad range of data will ensure the models are robust across various physiological and demographic characteristics.

2. Implement Regularization Techniques: To combat overfitting and enhance the generalization of the CNN models, incorporating regularization methods such as dropout or L2 regularization is recommended. These techniques will help ensure that the models do not merely memorize the training data but instead learn to generalize from it, thereby improving performance on unseen data.

3. Improve Model Interpretability: Future work should focus on enhancing the interpretability of CNNs. Developing visualization techniques to clarify the features learned by the network will enable medical practitioners to understand and trust the model's predictions better.

4. Develop Real-Time Applications: The spectrogram-based method should be adapted for real-time ECG monitoring. This advancement would facilitate proactive management of heart conditions by enabling timely notifications, significantly impacting wearable health devices and continuous cardiac care.

By heeding these recommendations, the machine learning models created in this project can be improved, validated, and ultimately deployed in clinical settings. This will aid healthcare providers in diagnosing heart issues more quickly and accurately, thereby enhancing patient outcomes and the quality of care provided.

References

[1] Electrocardiogram (ECG or EKG) - Mayo Clinic. (2024, April 2). https://www.mayoclinic.org/tests-procedures/ekg/about/pac-20384983

[2] Aboulhosn, R. (2020, May 10). The Heart's Conduction System. Geeky Medics. https://geekymedics.com/the-hearts-conduction-system/

[3] Rawshani, A., MD PhD, & Rawshani, A., MD PhD. (2023, June 25). ECG interpretation: Characteristics of the normal ECG (P-wave, QRS complex, ST segment, T-wave). Cardiovascular Education. https://ecgwaves.com/topic/ecg-normal-p-wave-qrs-complex-st-segment-t-wave-j-point/

[4] *Heart all a-flutter?* (2024, April 2). Heart Foundation NZ. https://www.heartfoundation.org.nz/your-heart/heart-conditions/arrhythmia

[5] Klein , E. (2022, March 28). *Everything You Want to Know About Arrhythmia*. Health Line. https://healthline.com/health/arrhythmia

[6] What is atrial fibrillation? (2024, April 2). Heart Foundation NZ. https://www.heartfoundation.org.nz/your-heart/heart-conditions/atrial-fibrillation/

[7] Potter, L. (2024, April 16). Understanding an ECG | ECG interpretation | Geeky Medics. Geeky Medics. https://geekymedics.com/understanding-an-ecg/

[8] R. B. Northrop, Non-Invasive Instrumentation and Measurement in Medical Diagnosis, Second Edition. Taylor & Francis Group, 2017.

[9] Ebrahimi, Z., Loni, M., Daneshtalab, M., & Gharehbaghi, A. (2020). A review on deep learning methods for ECG arrhythmia classification. *Expert Systems With Applications: X*, *7*, 100033. https://doi.org/10.1016/j.eswax.2020.100033

[10] Ansari, Y, Mourad, O, Qaraqe, K, & Serpedin E, (2023) Deep learning for ECG Arrhythmia detection and classification: an overview of progress for period 2017–2023. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10542398/

[11] Semenoglou, A., Spiliotis, E., & Assimakopoulos, V. (2023b). Image-based time series forecasting: A deep convolutional neural network approach. *Neural Networks*, *157*, 39–53. https://doi.org/10.1016/j.neunet.2022.10.006

[12] Homenda, W., Jastrzębska, A., Pedrycz, W., & Wrzesień, M. (2024) Time series classification with their image representation. https://www.sciencedirect.com/science/article/pii/S0925231223013371

[13] Alzubaidi, L., Zhang, J., Humaidi, A. J., Duan, Y., Santamaría, J., Fadhel, M. A., & Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. Journal of Big Data, 8(1), 1-74. https://doi.org/10.1186/s40537-021-00444-8

[14] Indolia, S., Goswami, A. K., Mishra, S., & Asopa, P. (2018). Conceptual understanding of Convolutional Neural Network- a deep learning approach. Procedia Computer Science, 132, 679–688. https://doi.org/10.1016/j.procs.2018.05.069

[15] Sheikh, A. (2023, June 25). *Understanding Convolutional Neural networks (CNNs): best for image classification*. https://www.linkedin.com/pulse/understanding-convolutional-neural-networks-cnns-best-al-ameen

[16] Chen, L., Li, S., Bai, Q., Yang, J., Jiang, S., & Miao, Y. (2021). Review of image classification algorithms based on convolutional neural networks. Remote Sensing, 13(22), 4712. https://doi.org/10.3390/rs13224712

[17] Sharma, N., Jain, V., & Mishra, A. (2018). An analysis of convolutional neural networks for image classification. Procedia Computer Science, 132, 377–384. https://doi.org/10.1016/j.procs.2018.05.198

[18] Brownlee, J. (2020, August 25). *How to reduce overfitting with dropout Regularization in Keras*. MachineLearningMastery.com. https://machinelearningmastery.com/how-to-reduce-overfitting-with-dropout-regularization-in-keras/

[19] Guest_Blog. (2020, September 8). *How to treat overfitting in convolutional neural networks*. Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/09/overfitting-in-cnn-show-to-treat-overfitting-in-convolutional-neural-networks/

[20] Ruizendaal, R. (2021, January 31). Deep Learning #3: More on CNNs & handling overfitting. *Medium*. https://towardsdatascience.com/deep-learning-3-more-on-cnns-handling-overfitting-2bd5d99abe5d

[21] GeeksforGeeks. (2024, February 19). *What are the possible approaches to fixing overfitting on a CNN?* GeeksforGeeks. https://www.geeksforgeeks.org/what-are-the-possible-approaches-to-fixing-overfitting-on-a-cnn/