CONVOLUTIONAL NEURAL NETWORK BASED CLASSIFICATION OF THREE CARDIAC CONDITIONS

DISSERTATION

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE
OF

MASTER OF TECHNOLOGY
IN
CONTROL AND INSTRUMENTATION

Submitted by:
Nitin Rahuja
2K19/C&I/04

Under the supervision of
Dr. Sudarshan Kumar Babu Valluru

DEPARTMENT OF ELECTRICAL ENGINEERING
DELHI TECHNOLOGICAL UNIVERSITY
(Formerly Delhi College of
Engineering) Bawana Road, Delhi-
110042

2021

DEPARTMENT OF ELECTRICAL ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of
Engineering) Bawana Road, Delhi-
110042

CANDIDATE’S DECLARATION

I, Nitin Rahuja, Roll No. 2K19/C&I/04, M.Tech. (Control & Instrumentation), hereby declare that the project Dissertation titled “CONVOLUTIONAL NEURAL NETWORK BASED CLASSIFICATION OF THREE CARDIAC CONDITIONS” which is submitted by me to the Department of Electrical Engineering, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology, is original and not copied from any source without proper citation. This work has not previously formed the basis for the award of any Degree, Diploma, Associateship, Fellowship, or any other similar title or recognition.

NRahuja

Place: Delhi **NITIN RAHUJA**

Date: 31/07/2021

DEPARTMENT OF ELECTRICAL ENGINEERING

DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of
Engineering) Bawana Road, Delhi-
110042

CERTIFICATE

I, Nitin Rahuja, Roll No. 2K19/C&I/04 student of M. Tech. Control and Instrumentation (C&I), hereby declare that the dissertation/project titled **“CONVOLUTIONAL NEURAL NETWORK BASED CLASSIFICATION OF THREE CARDIAC CONDITIONS”** under the supervision of Dr. Sudarshan Kumar Babu Valluru of Electrical Engineering Department, Delhi Technological University, Delhi in partial fulfillment of the requirement for the award of the degree of Master of Technology has not been submitted elsewhere for the award of any degree.

.

Place: Delhi
Date: 31/07/2021

Prof. Sudarshan Kumar Babu Valluru
(SUPERVISOR)

Department of Electrical Engineering
Delhi Technological University

ACKNOWLEDGEMENT

I am highly grateful to the Department of Electrical Engineering, Delhi Technological University (DTU) for providing this opportunity to carry out this project work. The constant guidance and encouragement received from my supervisor Prof. Sudarshan Kumar Babu Valluru of the Department of Electrical Engineering, DTU, has been of great help in carrying my present work and is highly acknowledged with reverential thanks. I would also like to thank my colleagues and classmates for their guidance and continuous support in the completion of this project work. Finally, I would like to express gratitude to all faculty members of the Electrical Engineering Department, DTU for their intellectual support throughout the course of this work.

NITIN RAHUJA
2K19/C&I/04
M. Tech. (Control and Instrumentation)
Delhi Technological University

ABSTRACT

Clinicians and healthcare professionals use Electrocardiogram (ECG) as a heart monitoring test for identifying unusual cardiovascular activity. The abnormal cardiac behavior is dependent on the heart's electrical activity. Detection of real-time and precise ECG anomalies will offer helpful knowledge for providing possible treatment for patients. On a fundamental level, ECG is a time-series signal generated because of the heart's electrical activity. Different techniques have been formulated to apply Machine learning algorithms for the classification of these time-dependent signals i.e. ECG. Machine learning techniques involve manual extraction of the features thus, leading to issues such as irregularity in the extracted features along with irregularity found in the ECG features. Deep learning methodologies make use of the capabilities of Convolution Neural Networks (CNNs) i.e., they provide the benefits of automatic extraction and identification of complex and intricate features from images. Along with that, pre-trained networks which are trained using some different datasets are successful in learning and extracting features associated with new data. Using and modifying such pre-trained networks can produce desired results. AlexNet, GoogLeNet, and SqueezeNet are some of the popular CNN models, used in various classification tasks using transfer learning.

In this scope, performance evaluation and comparison of these three CNN models for multi-class classification of ECG signals is presented in this thesis. Three distinct classes of ECG signal i.e. Arrhythmia (ARR), Congestive Heart failure (CHF), and Normal Sinus Rhythm (NSR) are considered in this work, which is representative of an individual's heart conditions. The classification methodology adopted in this thesis focuses on Continuous Wavelet Transform (CWT) which produces images having time-frequency information of available 1-dimensional samples of ECG signals. ECG dataset was collected from different PhysioBank databases. Images with time-frequency representation of ECG signals were applied as input to three CNN models. Using transfer learning approach and modification in certain layers of three models, ECG classification was performed and the performance of three DNN architectures was studied. The results revealed promising performance by three models with different internal architectures. Classification accuracies in the range of 97.22% to 97.78% were obtained. AlexNet outperformed GoogLeNet and SqueezeNet models in terms of accuracy as well as training time.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

1.1 MOTIVATION

Cardiovascular Diseases (CVD) causes for large number of deaths around the globe. Electrocardiogram (ECG) signal analysis helps investigate and interpret an individual's heart condition. An electrocardiogram is a bioelectric signal representative of heart’s electrical activity with respect to time. It is a crucial diagnostic tool for assessing the heart’s function and associated activities. The electrical potential generated by Sino Atrial (SA) node, can be recorded as an ECG signal using electrodes, placed on the body at few particular locations. The task of ECG signal classification is beneficial for medical community. Thus, an automated system that classifies the ECG signals into three different categories is proposed in this work. Convolutional Neural Network (CNN) along with a specific signal processing technique were utilized in the study, for developing a computerized ECG diagnostic system.

The ECG signal interpretation comes under the ambit of pattern recognition. Automatic categorization of input into one of several different classes is the main application of pattern recognition. An accomplished heart specialist can easily analyze different heart-related problems by taking a look at the ECG waveforms with naked eyes. In some particular circumstances, complex ECG analyzers have a greater level of accuracy than that of specialists; however, presently, there exists a collection of ECG waveforms that are too complex to be identified even by making use of computer-based techniques. Therefore, the utilization of computerized examination of ECG signal waveforms can significantly reduce the medical practitioner’s workload. A few ECG analyzer devices aids the medical specialist by providing a direct diagnosis while some others give a set of parameters for analysis to infer about the heart condition.

1.2 ANATOMY OF HUMAN HEART

Located slightly left of center in the chest, the human heart is a fist sized organ. The heart acts as a pump, supplying blood to various body parts. The structure of heart consists of a right and a left side, ensuring that the blood containing oxygen doesn’t get mixed with oxygen-less or oxygen-poor blood. Cardiovascular System includes the heart and blood vessels, collectively responsible for pumping and transporting the blood throughout the body. After the exchange of oxygen between the blood and various body parts, oxygen-poor blood is received back by the heart. Close to 4.7 liters of blood is pumped by the heart every 60 seconds. Every day, the heart generally beats about 100,000 times.

The right atrium and right ventricle, present in the right side of the heart, gathers and pushes the blood towards the lungs via pulmonary arteries. The lungs provide oxygen to blood making it oxygen-rich. Such oxygenated blood travels towards the left atrium and left ventricle, present in the left side of the heart, from where it gets pumped to the body for supplying oxygen. There 4 valves present inside the heart responsible for the appropriate movement of blood within the heart. The 4 valves are: Pulmonary valve, Aortic valve, Tricuspid valve, and Mitral valve which act as one-way doors i.e. they get open only in one particular direction only when a force is applied on it. Each valve completes one cycle of opening and closing per heartbeat or every one second.

The heart contracts and relaxes while beating. Systole is referred to as contraction of the heart whereas; Diastole is referred to as relaxing of the heart. During systole period, the ventricles are contracted to force blood in the vessels to go into the lungs and body. Contraction of the right ventricle occurs a bit before the left ventricle. Subsequently, relaxation of ventricles occurs during the diastole period and they get filled with blood coming via the left and right atrium, and then, this cycle repeats. Coronary Arteries are present around the heart and supply blood to heart muscles. Also, an electrical wiring type of system is present in the heart which keeps it pulsating. Originating from the right atrium, the electrical impulses travel towards the ventricles via specialized areas. In such a manner, the heart beats in a planned manner and under normal rhythm which in turn circulates the blood throughout the body.

1.3 THE CARDIAC CONDUCTION SYSTEM

The cardiac or heart conduction system, as shown in Fig. 1.1, is mainly accountable for generating and circulating the action potentials, which cause contraction of heart muscles resulting in pumping of blood. Such action potentials/small electric signals can be captured using electrodes. Such electrodes are placed at certain specific locations on the body using which, a holistic picture of overall electrical activity can be obtained in the form of an image or graph, known as an Electrocardiogram (ECG), or an EKG.

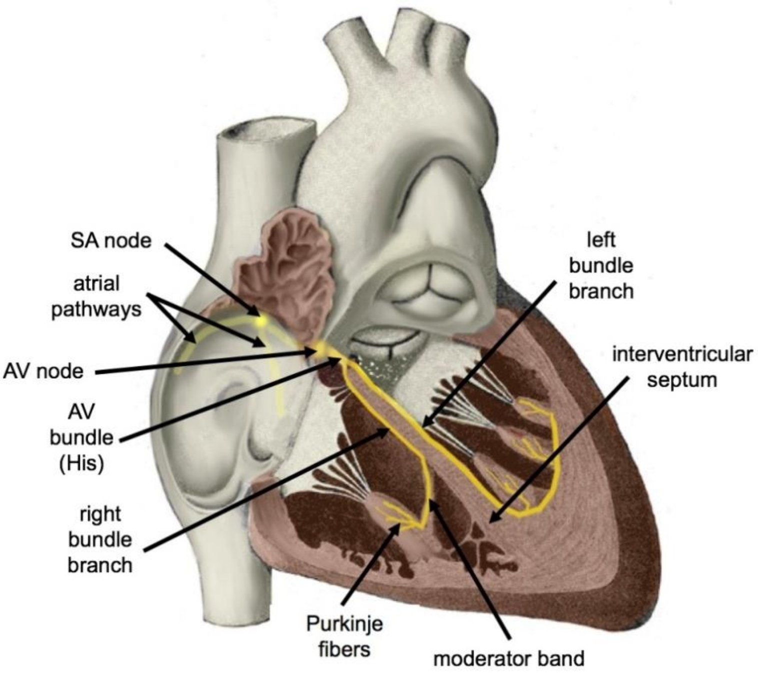


Fig. 1.1 Cardiac Conduction System

The sequence of events that occurs during a single heartbeat and how these events show up on an ECG are now described. Each beat begins in the right atrium with an action potential signal from the heart's natural pacemaker known as the Sino Atrial (SA) node. The signal or impulse then travels through both left and right atrium resulting in depolarization and contraction of muscle cells and bring on a phase called Atrial Systole. The P wave in ECG corresponds to this atrial depolarization. PR segment after the P wave represents the conduction period characterized by atrial systole and before the contraction of ventricles. After departure of the signal from atria, it reached ventricles via atrioventricular or AV node. The impulse then reaches the bundle of His and spreads through it’s branches and Purkinje fibers which are present along the ventricle walls. Subsequently, as the signal propagates via ventricles, rapid depolarization and contraction of contractile fibers are observed which leads to ventricular systole, represented by the QRS complex in the ECG. Finally, ventricular diastole is observed which is a result of relaxation and recovery of ventricular walls when the electric impulse exits the ventricles. This ventricular repolarization is represented with a dome-shaped T wave on the ECG. The ST segment depicts the period when ventricles are depolarized. The QT interval represents the total time taken for repolarization and depolarization of the ventricles.

The sequence of above mentioned actions, repeats every cycle of heartbeat, as shown below in Fig.1.2. The ECG signal is representative of the electrical activity of the heart, which itself comprises various action potentials.

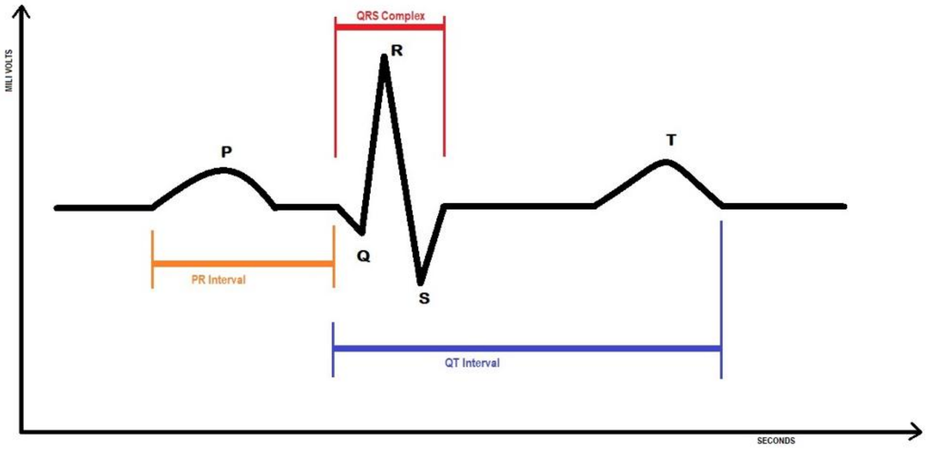


Fig. 1.2 ECG signal per cardiac cycle

1.4 HEART CONDITIONS IN THIS WORK

Deviation from the typical characteristics of the ECG signal can be utilized to analyze various sorts of heart conduction-related issues and arrhythmia. The electrocardiogram can be divided into various intervals and segments, which relate straightforwardly to periods of the heart's conduction; cutoff points can be set on these to analyze irregularity. The doctor regularly utilizes the ECG and different elements to decide the gross state of the heart.

Various heart abnormalities can be diagnosed by analyzing ECG waveforms. This study is aimed at the identification & classification of 3 types of ECG waveforms from the database namely, Arrhythmias (ARR), Congestive Heart Failure (CHF), and Normal Sinus Rhythm (NSR). The following section illustrates the abovementioned heart conditions.

1.4.1 ARRHYTHMIA

Arrhythmia (ARR) is a heart problem that influences the rhythm or rate at which the heartbeats. It occurs when there is improper functioning of electrical impulses that are responsible for directing and regulating heartbeats. As a result, heartbeats either too fast, too slow, too early or too erratically. If medical treatment is not received promptly, severe medical conditions such as heart failure, stroke, and abrupt cardiac arrest can develop. Arrhythmia can be of different types depending upon it's root cause and include Atrial Fibrillation, Supraventricular Tachycardia (or SVT), and Ventricular Fibrillation. Depending on the type of arrhythmia, appropriate treatment is suggested by the doctor. Fig.1.3 (a) shows the ECG signal representative of Arrhythmia.

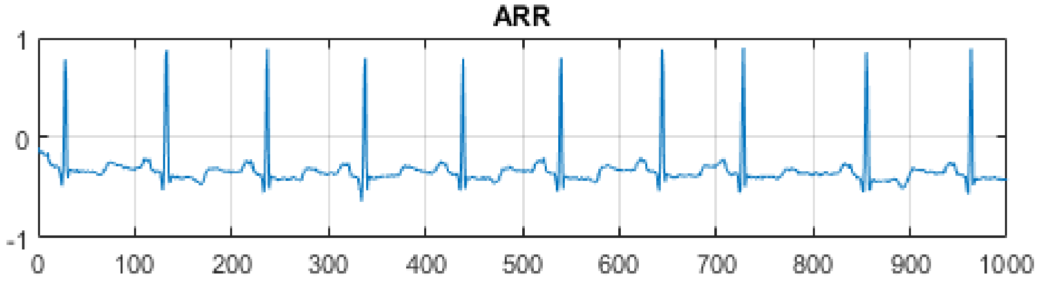


Fig. 1.3 (a) ECG signal corresponding to Arrhythmia

1.4.2 CONGESTIVE HEART FAILURE

Heart failure is a condition in which the heart can't supply enough blood to meet the body's demands. This can happen in two ways, either the heart's ventricles can't pump blood hard enough during systole, called systolic heart failure, or not enough blood fills the ventricles during diastole, called diastolic heart failure. In both cases, blood backs up into the lungs, causing congestion or fluid build-up, which is why it's also often known as congestive heart failure, or just CHF. Congestive heart failure affects millions of people around the world and since it means that the body's needs are not being met, it can ultimately lead to death. Part of the reason why so many people are affected by heart failure, is that there are a wide variety of heart diseases that can impair the heart's ability to pump out blood and over time, can ultimately cause the heart to fail. Fig.1.3 (b) shows the ECG signal representative of Congestive Heart Failure.

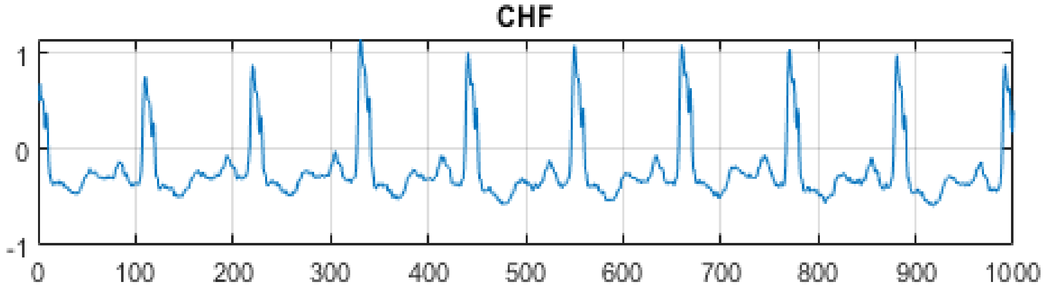


Fig. 1.3 (b) ECG signal corresponding to Congestive Heart Failure

1.4.3 NORMAL SINUS RHYTHM

The normal and efficient electric activity of the heart is referred to as Normal Sinus Rhythm. The heart’s pacemaker or SA node is the place of origination of heart’s electric impulses. The produced electric impulses travel towards the atrioventricular node and subsequently to the ventricles. If P waves aren’t observed in an ECG, then definitely the rhythm is not sinus. PR interval is the duration between QRS complex and P wave and the interval is always constant (generally, less than 200 milliseconds) in normal sinus rhythm. Finally, the ventricles are repolarized which is characterized by the T wave. This interval to repolarization should fall between 400-450 milliseconds after the initiation of the QRS complex and is called the QT interval.

A person can have rates below 60 beats per minute that are still sinus but these are referred to as Sinus Bradycardia and rates that are greater than 100 beats per minute and still sinus which are known as Sinus Tachycardia. Fig.1.3 (c) shows the ECG signal representative of Normal Sinus Rhythm.

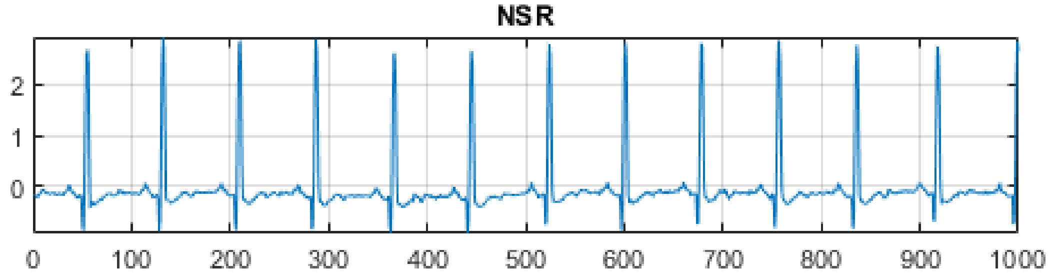


Fig. 1.3 (c) ECG signal corresponding to Normal Sinus Rhythm

1.5 ECG MEASUREMENT

In a standard 12-lead ECG, 10 electrodes are placed on the patient's skin; four of them on the limbs and six of them on the chest wall. One of the four electrodes on the limbs (the one attached to the right leg) is an earth electrode that helps minimize the artifacts so it doesn't record the electrical activity. The other three electrodes are attached to the right arm, left arm and left leg. Three unipolar leads and three bipolar leads are obtained from these three electrodes which look at the heart from different angles. In bipolar leads, the electrical activity of the heart is recorded between two of the three electrodes on the limbs.

The best way to understand the limb leads is using Einthoven's triangle. The bipolar limb leads include: lead one between right arm and left arm, lead two between right arm and left leg, and lead three between left arm and left leg. The unipolar limb leads use only one electrode. This leads are: AVR lead which is attached to the right arm, AVL lead which is attached to the left arm, and AVF lead which is attached to the left leg. The limb leads see the heart in a vertical plane. Leads 1 and AVL look at the lateral wall of the left ventricle. Leads 2, 3 and AVF look at the inferior wall of the heart.

Six more unipolar leads are obtained from the six electrodes that are placed on the chest wall. Those electrodes are: V1 attached to the right side of the sternum in the fourth intercostal space, V2 attached to the left side of the sternum in the fourth intercostal space, V3 midway between V2 and V4, V4 midclavicular line in the fifth intercostal space, V5 anterior axillary line in the fifth intercostal space, V6 mid-axillary line in the fifth intercostal space. The chest leads see the heart in an horizontal plan. Leads V1 and V2 look at the interior wall of the heart, lead V3 looks at the interventricular septum, lead V4 looks at the apex and leads V5 and V6 look at the lateral wall of the left ventricle [1]. Fig.1.4 shows the 12 Lead ECG system for capturing the heart's electrical activity.

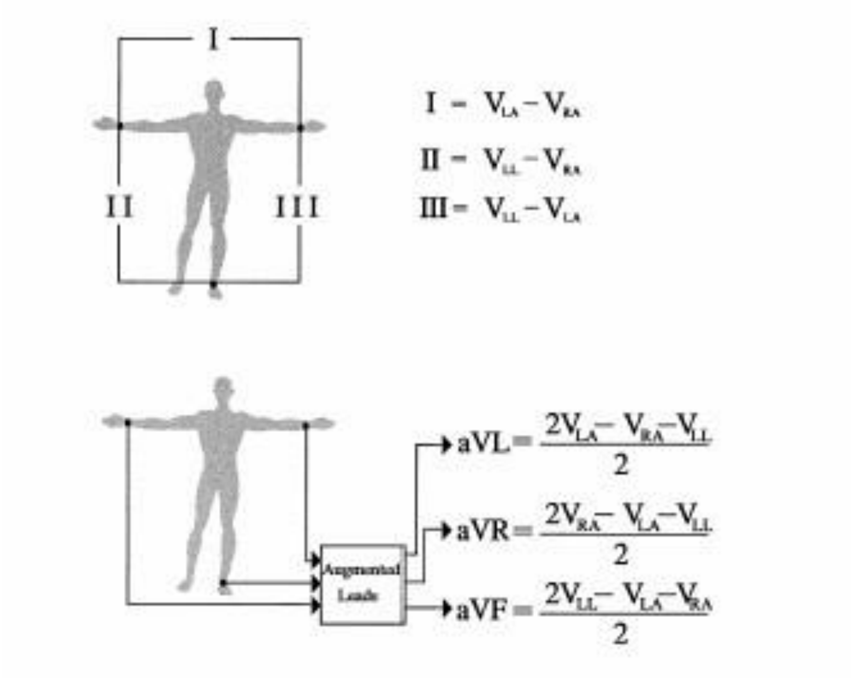

I
II
III
I - $V_{LA} - V_{RA}$
II - $V_{LL} - V_{RA}$
III - $V_{LL} - V_{LA}$
Augmented Leads
aVL = $\frac{2V_{LA} - V_{RA} - V_{LL}}{2}$
aVR = $\frac{2V_{RA} - V_{LA} - V_{LL}}{2}$
aVF = $\frac{2V_{LL} - V_{LA} - V_{RA}}{2}$

Fig. 1.4 The 12 Lead ECG

TABLE 1. TYPES OF LEADS

LEAD TYPES	**COMPONENTS**
STANDARD LEADS	Lead I, Lead II, Lead III
CHEST LEADS	V1, V2, V3, V4, V5, V6
LIMB LEADS	AVL, AVR, AVF

1.6 DATA PREPROCESSING

Generally, an ECG signal contains undesired information along with the main signal as noise. This noise can have different characteristics and kinds. Fig. 1.5 shows the common types of noise signals, which are generally present in ECG signals. Therefore, appropriate pre-processing of ECG signals is required before performing any analysis or examination of ECG signals.

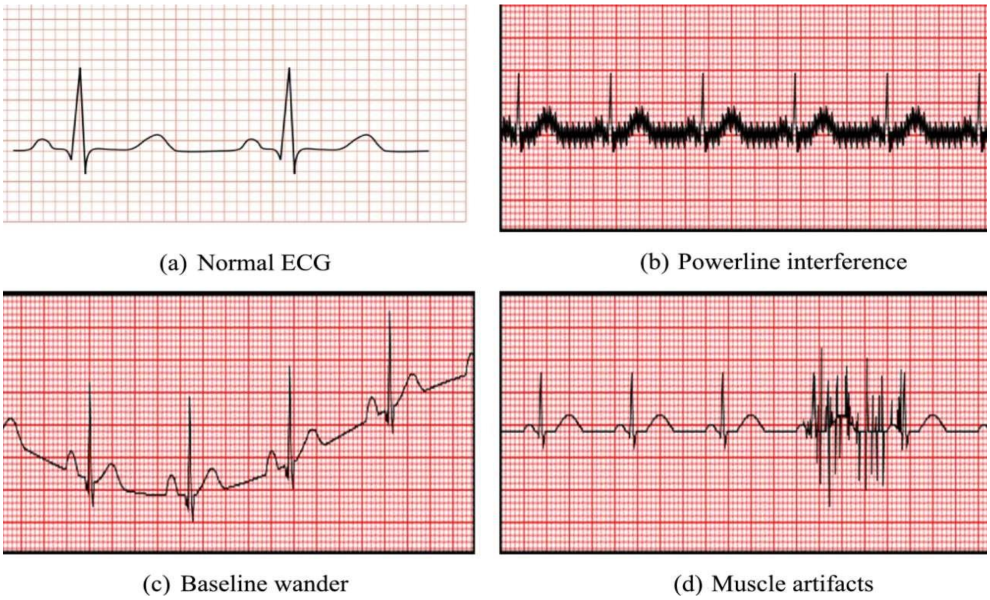


Fig. 1.5 Types of commonly encountered noise in ECG signal

1.7 ECG CLASSIFICATION USING COMPUTER-BASED METHODS

After using appropriate methods for pre-processing, ECG information can be inspected to notice the regular and irregular cardiovascular activities. To precisely distinguish arrhythmia and other heart conditions, various computer-based strategies are being utilized throughout the previous two decades. These strategies are divided into supervised and unsupervised methods.

With the development in the field of artificial intelligence for medical research, various machine learning algorithms like Random Forest, Support Vector Machine (SVM), Long Short Term Memory (LSTM), and Decision Tree are utilized to classify irregularities in ECG signal, e.g., atrial fibrillation, which is a type of arrhythmia. These machine learning-based methodologies involve manual extraction of the features in the images before the classification process. To overcome such shortcomings, Deep Learning, which is a specific branch of machine learning, is utilized. The need for human-assisted feature extraction and identification is eliminated by deep learning. Instead, the learning algorithm automatically learns, extracts, and identifies various intricate features, from the original input data. Fig. 1.6 represents the comparison among machine learning & deep learning techniques.

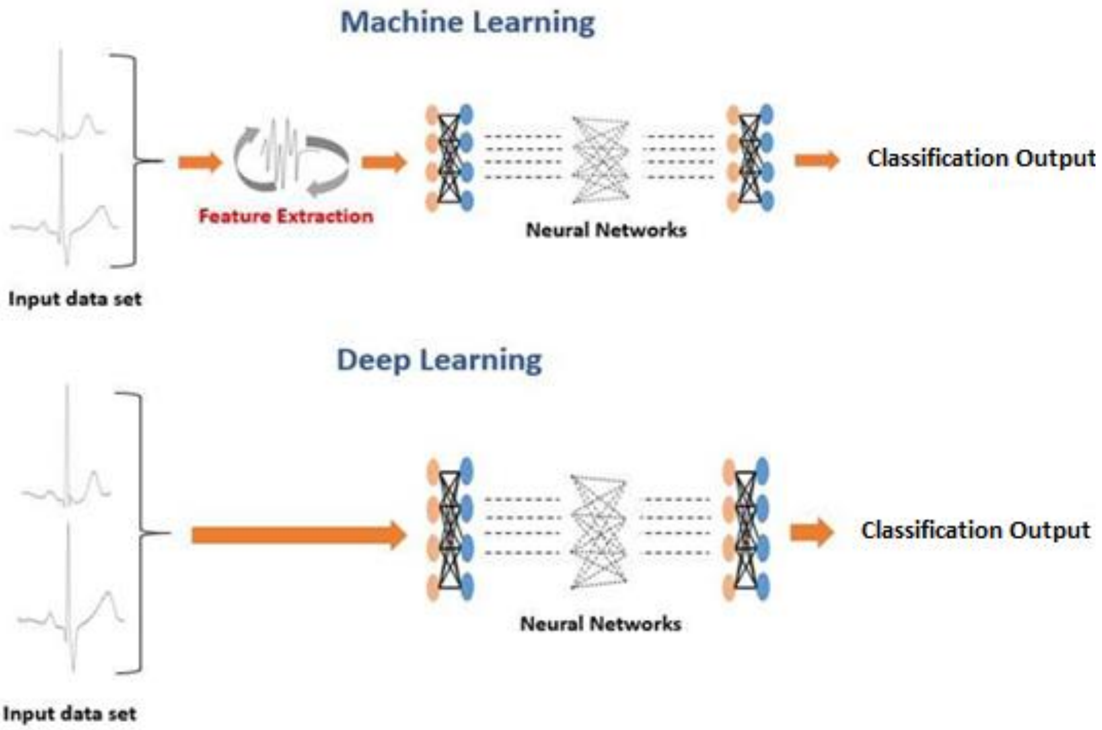


Fig. 1.6 Difference between Machine Learning and Deep Learning

1.8 OUTLINE OF THIS WORK

This work has been divided into 5 chapters which are described below:

1. Chapter 1 gives a brief introduction to the human heart, ECG, and ECG classification.
2. Chapter 2 presents the literature review on ECG signal classification using machine learning and deep learning-based algorithms and associated prerequisites.
3. Chapter 3 describes the deep learning-based methodology utilizing three different architectures adopted in this work.
4. Chapter 4 presents the results and performance comparison of different architectures utilized for multi-class ECG classification.
5. Chapter 5 contains the conclusion and future scope of work related to deep learning-based ECG classification.

CHAPTER 2
RELATED WORK

2.1 OVERVIEW

An overview of the existing work for multi-class classification of ECG signals and detecting arrhythmia and other heart conditions are presented in this chapter. The literature survey consists of machine learning as well as deep learning methodologies as this thesis is confined to the use of deep neural networks for ECG classification purposes.

2.2 LITERATURE REVIEW

Electrocardiogram (ECG) signal analysis helps in investigating and interpreting an individual's heart condition and electrical action. ECG analysis helps in detecting and analyzing the heart’s electric activity, so that appropriate medical action can be taken. However, if appropriate medical treatment is not received promptly, severe medical conditions such as heart failure, stroke, and abrupt cardiac arrest can develop. As a result, the detection of real-time and precise ECG anomalies offers helpful knowledge to physicians and health professionals [3], allowing them not just to avoid cardiac failure but also to provide personalized treatment for patients with heart disease [4]–[5]. Thus, it is critical to correctly detect the anomalies and classify the ECG signal [6].

Manual analysis and visual identification of irregularities in ECG signals require significant training and practice. Identifying slight deviations/ irregularities in ECG is a complex task. As a result, researchers have turned to the study and evaluation of the ECG signals using computer-based techniques to perform their studies. [6][7]-[8].

With the advancements in the emerging field of artificial intelligence for medical research, various machine learning algorithms including Random Forest [9], Support Vector Machine (SVM) [10], and Decision Tree Ensemble [11] are utilized to classify irregularities in ECG signal, e.g., atrial fibrillation. For interpretation of ECG and detecting arrhythmia from the ECG signal, comparative performance analysis of various machine learning-based classification methods have been conducted [12].

These methodologies involve manual extraction of the features in the images before the classification process. To overcome such shortcomings, artificial neural networks are utilized. Particularly, Convolutional Neural Network (CNN) is extensively researched by various researchers for problems involving image identification, image clustering and image classification in various fields. Studies involving Convolutional Recurrent Neural Network (CRNN) [13] have already been carried out. Deep CNN [14] is also explored in some studies.

Deep Learning methods utilize neural networks having multiple layers of neurons. Deep learning accomplishes feature extraction by using many layers, each acting as a processing unit. Each layer for extracting a particular feature takes the output of the previous layer which acts as input [15].

Deep learning methodologies have outperformed previously existing methods in various pattern recognition competitions and have motivated the research community to make use of these latest and growing methodologies in the field of biomedical image processing. In the same context, new research studies using Recurrent Neural Networks (RNN), particularly Long Short Term Memory Networks (LSTM) [16], and Convolutional Neural Network (CNN) [17] have shown promising results.

Automatic extraction and identification of complex and intricate features of images is the main advantage provided by Deep Neural Networks (DNNs) and thus eliminates the requirement of extracting features manually, as required by conventional Machine Learning (ML) methodologies. This advantage provides an opportunity for creating an end-to-end pipeline, having ECG signal at the input end and the result of classification at the output. ECG classification task can be binary or multiple. However, more and more accuracy and optimum output is possible to achieve only when a sufficient amount of data is available for extraction of intricate ECG features, and learn from a variety of available inputs.

Even though there are the benefits of using DNNs for ECG signal analysis and classification, there is one reason predominantly hampering the wide utilization of these methods i.e., the hindrance due to the availability of volume of data required to learn and perform. When contrasted with the conventional classification approaches based on ML, a humongous amount of data is required by DNNs for training. Such an issue leads to a gap between sufficient intricate ECG features and the size of the dataset, due to the lack of volume of the datasets that are openly accessible in this domain [17].

To overcome the above-mentioned issue, this thesis presents an automatic ECG classification method utilizing wavelet transform and transfer learning with DNNs. Particularly, 1-dimensional ECG signal samples were converted to 2-dimensional data i.e. images (Scalogram), and pre-trained DNN models were used for the classification task. Instead of training the DNNs from scratch with ECG images, architectures pre-trained on image data associated with image classification and object recognition (1000 classes like a chair, mouse, table lamp, etc.) are utilized. Due to the availability of huge datasets in these domains, efficient training & extraction of feature maps representative of complex features and patterns in the images is possible. Such learned and available feature maps were transferred for ECG classification purposes, by training and optimizing AlexNet, GoogLeNet, and SqueezeNet on a small dataset of 2-dimensional images of ECG signal.

2.3 CONCLUSION

This chapter describes the literature review in the field of ECG classification using various machine learning and deep learning methodologies. The benefits of using deep learning methods over machine learning methods along with their performance were also discussed. The importance of availability of data in humongous amounts as required by various deep learning methods for optimal performance was also touched upon.

CHAPTER 3

METHODOLOGY AND CONFIGURATION FOR MULTIPLE CLASS ECG CLASSIFICATION

3.1 ECG DATABASE AND PRE-PROCESSING

The dataset utilized in this work is a collection of modified versions of the files obtained from the following PhysioNet databases:

- MIT-BIH Normal Sinus Rhythm Database [18]
- MIT-BIH Arrhythmia Database [18], [19]
- The BIDMC Congestive Heart Failure Database [18] [20]

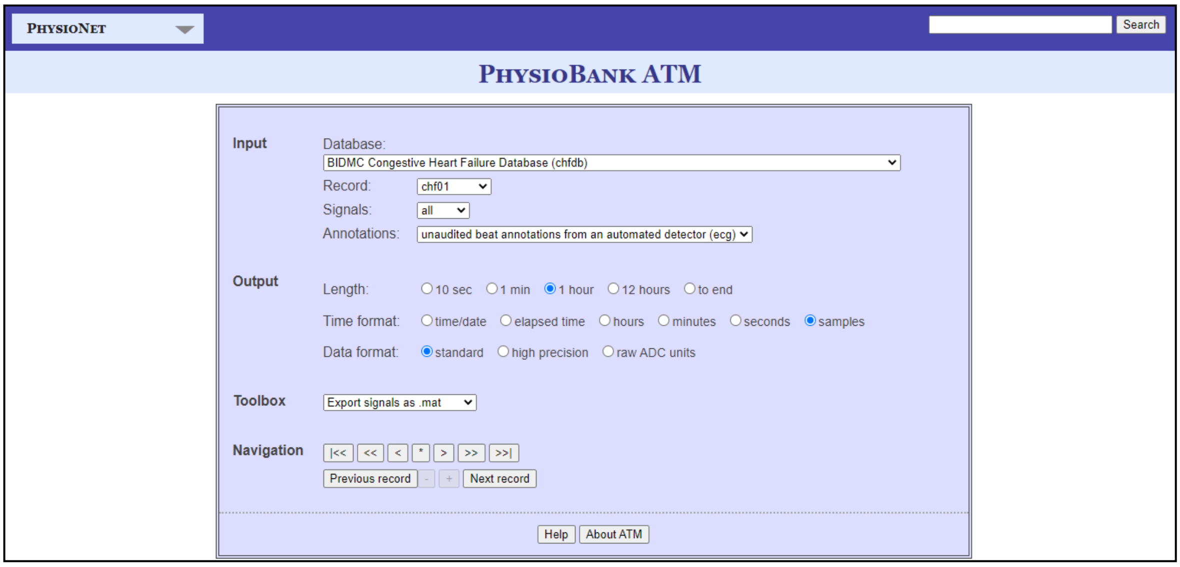
PhysioNet is a platform providing freely accessible pool of recorded physiological signals data. The Members of the MIT Laboratory (Computational Physiology) manage and maintain the PhysioNet platform. The PhysioNet mainly consists of three modules:

1. PhysioBank: An archive of well-structured digital recordings of various physiological signals for research & development.
2. PhysioToolkit: “An extensive and growing library of software for physiological signal processing and analysis, detection of physiologically significant events using classical techniques and novel methods, interactive display and characterization of signals” [2].
3. Various tutorials and educational material.

The raw data files were extracted from PhysioBank ATM with selected output settings as shown in Fig. 3.1 (a). After obtaining the raw data files, each raw data file underwent the following sequence of processing steps to finally curate the ECG dataset, which will be used for the study in this research.

1. Resampling of data to 128 hertz.
2. Two ECG recordings present in each data file were divided into two individual data records.
3. Finally, after truncation of the data length, a common length of 65536 samples per record was chosen.

The ECG dataset obtained after the abovementioned processing steps, was a MATLAB structure array consisting of two different fields, namely: Data and Labels. A matrix with 162 rows and 65536 columns is Data whereas, Labels is a 162x1 array of character vectors. Fig. 3.1 (b) shows plots of three randomly selected records from the ECG dataset having 3000 samples each.



PhysioBank ATM

Input

Database: BIDMC Congestive Heart Failure Database (chfdb)

Record: chf01

Signals: all

Annotations: unaudited beat annotations from an automated detector (ecg)

Output

Length: ○ 10 sec ○ 1 min ● 1 hour ○ 12 hours ○ to end

Time format: ○ time/date ○ elapsed time ○ hours ○ minutes ○ seconds ● samples

Data format: ● standard ○ high precision ○ raw ADC units

Toolbox: Export signals as .mat

Fig. 3.1 (a) PhysioBank ATM

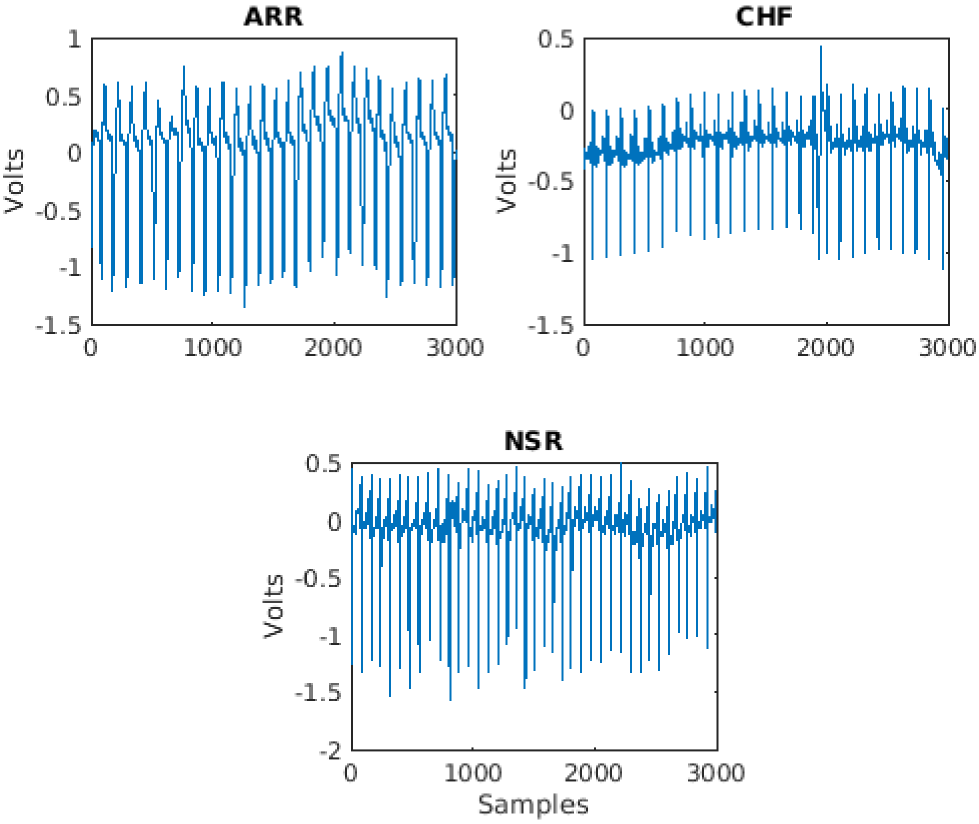


Fig. 3.1 (b) Plots of three random ECG signals of type ARR, CHF, and NSR from the ECG dataset

The 48 data files present in the MIT-BIH Arrhythmia Database [18] [19] were modified to obtain ECG samples present in the first 96 rows of the ECG dataset (162 x 65536 matrix) with corresponding labels as “ARR”. Similarly, the two ECG recordings represented by the 15 data files of BIDMC Congestive Heart Failure Database [18] [20] were modified to obtain the next 30 rows in the ECG dataset with corresponding labels as “CHF”. Finally, the two ECG recordings represented by the 18 data files of MIT-BIH Normal Sinus Rhythm database [18] were modified to obtain the last 36 rows in the ECG dataset with corresponding labels as “NSR”

3.2 CONTINUOUS WAVELET TRANSFORM

ECG is an electrical signal which exhibits non-stationary behavior i.e. it’s frequency component changes with time. Just utilizing the temporal or spatial portrayal using Fourier Transform cannot successfully represent the ECG signal's complete information. Therefore, to investigate such non-stationary behavior possessed by ECG, Continuous Wavelet Transform (CWT) [21] which is a time-frequency transform, is utilized in this work. Localized analysis of significant signals is another advantageous feature provided by wavelet analysis. For a continuous-time signal $f(t)$, the mathematical expression for CWT is defined as given in (3.1) below.

$$F(\tau, s) = \frac{1}{\sqrt{|s|}} \int_{-\infty}^{+\infty} f(t)\psi^*\left(\frac{t-\tau}{s}\right)dt \tag{3.1}$$

The transformed signal $F(\tau, s)$ is a function of “τ” and “s” i.e., translation/window parameter and scaling parameter, respectively. “ψ^*” represent complex conjugate of the wavelet function. Analytic Morlet (Gabor) wavelet, shown in Fig. 3.2, is utilized in this work.

CWT is used to analyze the correlation between the ECG signal and the wavelet function to obtain wavelet coefficients. All the obtained coefficients are then arranged, resulting in a Two-Dimensional (2D) representation known as Scalogram which is plot of coefficients arranged as a function of time and frequency. The wavelet coefficient can be calculated with the help of the expression given in (3.2).

$$|F(\tau, s)| = \sqrt{F^2(\tau, s)} \tag{3.2}$$

Due to the complex-valued nature of wavelet coefficients, the coefficients can deliver information about the phase and amplitude of the signal being analyzed. Analytic wavelets are suitable for analyzing frequency content in a practical non-stationary signal as a function of time. In this manner, raw ECG signal samples were processed to obtain corresponding color varying 2D CWT Scalograms.

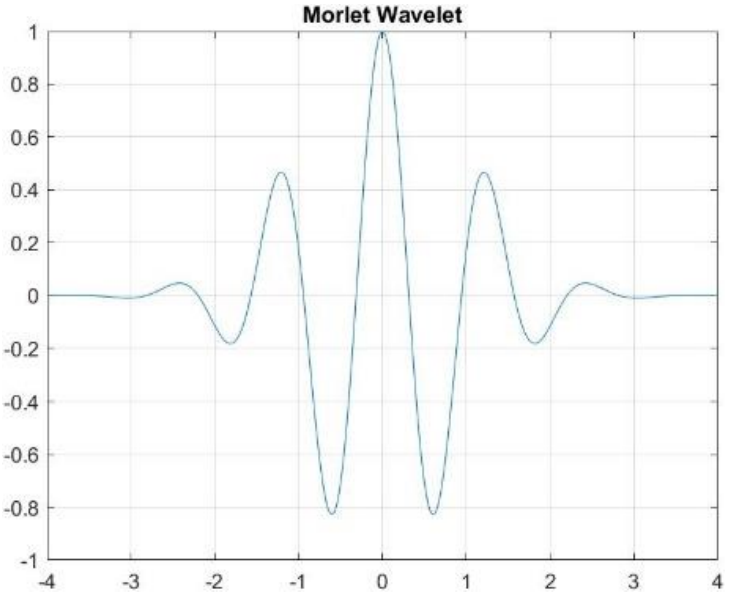


Fig. 3.2 Morlet Wavelet

Thus, using the ECG dataset having ECG signal samples, corresponding 2D CWT scalograms were obtained for all the 1D ECG signals of three heart conditions under consideration. The 2D Scalograms were obtained using MATLAB, to feed them as input for the next stages of the classification task, particularly for Convolutional Neural Network models considered in this work. Fig. 3.3 shows an ECG signal representative of Arrhythmia heart condition, and its corresponding scalogram obtained using CWT.

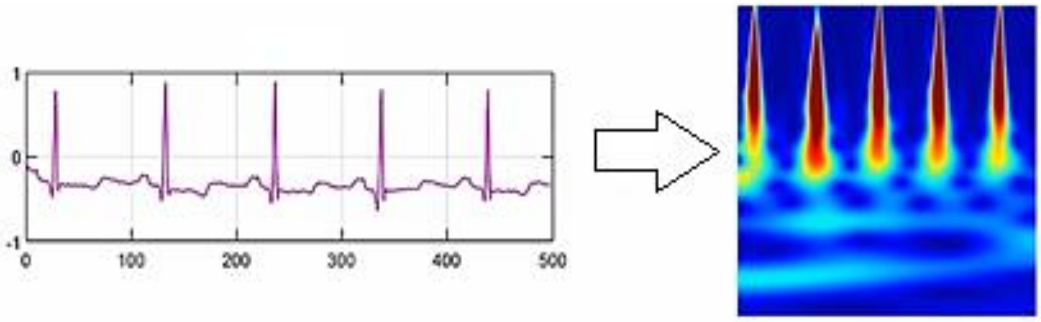


Fig. 3.3 Scalogram of ECG signal corresponding to Arrhythmia (ARR)

3.3 NEURAL NETWORK

Inspired by the design and functioning of neurons, Neural Networks are made up of small mathematical processing units called neuron, arranged in a layer-by-layer format and enables various deep learning methodologies. On a functional level, neural networks accept data, and succeesflly identify various shapes and features in the data, and finally provides outputs for previously unseen but same type of data. Specifically, there are three different types of layers present in neural networks.

The input values are accepted using an Input layer, and then there are hidden layers for applying different weights associated with different layers to the input data. Output values are obtained by using an activation function, and finally, the results are obtained using the output layer [22].

Let's understand how this is done by looking at the structure of the neural network as shown in Fig. 3.4.

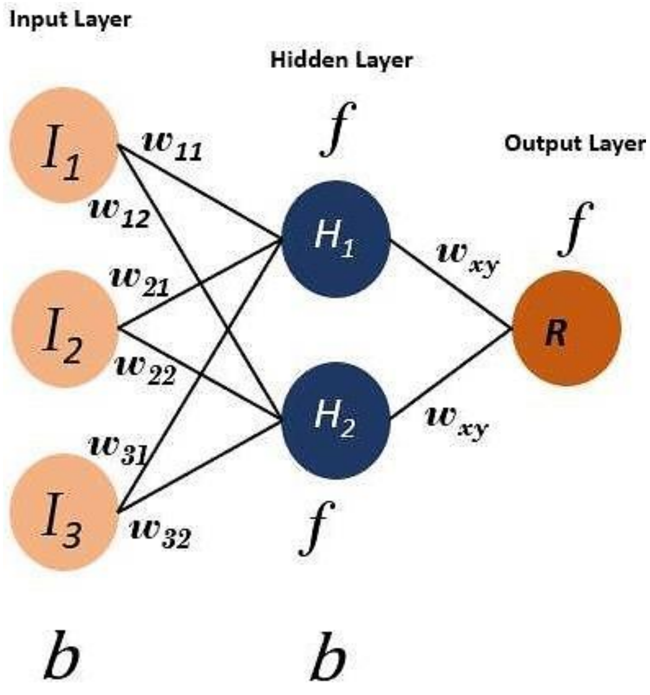


Fig. 3.4 A simple neural network

A neural network consists of various layers of neurons. Neurons are the core processing units of the neural network. First, we have the input layer which receives the input (I_1, I_2, and I_3) which are input values from a data set. The output layer (R) predicts our final output. In between the input and output layer, there exist the hidden layers (H_1, H_2) which perform most of the computations required by the neural network.

Neurons of one layer are connected to neurons of the next layer through channels. Each of these channels is assigned a numerical value known as Weight (w_{11}, w_{12}…w_{33}, w_{xy}). The inputs are multiplied by the corresponding weights and their sum is sent as input to the neurons in the hidden layer. Each of these neurons is associated with a numerical value called the Bias (b), which is then added, to the input sum. This value is then passed through a threshold function called the Activation Function (f). The result of the activation function determines if the particular neuron will get activated or not. An activated neuron transmits data to the neurons of the next layer over the channels. In this manner, the data is propagated through the neural network. This is called forward propagation. In the output layer, the neuron with the highest value determines the output. These values are nothing but a probability of belongingness to a particular class. The output with the highest probability is the output predicted by the neural network.

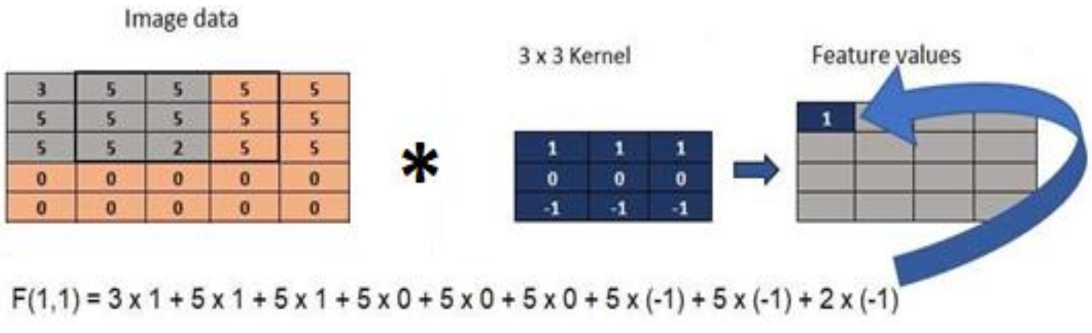
Further, a comparison of predicted output and the actual output is done computer error in prediction. The value of error represents how inaccurate the neural network is in predicting the output and the sign of error (positive or negative) suggests that the predicted values are higher or lower than expected.

This error information is then transferred backward through the network to modify weights for improving the learning process. This is known as Back Propagation. Now based on this error information, the weights are adjusted. This cycle of forward propagation and backpropagation is iteratively performed with multiple inputs. This process continues until the weights are assigned such that the network can predict the output correctly in most cases. This brings the training process to an end.

3.4 CONVOLUTION NEURAL NETWORKS

The focus of this project is to classify ECG signals representative of three different heart conditions into respective classes through training and learning three neural network models namely: AlexNet, GoogLeNet, and SqueezeNet. Particularly, this project utilizes models based on Convolutional Neural Networks (CNNs) having different internal architectures. Generally, CNNs consist of convolution and pooling layers which are responsible for the extraction of significant features out of the images [23].

For performing the classification task, a three-dimensional tensor is required by the CNN as input. The three-dimensional tensor is an image having information about its height, width, and RGB (Red, Green, and Blue) channels value. Additional processing is performed at various stages on this input image as it proceeds towards different layers. The convolution process begins by positioning a kernel (e.g. A 1x1 or 2x2 or 3x3 matrix used as a filter) on the first pixels from the top-left position of the input image, such that it fits the kernel size. Subsequently, corresponding pixel-wise multiplication of kernel and input image is performed and aggregated to obtain a resultant value in the obtained feature map. The feature map has less dimensionality than the input image and contains the details of input image features. Fig. 3.5 shows the calculations as described above, with an input image of 5x5 pixels and a kernel size of 3x3 pixels.



3	5	5	5	5
5	5	5	5	5
5	5	2	5	5
0	0	0	0	0
0	0	0	0	0

1	1	1
0	0	0
-1	-1	-1

$F(1,1) = 3 \times 1 + 5 \times 1 + 5 \times 1 + 5 \times 0 + 5 \times 0 + 5 \times 0 + 5 \times (-1) + 5 \times (-1) + 2 \times (-1)$

Fig. 3.5 Extracting Feature value in CNN

After obtaining the feature value of the feature map, the kernel slides towards the right on image data in step increments of 1 known as stride.

$$F(i, j) = \sum_p \sum_q K(p,q) \cdot f(i-p, j-q) \tag{3.3}$$

The above equation (3.3) represents the convolution operation where the input image is represented by " f " having dimensions i x j, "K" represents a kernel of size m x n, and "F(i, j)" is the final feature map or the output convolution matrix [24]. Generally, the following components are present in any CNN model:

1. Activation Function,
2. Pooling Layers,
3. Fully Connected Layers,
4. Dropout, Flatten layers, and
5. Output Layers.

3.4.1 ACTIVATION FUNCTION

Generally, in CNN, the presence of an activation function after every convolutional layer is commonly observed. One of the common & popular option of activation function for most CNN models is Rectified Linear Unit (ReLu). When negative values are present in the input feature matrix, ReLu makes it zero [25]. Mathematically, the ReLu activation function is defined in (3.4) below and is shown in Fig. 3.6.

$$g(x) = \begin{Bmatrix} 0, & for \quad x < 0 \\ x, & for \quad x \geq 0 \end{Bmatrix} \tag{3.4}$$

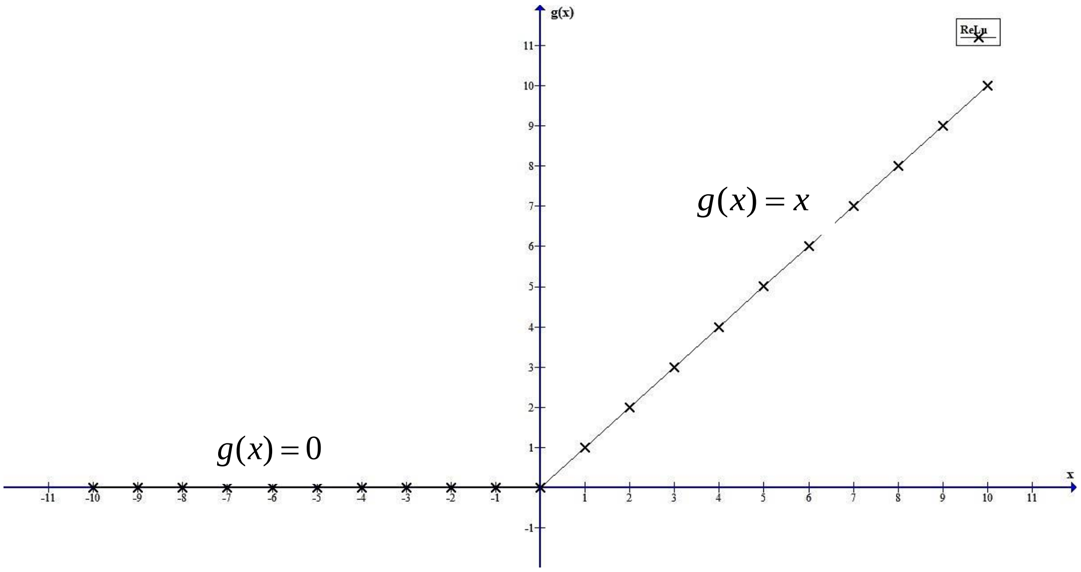


Fig. 3.6 Rectified Linear Unit (ReLu) Activation Function

Another popular activation function used in CNNs is known as the softmax activation function. The mathematical expression for softmax is described in (3.5) and it's graph is shown in Fig. 3.7.

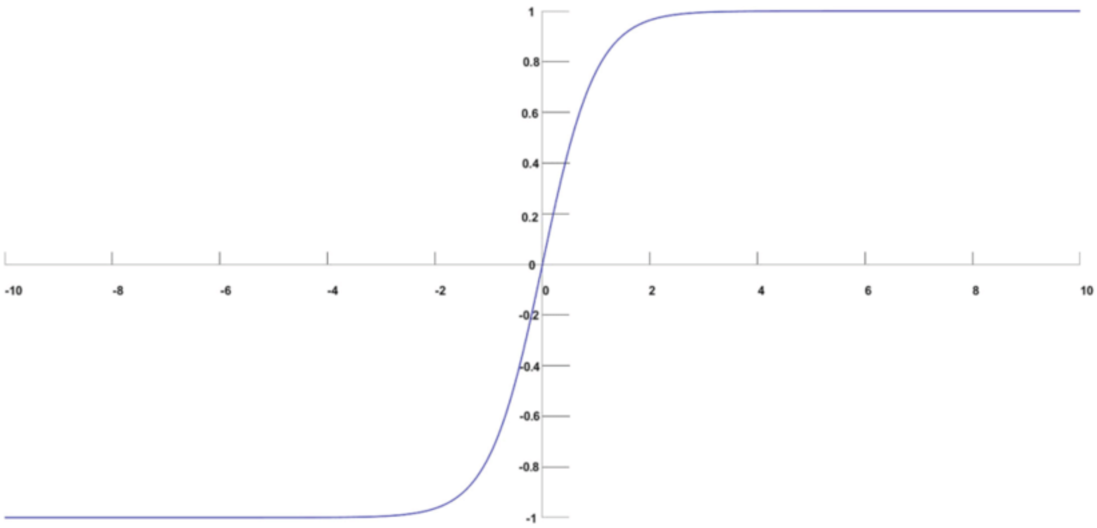


Fig. 3.7 Softmax Activation Function

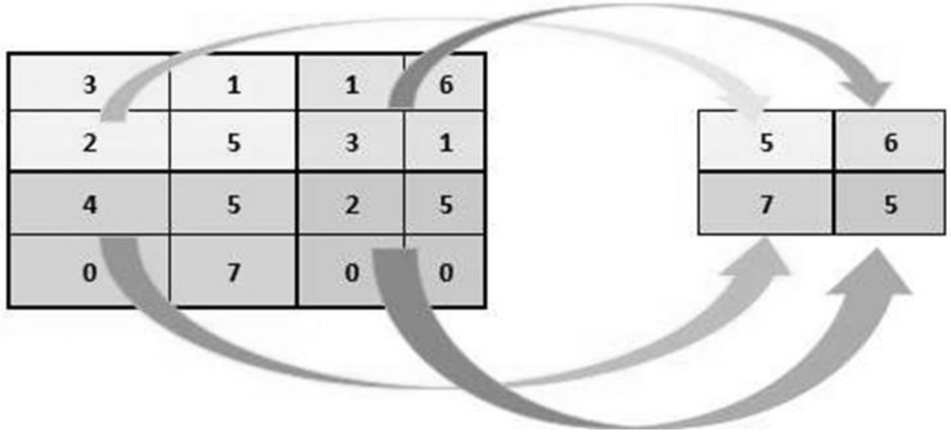
$$\sigma(z_i) = \frac{e^{z_i}}{\sum_j^k e^{z_j}} \tag{3.5}$$

3.4.2 POOLING LAYERS

Reduction of the feature map's spatial dimensions is the task performed by the pooling layer in CNNs. Pooling layers can be seen after convolution layers in various CNN architectures.

Fig. 3.8 shows, the conversion of a *4x4* matrix into a feature value set of *2x2*. Pooling layers assist in dodging over-fitting problems as pooling layers decreases the number of parameters involved in a CNN. When CNNs are trained on original details along with noisy information, it results in the problem of over-fitting in CNN models. Such a problem harms the classification capabilities of CNN [26]. Although, pooling layers down-samples input features and decreases the dimensions of feature maps, they still extract and remember important features of the input image to feed further layers in the network.

In this project, Max pooling operation using a 2 x 2 window is used. In this operation, the maximum value from every group is chosen as a feature input for final set of feature values, as shown in Fig. 3.8.



3	1	1	6
2	5	3	1
4	5	2	5
0	7	0	0

5	6
7	5

Fig. 3.8 Max pooling operation steps

3.4.3 FULLY CONNECTED LAYER

Fully connected layers takes in feature map from the output of the convolution and pooling layers. In fully connected layers, each of the nodes (i.e. neuron) is connected to all the nodes in the adjacent layer. The fully connected layer can also be thought of as a simple feed-forward neural network. Dense layer is another name given to the fully connected layer. Fully connected layers utilize the significant features extracted from previous layers to perform early classification on the input data fed to a neural network [25].

3.5.4 DROPOUT AND FLATTEN LAYERS

To avoid the over-fitting problems, dropout layers are employed for dropping some of the information associated with certain neurons in the neural network [27]. On the other, before the final output layer, flatten layers are used to transform the multi-dimensional output into a vector of a specific size [28]. The use of flatten layers becomes compulsory to get the output shape as per the number of classes trained and predicted. For example, while training a CNN, the input data consisted of three categories, then the flatten layer is required to reduce the fully connected layer's result to a vector of 3 values, one corresponding to each of the categories. In the same way, a larger vector is generated in case more categories are present in the classification task at hand.

3.4.5 BACKPROPAGATION AND GRADIENT DESCENT

In general, image data is fed as input to a CNN which propagates in a forward direction starting from convolution layers towards fully connected layers. After that, output values are computed for various training data classes. During the first iteration, the output values obtained are generally random because the weights are also assigned randomly. In such a situation, the error between predicted output and actual output is calculated. Error is then backpropagated by calculating the associated gradients using the weights of the previous layer of CNN. Subsequently, parameters defining the CNN are then updated by utilizing gradient descent. Gradient descent is an iterative optimization algorithm that minimizes a given objective function to its local minimum value.

After training of CNN model, the classification process takes place and the network predicts the output. Considering, the prediction accuracy of the CNN is given by "p" and desired CNN model accuracy by "y". Therefore, for a set of "n" predictions if the predicted values are "p" and the desired target is "y", then the error can be calculated as shown in (3.6) below.

$$E = \frac{1}{2}\sum_{n}(y - p)^2 \tag{3.6}$$

This error value "***E***" is minimized by adaptively changing the weights in such a manner that "***p***" and "***y***" values approach to each other. This error minimization is achieved using the backpropagation algorithm, which alters the different weight values using the methods of gradient descent. These methods, makes use of the partial derivatives of the weight ($\boldsymbol{w_{xy}}$) values with reference to error (***E***) values as shown in (3.7) below.

$$\frac{\partial E}{\partial Wx, y} \tag{3.7}$$

3.5 TRANSFER LEARNING WITH DEEP CONVOLUTION NEURAL NETWORK

A pre-trained CNN model is a saved architecture trained on an enormous dataset for classification purposes. Transfer learning is the process of utilizing this existing knowledge of a pre-trained CNN model for the classification of dissimilar but related classes of objects [29].

As we move through the different layers of a CNN model, the complexity level of features extracted by layers keeps increasing and finally, the last few layers predicts the outcome. Therefore, a significant number of layers of a pre-trained CNN model, are useful in extraction of similar patterns in the new application datasets. In some cases, the previously trained CNN model is directly used on a completely new, but related dataset.

Transfer learning eliminates the need to build the CNN model from scratch and reduces the training time significantly. Fig. 3.9 shows how traditional machine learning is different as compared to the transfer learning approach.

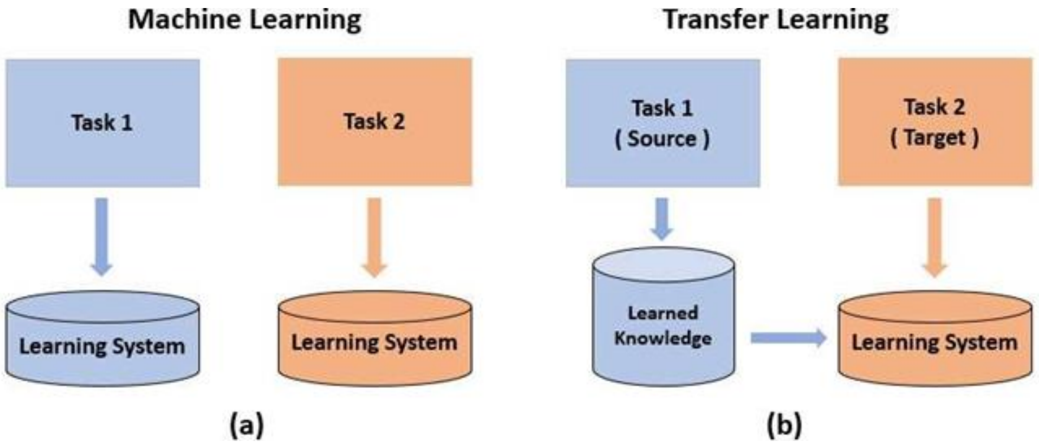


Fig. 3.9 (a) Machine learning versus (b) Transfer learning

This project utilized three pre-trained CNN models, initially trained on more than a million images of the ImageNet [30] database. ImageNet data consists of 1000 different classes. The three CNN-based models were re-trained with the ECG scalogram dataset having three classes. To make the CNN models suitable for ECG classification, the final few layers of CNN models were replaced and tuned. Particularly, the replaced and fine-tuned layers included one or more from: dropout layer, classification layer, fully connected layer, and softmax layer. The following sections describe the three CNN models utilized in this work.

3.5.1 ALEXNET

AlexNet is a convolutional neural network (CNN), developed by Alex Krizhevsky [31] and gained popularity after competing and winning in the ImageNet Challenge in 2012 (ILSVRC-

2012) with a significant edge. AlexNet is an 8 layers deep convolutional neural network. More than 1 million images were used for training AlexNet, and testing was done on 150,000 ImageNet dataset test pictures. Consequently, it was successful in classifying images into 1000 different categories of ImageNet database (such as cat, dog, table, etc.). As a result, wide range of features of large variety of images are learned by the model. The AlexNet architecture accepts a size of 227 x 227 x 3 for an input image, where ‘3’ represents three color channels namely, Red, Green, and Blue (RGB)

AlexNet provides a balance between accuracy and speed. Also, it is a popular deep CNN among researchers for various classification problems. Fig. 3.10 outlines the architecture of AlexNet. AlexNet is a deep architecture comprising 5 convolutional layers combined with 3 fully connected layers and max-pooling layers. The AlexNet accepts an RGB image (here scalograms are fed as input) having 227x227 pixels. In AlexNet, all layers except output layers make use of the activation function known as Rectified Linear Unit (ReLu). ReLu is responsible for introducing non-linearity in the AlexNet.

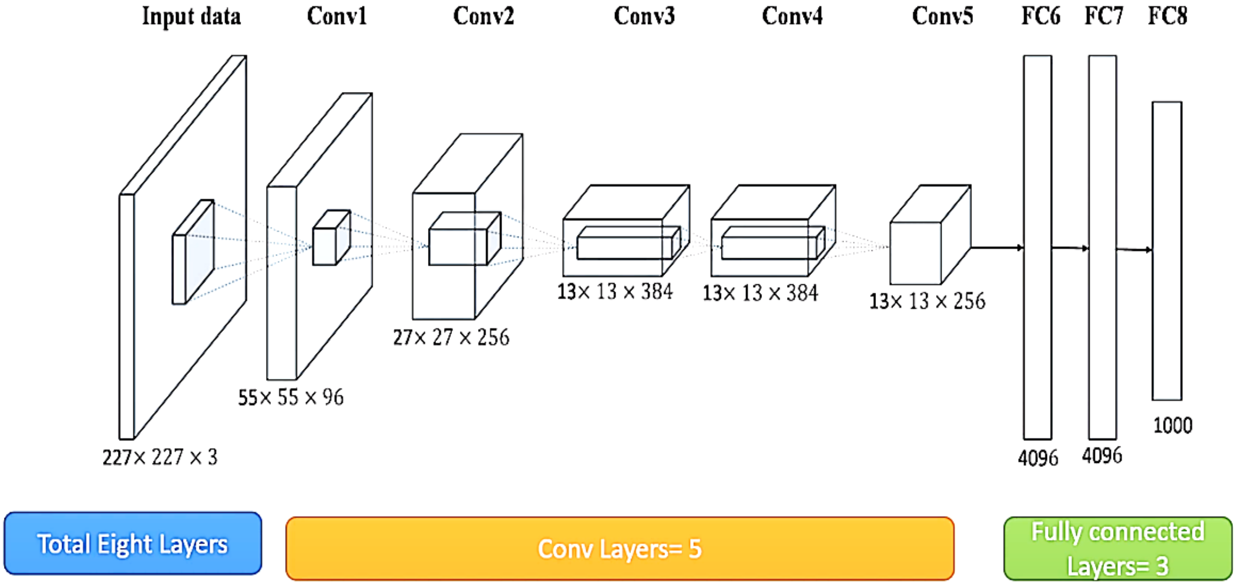


Fig. 3.10 AlexNet Architecture

Last 3 layers of AlexNet were substituted and fine-tuned in the experiment, to obtain modified AlexNet, for multi-class ECG classification. The replaced and fine-tuned layers to modify AlexNet include: fully connected, softmax, and classification layers. Fig. 3.11 (a) and (b) shows the layers of modified AlexNet and detailed information about layers and structure of modified AlexNet respectively.

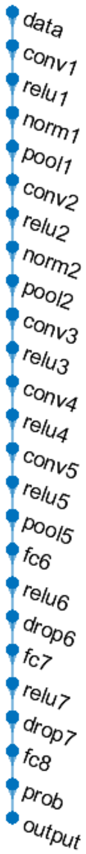
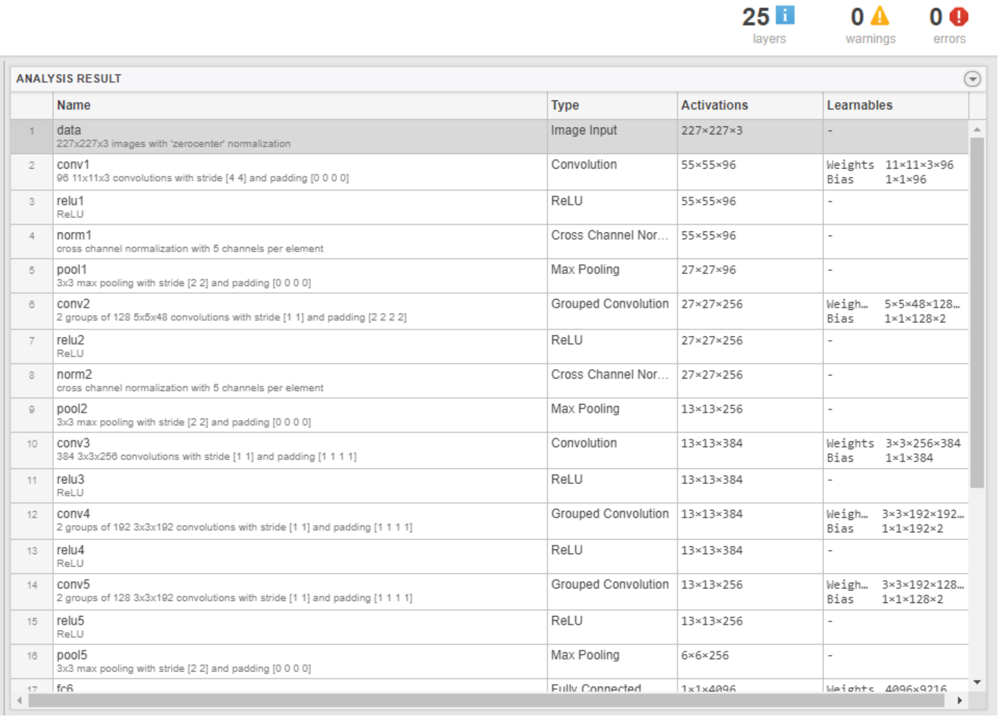


Fig. 3.11 (a) Layers of Modified AlexNet



25 layers   0 warnings   0 errors

ANALYSIS RESULT

	Name	Type	Activations	Learnables
1	data 227x227x3 images with 'zerocenter' normalization	Image Input	227×227×3	-
2	conv1 96 11x11x3 convolutions with stride [4 4] and padding [0 0 0 0]	Convolution	55×55×96	Weights 11×11×3×96 Bias 1×1×96
3	relu1 ReLU	ReLU	55×55×96	-
4	norm1 cross channel normalization with 5 channels per element	Cross Channel Nor…	55×55×96	-
5	pool1 3x3 max pooling with stride [2 2] and padding [0 0 0 0]	Max Pooling	27×27×96	-
6	conv2 2 groups of 128 5x5x48 convolutions with stride [1 1] and padding [2 2 2 2]	Grouped Convolution	27×27×256	Weigh… 5×5×48×128… Bias 1×1×128×2
7	relu2 ReLU	ReLU	27×27×256	-
8	norm2 cross channel normalization with 5 channels per element	Cross Channel Nor…	27×27×256	-
9	pool2 3x3 max pooling with stride [2 2] and padding [0 0 0 0]	Max Pooling	13×13×256	-
10	conv3 384 3x3x256 convolutions with stride [1 1] and padding [1 1 1 1]	Convolution	13×13×384	Weights 3×3×256×384 Bias 1×1×384
11	relu3 ReLU	ReLU	13×13×384	-
12	conv4 2 groups of 192 3x3x192 convolutions with stride [1 1] and padding [1 1 1 1]	Grouped Convolution	13×13×384	Weigh… 3×3×192×192… Bias 1×1×192×2
13	relu4 ReLU	ReLU	13×13×384	-
14	conv5 2 groups of 128 3x3x192 convolutions with stride [1 1] and padding [1 1 1 1]	Grouped Convolution	13×13×256	Weigh… 3×3×192×128… Bias 1×1×128×2
15	relu5 ReLU	ReLU	13×13×256	-
16	pool5 3x3 max pooling with stride [2 2] and padding [0 0 0 0]	Max Pooling	6×6×256	-
17	fc6	Fully Connected	1×1×4096	Weights 4096×9216

Fig. 3.11 (b) Network analysis of Modified AlexNet

3.5.2 GOOGLENET

GoogLeNet [32] was designed by a team of researchers at Google and has performance capability very close to humans. GoogleNet, also known as Inception, won the ILSVRC-2014 competition. Fig. 3.12 (a) outlines the architecture of GoogLeNet. GoogLeNet architecture was inspired by LeNet-5 [33]. The GoogleNet architecture accepts an RGB image of size 224 x 224 pixels. GoogLeNet makes use of an Inception Module, shown in Fig. 3.12 (b), the main structural component providing the benefit of convolution filters of different sizes. The inception module comprises convolution and max-pooling layers, arranged in parallel, which work together to combine their respective feature maps.

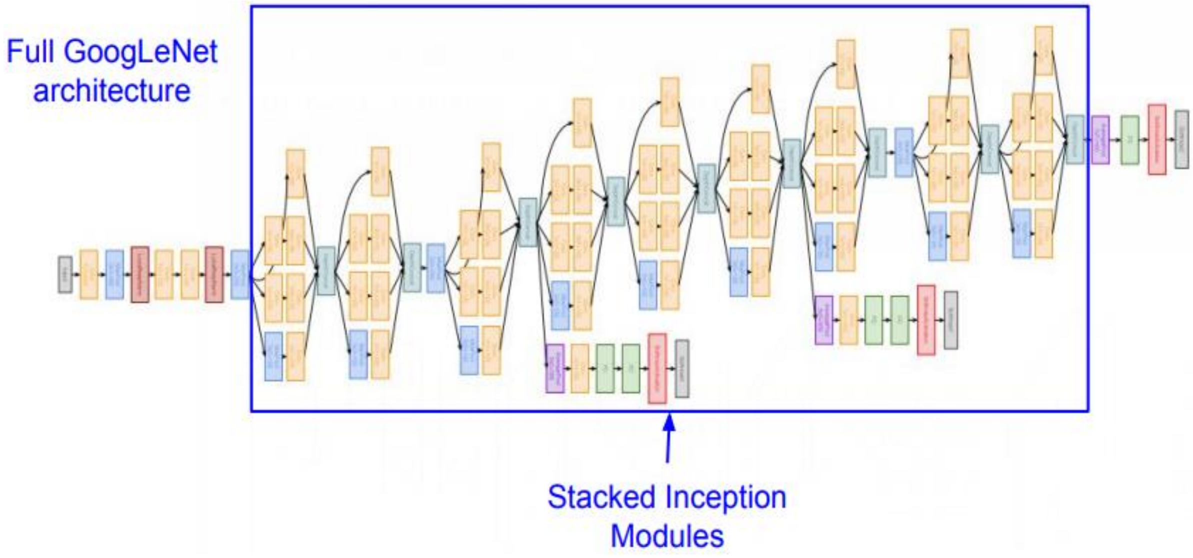


Fig. 3.12 (a) Architecture of GoogLeNet

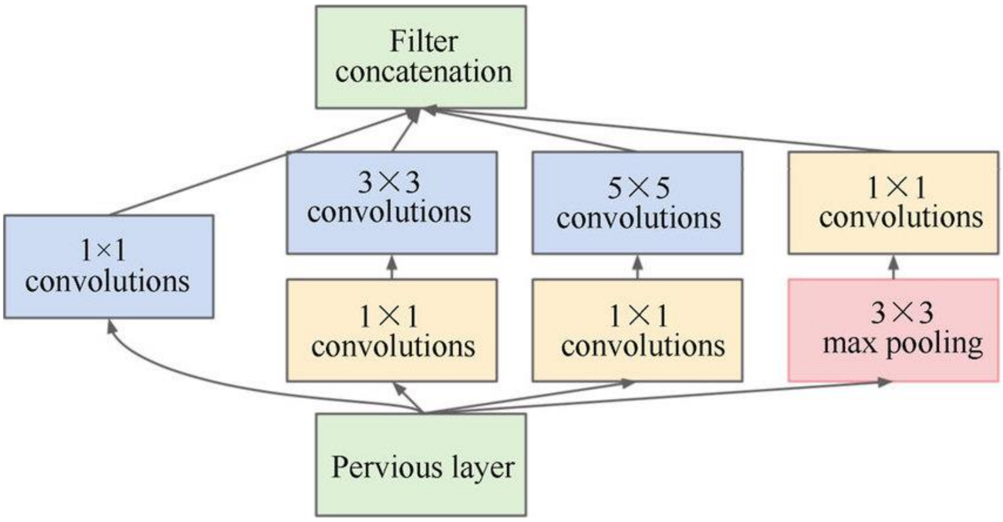


Fig. 3.12 (b) Inception Module present in GoogleNet [32]

By stacking 9 such inception modules linearly along with other layers and functions, 22 layers deep GoogLeNet architecture was obtained. Such architecture is effective in reducing the number of parameters i.e. from 60 million parameters in AlexNet to just 4 million parameters. Similar to AlexNet, the output layer was customized to deal with three ECG classes considered in this project. Specifically, the last learnable layer (i.e. fully connected layer) and final classification layer were replaced and the model was retrained for classification of three ECG classes.

3.5.3 SQUEEZENET

Researchers at the University of Berkeley, California, and Stanford, and DeepScale brought SqueezeNet [34] into existence in 2016. SqueezeNet is an 18 layer deep small CNN architecture, which is capable of providing an accuracy same as that of AlexNet, on the ImageNet database. It performs 3 times faster as compared to AlexNet due to 50 times fewer parameters. Further, using the deep compression technique, the model can be compressed to have a small size of just 0.47 MB from 4.8MB. Such small DNN models can be easily implemented on Field Programmable Gate Arrays (FPGAs) i.e. hardware with limited resources. Just like AlexNet, SqueezeNet architecture also accepts input RGB image size of 227 x 227. Fig. 3.13 (a) presents the architecture of SqueezeNet.

Fig. 3.13 (b) shows the Fire module, which is the fundamental component of SqueezeNet architecture. The fire module essentially contains a convolution layer (squeeze) having 1×1 filters, which provide input to an expanding layer having 1x1 and 3x3 filters. As we move from the beginning to the end of the network, the number of filters present per Fire Module gradually increases. All fire modules in SqueezeNet architecture make use of the ReLu activation function.

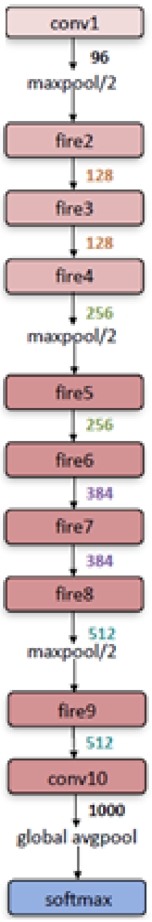


Fig. 3.13 (a) SqueezeNet Architecture

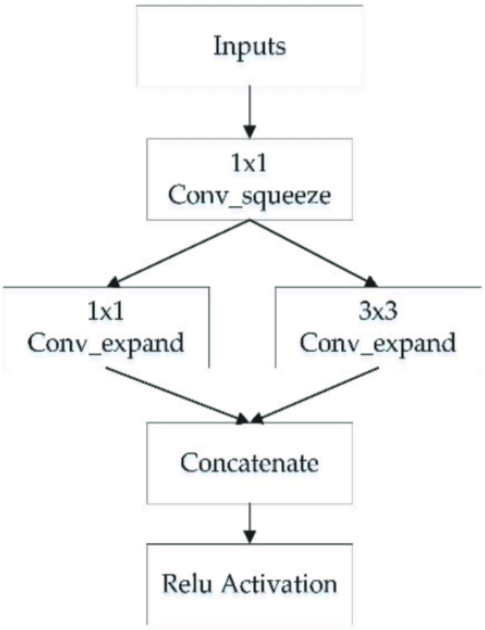


Fig. 3.13 (b) Fire Module used in SqueezeNet

Just as in the case of AlexNet and GoogLeNet, the output layer was replaced and fine-tuned to obtain modified SqueezeNet, which is capable of classifying three ECG classes i.e. NSR, CHF, and ARR. Specifically, the last convolutional and classification layers were replaced and modified in this project.

3.6 PROPOSED METHODOLOGY

Fig. 3.14 illustrates the methodology adopted in this work, for ECG signal classification. The proposed methodology makes use of CWT to obtain CWT coefficients, which are then plotted to obtain Scalograms of ECG signal samples. Scalograms are then re-sized and reconfigured, to feed them as input to modified CNN architectures i.e. AlexNet, GoogLeNet, and SqueezeNet, for ECG signal classification into three different classes namely: ARR, CHF, and NSR.

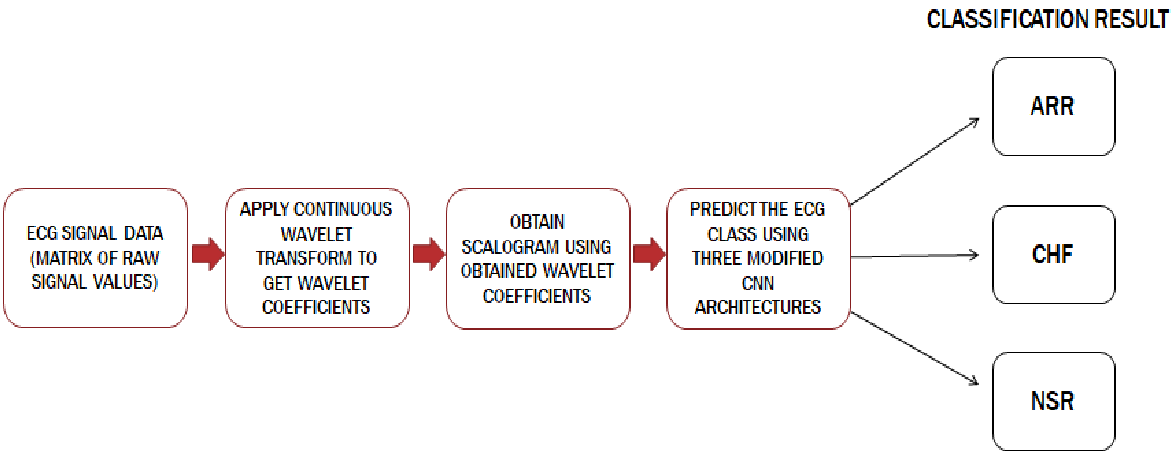

CLASSIFICATION RESULT
ECG SIGNAL DATA (MATRIX OF RAW SIGNAL VALUES)
APPLY CONTINUOUS WAVELET TRANSFORM TO GET WAVELET COEFFICIENTS
OBTAIN SCALOGRAM USING OBTAINED WAVELET COEFFICIENTS
PREDICT THE ECG CLASS USING THREE MODIFIED CNN ARCHITECTURES
ARR
CHF
NSR

Fig. 3.14 Architecture of proposed method in this work for Multi-class ECG classification

CHAPTER 4

RESULTS

4.1 SYSTEM SETUP

The following setup and resources were used for conducting the experiments:

I. HARDWARE

1. Intel Core i3-5005U Single Processor
2. 8 GB DDR-3 RAM
3. 240 GB SSD Storage

II. SOFTWARE & SIMULATION TOOLBOXES

1. MATLAB (2020a)
2. Wavelet Toolbox
3. Deep Learning Toolbox

4.2 PRE-PROCESSED SCALOGRAM DATASET

As described in Chapter 3, 1D ECG signals are converted to 2D scalograms, to extract useful information from the 1D ECG signals, as well as, to make them suitable for feeding them as input to Convolutional Neural Network (CNN) based architectures. Continuous Wavelet Transform (CWT) with the following settings and parameters was used to obtain corresponding scalograms:

1. Analytic Morlet (Gabor) wavelet (also called 'amor') is used for analysis.
2. Gabor wavelet has equal time and frequency variance, as shown in Fig. 4.1.
3. Analytic Morlet wavelet with 12 wavelet bandpass filters per octave is utilized in this work.
4. A color map of type "*jet*" having "*128*" different colors were used for representing each scalogram

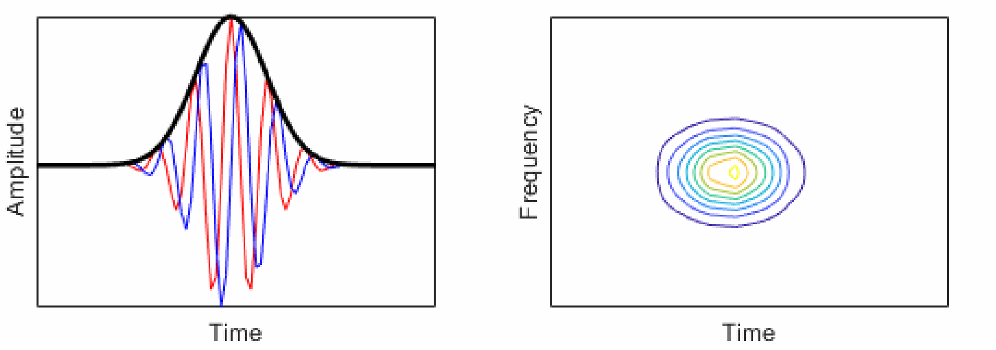


Fig. 4.1 Time domain and Time-Frequency domain representation of Analytic Morlet Wavelet

Wavelets characterized by the complex-valued nature in the time domain along with single-sided spectrum, are referred to as Analytic wavelets. For time-frequency analysis using CWT, such wavelets are found to be suitable.

In this work, a custom function “ecg2scg” using MATLAB was developed which perform the following tasks:

1. Takes 500 samples per signal of 1D ECG dataset as input.
2. Creates time-frequency representation from the ECG dataset using CWT.
3. Obtain scalogram representation from the wavelet coefficients obtained above.
4. Convert obtained scalogram into an RGB image of size 224 x 224 pixels or 227 x 227 pixels depending on the input image size requirements of the model, and
5. Save the obtained scalograms into three different directories corresponding to each type of signal i.e., ARR, CHF, and NSR, as shown below in Fig. 4.2.

Fig. 4.2 Three directories containing scalograms of type ARR, CHF, and NSR

Fig. 4.3 to 4.5 shows the scalograms obtained of type ARR, CHF, and NSR in their respective directories.

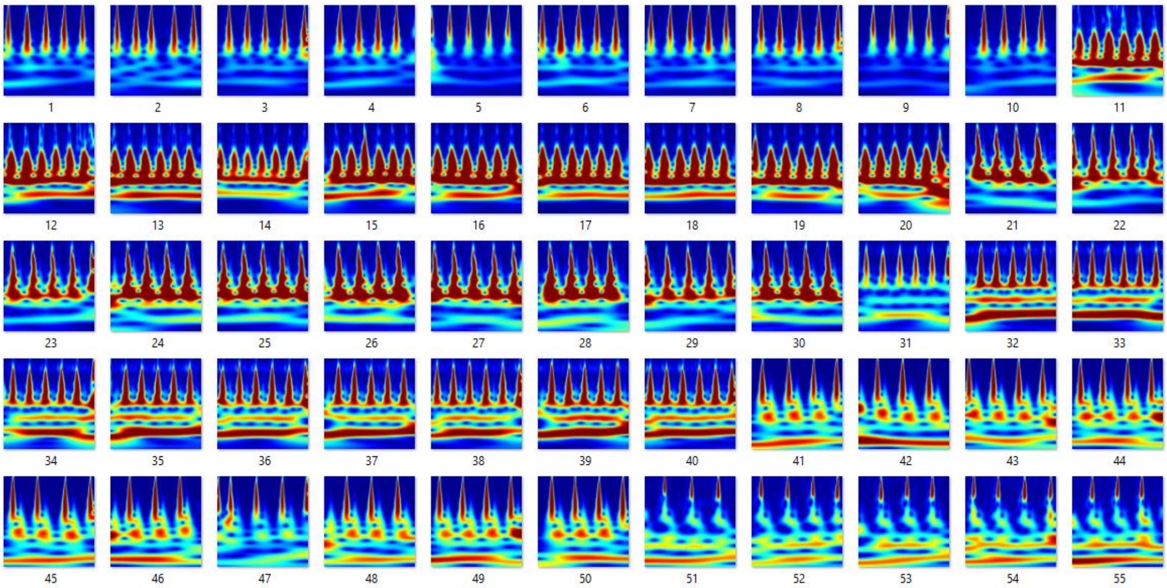


Fig. 4.3 Scalograms of type ARR in respective directory

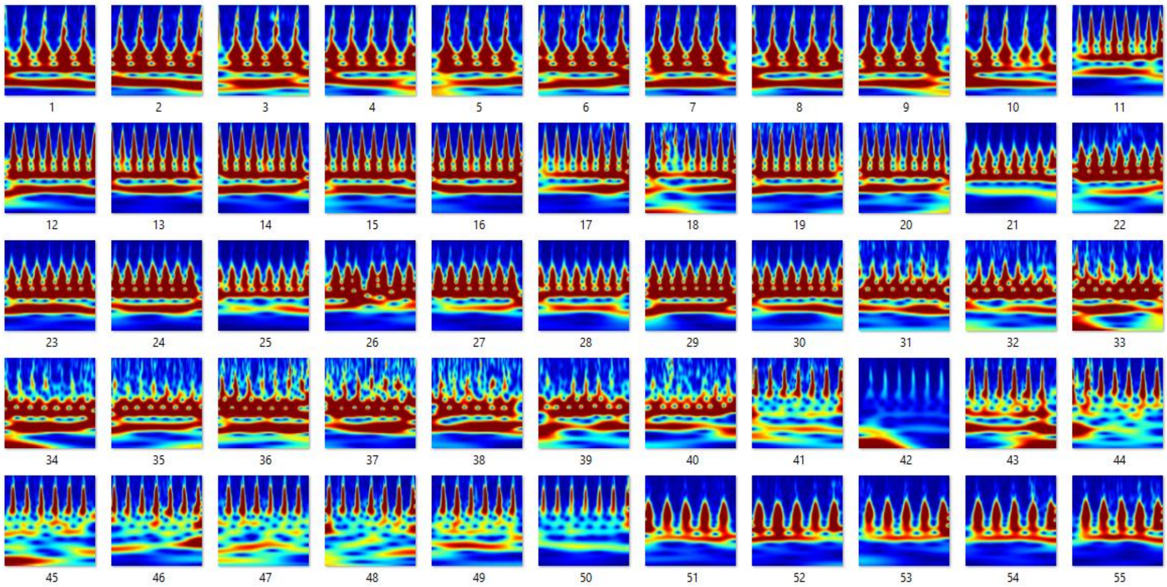


Fig. 4.4 Scalograms of type CHF in respective directory

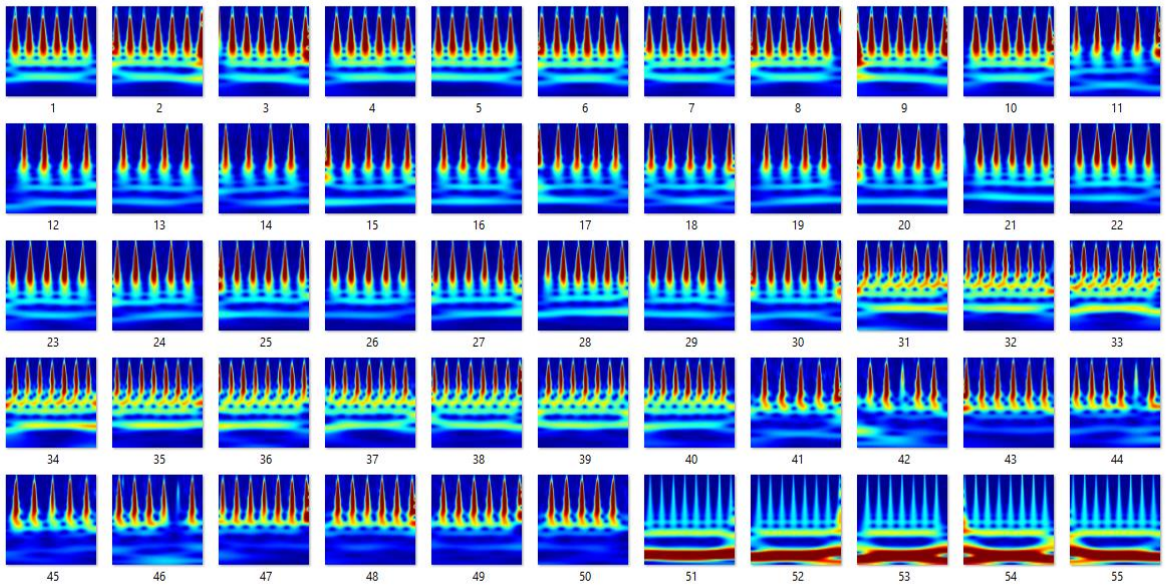


Fig. 4.5 Scalograms of type NSR in respective directory

Finally, a total of 900 scalograms were obtained, having 300 of each type and stored in their respective directory.

4.3 MODIFIED ARCHITECTURES

4.3.1 MODIFIED ALEXNET

As discussed in chapter 3, this work uses transfer learning approach. To perform the multi-class ECG classification task, the last three layers were replaced and fine-tuned. The configuration of the last three layers was modified as below:

a) For Fully Connected Layer:
   - The number of classes was set to 3.
   - The values of the Weight Learn Rate Factor and the Bias Learn Rate Factor were chosen to be 20.

b) Soft-max Layer was used as default.

c) Classification Layer was used as default.

4.3.2 MODIFIED GOOGLENET

All the layers of GoogLeNet were taken as is, except the last three layers which were replaced and fine-tuned. Following modifications were done in the last three layers:

- A new dropout layer with a probability of 0.6 replaced the final dropout layer.
- A new fully-connected layer with the number of filters set to 3 replaced the original fully connected layer.
- A new classification layer without a class label was used instead of the original classification layer.

4.3.3 MODIFIED SQUEEZENET

All the layers of SqueezeNet were taken as is, except the last three layers which were replaced and fine-tuned. Following modifications were done in the last three layers:

- A new dropout layer with a probability of 0.6 replaced the final dropout layer.
- A new convolutional layer with the number of filters set to 3 replaced the original convolutional layer.
- A new classification layer without a class label was used instead of the original classification layer.

4.4 TRAINING AND TESTING

For multi-class ECG classification, the obtained scalogram images were fed as input to three modified CNN architectures. The Scalogram dataset was split into training and testing, as shown in Table 4.1. The whole scalogram dataset was arbitrarily split into 80% and 20% per class, for training and testing, respectively; i.e. total of 240 scalogram files were chosen randomly from each directory for training, resulting in a total of 720 training files. Correspondingly, 60 scalogram files of each type were used for testing.

TABLE 4.1 SUMMARY OF SCALOGRAM DATASET USED FOR TRAINING AND TESTING

TOTAL	TRAINING	TESTING
900	750	180

For training of modified CNN architectures, specific training parameters were chosen based on trial and error which yielded the best result for the ECG classification task at hand. Table 4.2 shows lists the various training parameters along with their chosen values.

TABLE 4.2 TRAINING PARAMETERS

TRAINING PARAMETERS	PARAMETER VALUE
Solver	Stochastic Gradient Descent with Momentum (sgdm)
Epochs	10
Iterations Per Epoch	48
Mini-Batch size	15
Shuffle	Every Epoch
Initial Learn Rate	0.0001

Using above mentioned dataset and training settings, three modified CNN architectures were trained and tested. Fig. 4.6 shows the plot of Accuracy and Loss/Error while the three networks are trained on ECG scalograms. The Validation accuracy and training time obtained are shown in Table 4.3.

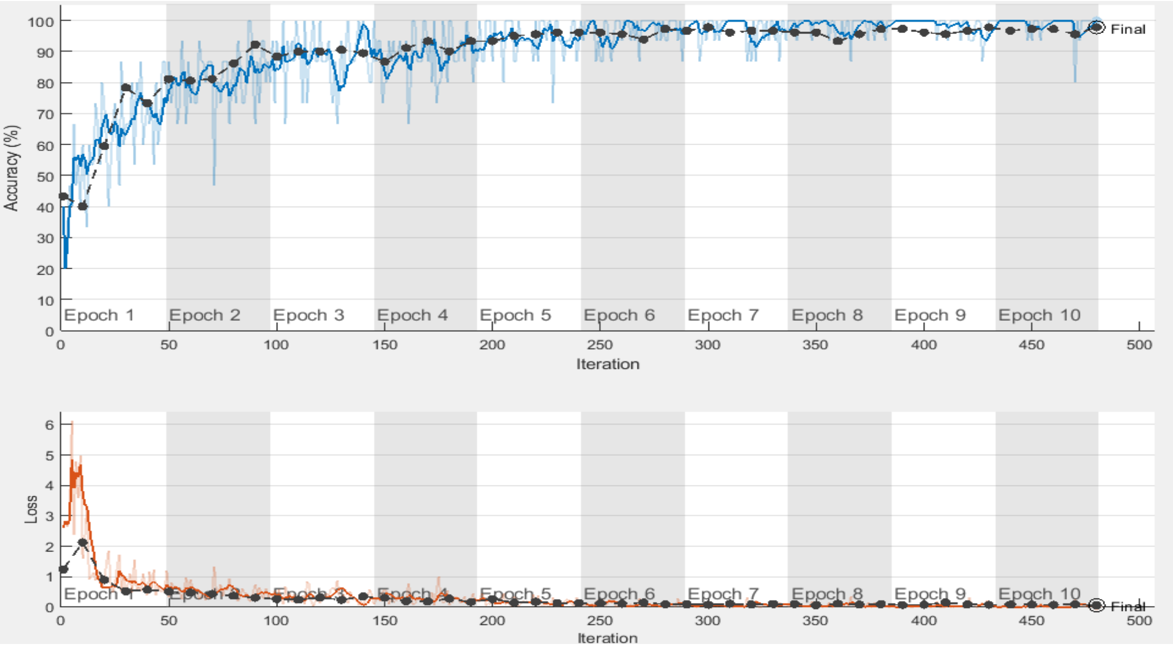


Fig. 4.6 (a) Accuracy and Loss graphs during training of Modified AlexNet

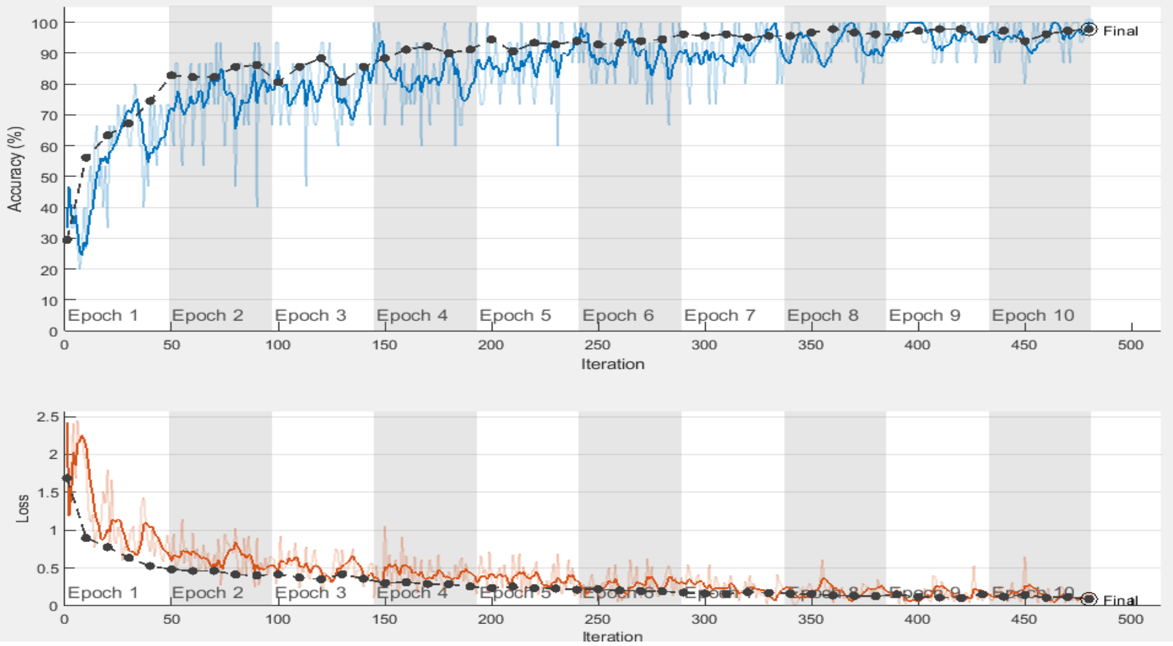


Fig. 4.6 (b) Accuracy and Loss graphs during training of Modified GoogLeNet

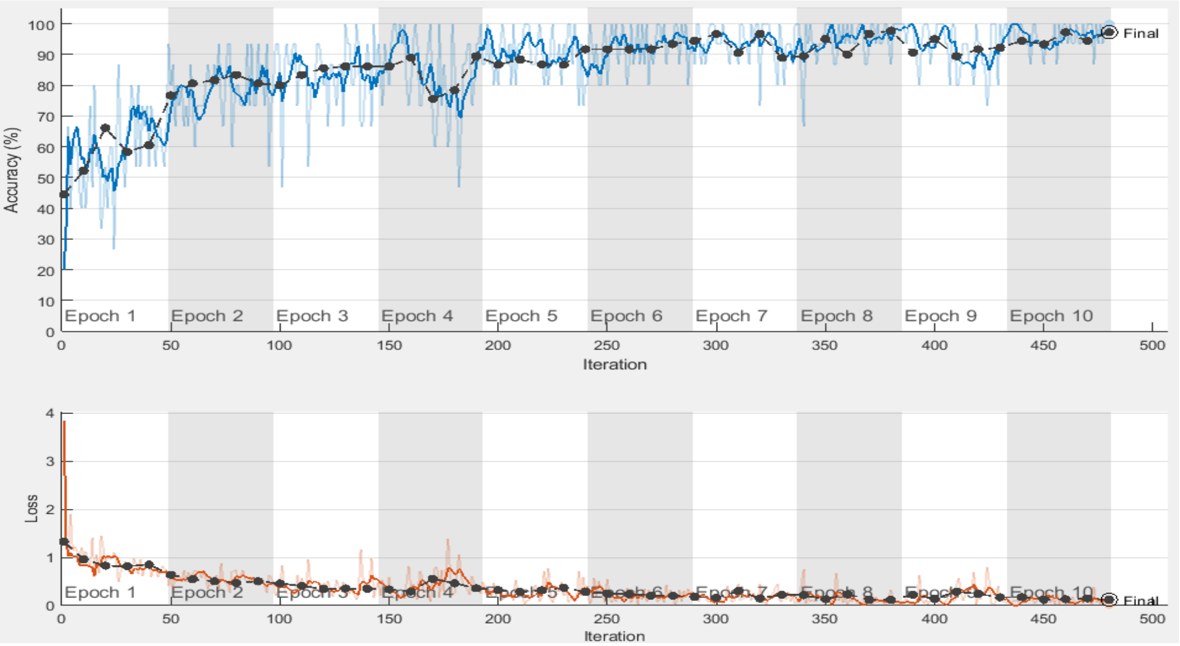


Fig. 4.6 (c) Accuracy and Loss graphs during training of Modified SqueezeNet

TABLE 4.3 ACCURACY AND TRAINING TIME FOR VARIOUS MODELS

MODEL	VALIDATION ACCURACY	TRAINING TIME
AlexNet	97.80	22 min. 57 sec.
GoogLeNet	97.78	69 min. 03 sec.
SqueezeNet	97.22	25 min. 27sec.

4.5 PERFORMANCE METRICS

For investigating and evaluating the classification performance of three modified CNN architectures, the following parameters were chosen and calculated:

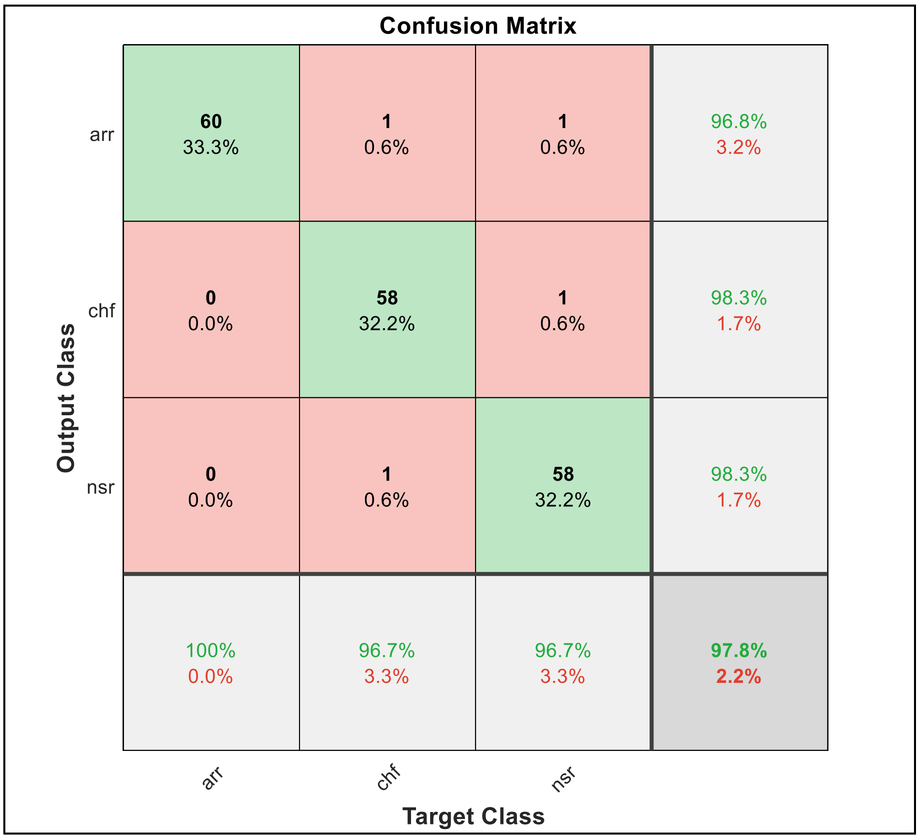
a) Precision
b) Recall, and
c) F-1 Score, and
d) Overall Accuracy

Fig. 4.7 shows the three-class confusion matrices obtained by three different models upon successful training and classification.

The confusion matrix essentially contains four parameters which are True Positive (TP), True-Negative (TN), False-Positive (FP), and False-Negative (FN).

TP and TN correspond to the number of accurately predicted positive and negative classes respectively and are positioned along two diagonals of the confusion matrix. Whereas, FP and FN correspond to the number of incorrectly predicted positive and negative classes respectively.

In confusion matrices shown in Fig 4.7, the row depicts the predicted class and the column depicts the true or actual class.

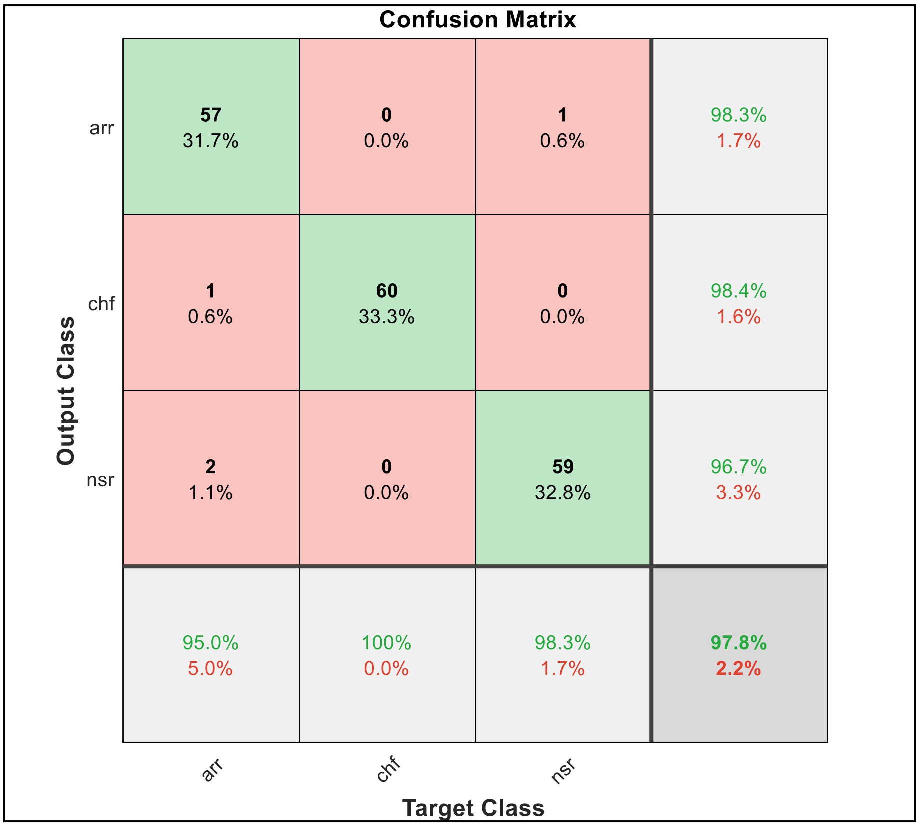


Confusion Matrix

Output Class	arr	chf	nsr	
arr	60 33.3%	1 0.6%	1 0.6%	96.8% 3.2%
chf	0 0.0%	58 32.2%	1 0.6%	98.3% 1.7%
nsr	0 0.0%	1 0.6%	58 32.2%	98.3% 1.7%
	100% 0.0%	96.7% 3.3%	96.7% 3.3%	**97.8%** **2.2%**

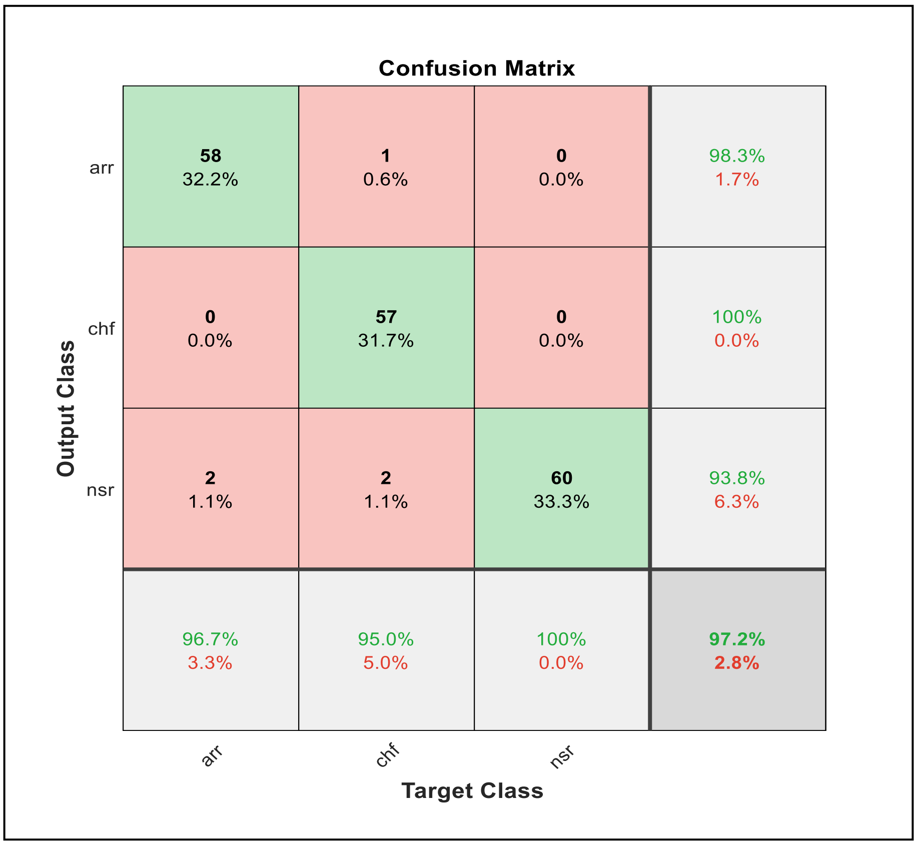
Target Class

Fig. 4.7 (a) Confusion Matrix of Modified AlexNet



	arr	chf	nsr	
arr	57 31.7%	0 0.0%	1 0.6%	98.3% 1.7%
chf	1 0.6%	60 33.3%	0 0.0%	98.4% 1.6%
nsr	2 1.1%	0 0.0%	59 32.8%	96.7% 3.3%
	95.0% 5.0%	100% 0.0%	98.3% 1.7%	97.8% 2.2%

Fig. 4.7 (b) Confusion Matrix of GoogLeNet



	arr	chf	nsr	
arr	58 32.2%	1 0.6%	0 0.0%	98.3% 1.7%
chf	0 0.0%	57 31.7%	0 0.0%	100% 0.0%
nsr	2 1.1%	2 1.1%	60 33.3%	93.8% 6.3%
	96.7% 3.3%	95.0% 5.0%	100% 0.0%	97.2% 2.8%

Fig. 4.7 (c) Confusion Matrix of SqueezeNet

The rightmost column in the confusion matrix has percentages of all samples predicted to belong to each of the three classes which are correctly and incorrectly classified by the model. These metrics are called Precision and False Positive Rate respectively.

The row present at the bottom of the confusion matrix has percentages of all samples belonging to each of the three classes which are classified correctly and incorrectly by the model. These metrics are called Recall and False Negative Rate respectively. The cell present at the bottom right of the confusion matrix represents the overall accuracy achieved by the modified AlexNet network.

Precision and Recall are combined to obtain a new performance metric known as F1-score. Mathematically, the F1 score is the harmonic mean of Precision and Recall. It is representative of the effectiveness of classification when equal importance is given to recall and precision. The expressions for various performance metrics that can be calculated for each class are mentioned below.

$$\text{Precision} = \frac{\text{TRUE POSITIVE}}{\text{TRUE POSITIVE} + \text{FALSE POSITIVE}} \tag{4.1}$$

$$\text{Recall} = \frac{\text{TRUE POSITIVE}}{\text{TRUE POSITIVE} + \text{FALSE NEGATIVE}} \tag{4.2}$$

$$\text{F1-score} = 2 \times \frac{\text{PRECISION} \times \text{RECALL}}{\text{PRECISION} + \text{RECALL}} \tag{4.3}$$

The values of the above-mentioned performance metrics lie in the range of 0 and 1, where a value of 1 signifies perfect classification without any error. Table 4.4 shows the values of various performance metrics obtained after macro-averaging the class-wise results. The obtained values indicate promising results even when such a small dataset of 900 images is considered. The modified architectures using the transfer learning approach can successfully classify the ECG images into three categories with almost human-level accuracy and minimal error.

TABLE 4.4 PERFORMANCE METRICS OBTAINED AFTER MACRO-AVERAGING

MODELS PARAMETERS	ALEXNET	GOOGLENET	SQUEEZENET
Precision	0.977	0.978	0.973
Recall	0.978	0.977	0.972
F1- Score	0.977	0.977	0.973

CHAPTER 5

CONCLUSION & FUTURE SCOPE

The above sections of this chapter present the processed scalogram dataset, observations, accuracies, and performance metrics obtained on the classification of the ECG scalogram dataset using three different convolutional neural network architectures. Deep Neural Networks i.e. AlexNet, GoogLeNet, and SqueezeNet were studied in this work for ECG scalogram image classification with the dataset having very little significant information varying among three different classes.

The samples from the raw ECG dataset were successfully converted to corresponding scalograms which were competent in preserving the little intricacies of the ECG signal. ECG scalogram image classification using modified pre-trained CNN models was done and the overall Classification accuracies in the range of 97.22% to 97.78% were obtained. CNN models were fine-tuned and utilized the transfer learning approach to achieve such accuracy. Such an approach also made it possible to perform such classification task on limited hardware capabilities. Also, because of transfer learning, it was possible to obtain such accuracy in the classification task, even with such a small dataset of ECG signals. Modified CNN models took different training times based upon their internal architectures and structures.

A novel methodology involving ECG signal samples being converted to images using Continuous Wavelet Transform coefficients and classified using pre-trained AlexNet, GoogLeNet, and SqueezeNet models is presented in this work. For future work, we aim to collect more and more data for training and testing. Advanced CNN models and their structure will be studied along with a comparative analysis of their performance.

REFERENCES

[1] http://www.alilamedicalimages.org/2016/09/26/understanding-12-lead-ecg-system-animation

[2] https://physionet.org/about/

[3] W. Zhang, L H. Wang et al., "A Low-Power High-Data-Transmission multi-lead ECG Acquisition Sensor System", IEEE Sensors J., vol. 19, no. 22, pp. 1-3, Nov. 2019.

[4] F. K. Shaikh, S. Zeadally et al., "Enabling Technologies for Green Internet of Things", IEEE Syst. J., vol. 11, no. 2, pp. 983-994, Jun. 2017.

[5] L. H. Wang, W. Z. Dong et al., "Low-Power Low-Data-Loss Bio-Signal Acquisition System for Intelligent Electrocardiogram Detection", IEICE Electron. Express, vol. 14, pp. 1-9, 2017.

[6] A. Diker, E. Avci, Z. Cömert, D. Avci, E. Kaçar and İ. Serhatlioğlu, "Classification of ECG signal by using machine learning methods", 2018 26th Signal Processing and Communications Applications Conference (SIU), pp. 1-4, 2018.

[7] A. Diker, Z. Cömert, E. Avci, and S. Velappan, "Intelligent system based on Genetic Algorithm and support vector machine for detection of myocardial infarction from ECG signals", 2018 26th Signal Processing and Communications Applications Conference (SIU), pp. 1-5, 2018.

[8] A. Diker, Z. Cömert and E. Avcı, "A Diagnostic Model for Identification of Myocardial Infarction from Electrocardiography Signals", Bitlis Eren Univ. J. Sci. Technol., vol. 7, no. 2, pp. 132-140, 2017

[9] C. H. Antink, S. Leonhardt and M. Walter, "Fusing QRS detection and robust interval estimation with a random forest to classify atrial fibrillation", 2017 Computing in Cardiology (CinC), pp. 1-4, 2017.

[10] R. Colloca, A. E. Johnson, L. Mainardi, and G. D. Clifford, A Support Vector Machine Approach for Reliable Detection of Atrial Fibrillation Events Computing in Cardiology Conference. IEEE, 2013.

[11] G. Bin, M. Shao, G. Bin, J. Huang, D. Zheng, and S. Wu, "Detection of Atrial Fibrillation Using Decision Tree Ensemble", 2017 Computing in Cardiology Conference, 2017.

[12] Sahoo, S., Dash, M., Behera, S. and Sabut, S., 2020. Machine Learning Approach to Detect Cardiac Arrhythmias in ECG Signals: A Survey. IRBM, Volume 41, Issue 4, 2020, Pages 185-194, ISSN 1959-0318, https://doi.org/10.1016/j.irbm.2019.12.001.

[13] Z. Xiong, M. P. Nash, E. Cheng, V. V. Fedorov, M. K. Stiles, and J. Zhao, "ECG signal classification for the detection of cardiac arrhythmias using a convolutional recurrent neural network", Physiological measurement, vol. 39, no. 9, pp. 094006, 2018.

[14] S. Ghiasi, M. Abdollahpur, Nasimalsadat Madani, Kamran Kiyani, and Ali Ghaffari, "Atrial Fibrillation Detection Using Feature-Based Algorithm and Deep Conventional Neural Network", 2017 Computing in Cardiology Conference, 2017.

[15] L. Deng and D. Yu, “Deep Learning: Methods and Applications,” Found. Trends® Signal Process., vol. 7, no. 3–4, pp. 197–387, 2014. Douangnoulack, Phonethep, and Veera Boonjing. "Building Minimal Classification Rules for Breast Cancer Diagnosis." 2018 10th International Conference on Knowledge and Smart Technology (KST). IEEE, 2018.

[16] Özal Yildirim, A novel wavelet sequence based on deep bidirectional LSTM network model for ECG signal classification, Computers in Biology and Medicine, Volume 96, 2018, Pages 189-202, ISSN 0010- 4825.

[17] Rajpurkar, Pranav & Hannun, Awni & Haghpanahi, Masoumeh & Bourn, Codie & Y. Ng, Andrew. (2017). Cardiologist-Level Arrhythmia Detection with Convolutional Neural Networks.

[18] Goldberger AL, Amaral LAN, Glass L, Hausdorff JM, Ivanov PCh, Mark RG, Mietus JE, Moody GB, Peng C-K, Stanley HE. PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals. Circulation 101(23): e215-e220 [Circulation Electronic Pages; http://circ.ahajournals.org/content/101/23/e215.full]; 2000 (June 13).

[19] Moody GB, Mark RG. The impact of the MIT-BIH Arrhythmia Database. IEEE Eng in Med and Biol 20(3):45-50 (May-June 2001).

[20] Baim DS, Colucci WS, Monrad ES, Smith HS, Wright RF, Lanoue A, Gauthier DF, Ransil BJ, Grossman W, Braunwald E. Survival of patients with severe congestive heart failure treated with oral milrinone. J American College of Cardiology 1986 Mar; 7(3):661-670.

[21] S. Mallat, A wavelet tour of signal processing, Elsevier, 1999.

[22] Gurney, K., 1997. Neural Networks – An Overview, An Introduction to Neural Networks. London: UCL Press. ISBN 0-203-45151-1.

[23] Mebsout, I., 2020. Convolutional Neural Networks’ Mathematics. [online] Medium. Avail- able at: https://medium.com/swlh/convolutional-neural-networks-mathematics-1beb3e6447c0.

[24] Skalski, P., 2019. Gentle Dive into Math Behind Convolutional Neural Networks. [online] Medium. Available at: https://towardsdatascience.com/gentle-dive-into-math-behind- convolutional-neural-networks-79a07dd44cf9.

[25] Ujjwalkarn, U., 2016. An Intuitive Explanation Of Convolutional Neural Networks. [online] The data science blog. Available at: https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/.

[26] Brownlee, J., 2016. Overfitting and Underfitting With Machine Learning Algorithms. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/.

[27] Srivastava, N., Sutskever, I. and Salakhutdinov, R., 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research 15.

[28] Brownlee, J., 2018. How to Develop 1D Convolutional Neural Network Models for Human Activity Recognition. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/cnn-models-for-human-activity-recognition-time-series-classification/.

[29] Guo Z, Chen Q, Wu G, Xu Y, Shibasaki R, Shao X., Village Building Identification Based on Ensemble Convolutional Neural Networks. Sensors.;17(11):2487, 2017.

[30] ImageNet. http://www.image-net.org.

[31] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet classification with deep convolutional neural networks", Neural Information Processing Systems, vol. 25, no. 2, 2012.

[32] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, “Going deeper with convolutions,” in IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.

[33] Huang, Gao, et al. "Densely connected convolutional networks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2017.

[34] Iandola, F.N.; Han, S.; Moskewicz, M.W.; Ashraf, K.; Dally, W.J.; Keutzer, K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. arXiv 2016; arXiv: 1602.07360

APPENDIX-I

MATLAB CODE

- **To obtain Scalograms using Continuous Wavelet Transform**

```
load('ECGData.mat');
data= ECGData.Data;
labels = ECGData.Labels;

ARR=data(1:30,:);
CHF=data(97:126,:);
NSR=data(127:156,:);
signallength = 500;

fb = cwtfilterbank('SignalLength', signallength,
'Wavelet','amor','VoicesPerOctave',12);

mkdir('ECG dataset');
mkdir('ECG dataset\arr');
mkdir('ECG dataset\chf');
mkdir('ECG dataset\nsr');

ecgtype={'ARR', 'CHF', 'NSR'};

ecg2cwtscg(ARR,fb,ecgtype{1});
ecg2cwtscg(CHF,fb,ecgtype{2});
ecg2cwtscg(NSR,fb,ecgtype{3});
```

- **Custom function for creating scalogram database**

```
function ecg2cwtscg(ECGdata,cwtfb,ecgtype)
nos=10;
sl=500;
colormap=jet(128);
if ecgtype=='ARR'
    folderpath=strcat('C:\ECG_DEEP LEARNING\ECG dataset\ARR\');
    findx=0;
    for i=1:30
        indx=0;
        for k=1:nos
            ecgsignal=ECGdata(i,indx+1:indx+sl);
            cfs = abs(cwtfb.wt(ecgsignal));
            im = ind2rgb(im2uint8(rescale(cfs)),colormap);
            filenameindex=findx+k;
            filename=strcat(folderpath,sprintf('%d.jpg',filenameindex));
            imwrite(imresize(im,[227 227]),filename);
            indx=indx+sl;
        end
        findx=findx+nos;
    end
```

```
elseif ecgtype=='CHF'
    folderpath=strcat('C:\ECG_DEEP LEARNING\ECG dataset\CHF\');
    findx=0;
    for i=1:30
        indx=0;
        for k=1:nos
            ecgsignal=ECGdata(i,indx+1:indx+sl);
            cfs = abs(cwtfb.wt(ecgsignal));
            im = ind2rgb(im2uint8(rescale(cfs)),colormap);
            filenameindex=findx+k;
            filename=strcat(folderpath,sprintf('%d.jpg',filenameindex));
            imwrite(imresize(im,[227 227]),filename);
            indx=indx+sl;
        end
        findx=findx+nos;
    end
elseif ecgtype=='NSR'
    folderpath=strcat('C:\ECG_DEEP LEARNING\ECG dataset\NSR\');
    findx=0;
    for i=1:30
        indx=0;
        for k=1:nos
            ecgsignal=ECGdata(i,indx+1:indx+sl);
            cfs = abs(cwtfb.wt(ecgsignal));
            im = ind2rgb(im2uint8(rescale(cfs)),colormap);
            filenameindex=findx+k;
            filename=strcat(folderpath,sprintf('%d.jpg',filenameindex));
            imwrite(imresize(im,[227 227]),filename);
            indx=indx+sl;
        end
        findx=findx+nos;
    end
end
```

• Modified AlexNet for training and classification

```
DatasetPath= 'C:\ECG_DEEP LEARNING\ECG dataset';
images =
imageDatastore(DatasetPath,'IncludeSubfolders',true,'LabelSource','folde
rnames');

numTrainFiles = 240;
[TrainImages,TestImages] =
splitEachLabel(images,numTrainFiles,'randomize');

net = alexnet;
layersTransfer = net.Layers(1:end-3);
numClasses =3;
layers = [
    layersTransfer

fullyConnectedLayer(numClasses,'WeightLearnRateFactor',20,'BiasLearnRate
Factor',20)
    softmaxLayer
```

```
        classificationLayer];


    options = trainingOptions('sgdm', ...
        'MiniBatchSize',15, ...
        'MaxEpochs',20, ...
        'InitialLearnRate',1e-4, ...
        'Shuffle','every-epoch', ...
        'ValidationData',TestImages, ...
        'ValidationFrequency',10, ...
        'Verbose',false, ...
        'Plots','training-progress');

    netTransfer = trainNetwork(TrainImages,layers,options);

    YPred = classify(netTransfer,TestImages);
    YValidation = TestImages.Labels;
    accuracy=sum(YPred == YValidation)/numel(YValidation)
```

- **Modified GoogLeNet for training and classification**

```
DatasetPath= 'C:\ECG_DEEP LEARNING\ECG dataset';
 images =
imageDatastore(DatasetPath,'IncludeSubfolders',true,'LabelSource','folderna
mes');

numTrainFiles = 240;
[TrainImages,TestImages] =
splitEachLabel(images,numTrainFiles,'randomize');

net = googlenet;
lgraph = layerGraph(net);
newDropoutLayer = dropoutLayer(0.6,'Name','new_Dropout');
lgraph = replaceLayer(lgraph,'pool5-drop_7x7_s1',newDropoutLayer);

numClasses = 3;
newConnectedLayer = fullyConnectedLayer(numClasses,'Name','new_fc',...
    'WeightLearnRateFactor',5,'BiasLearnRateFactor',5);
lgraph = replaceLayer(lgraph,'loss3-classifier',newConnectedLayer);

newClassLayer = classificationLayer('Name','new_classoutput');
lgraph = replaceLayer(lgraph,'output',newClassLayer);


options = trainingOptions('sgdm',...
    'MiniBatchSize',15,...
    'MaxEpochs',10,...
    'InitialLearnRate',1e-4,...
    'Shuffle','every-epoch', ...
    'ValidationData',TestImages,...
    'ValidationFrequency',10,...
    'Verbose',0,...
    'Plots','training-progress');

trainGN = trainNetwork(TrainImages,lgraph,options);

YPred = classify(trainGN,TestImages);
```

```
YValidation = TestImages.Labels;
      accuracy=sum(YPred == YValidation)/numel(YValidation)
```

- ### Modified SqueezeNet for training and classification

```
sqz = squeezenet;
DatasetPath= 'C:\ECG_DEEP LEARNING\ECG dataset';
images =
imageDatastore(DatasetPath,'IncludeSubfolders',true,'LabelSource','folderna
mes');

numTrainFiles = 240;
[TrainImages,TestImages] =
splitEachLabel(images,numTrainFiles,'randomize');

net = squeezenet;
lgraphSqz = layerGraph(sqz);

tmpLayer = lgraphSqz.Layers(end-5);
newDropoutLayer = dropoutLayer(0.6,'Name','new_dropout');
lgraphSqz = replaceLayer(lgraphSqz,tmpLayer.Name,newDropoutLayer);

numClasses = 3;
tmpLayer = lgraphSqz.Layers(end-4);
newLearnableLayer = convolution2dLayer(1,numClasses, ...
        'Name','new_conv', ...
        'WeightLearnRateFactor',10, ...
        'BiasLearnRateFactor',10);
lgraphSqz = replaceLayer(lgraphSqz,tmpLayer.Name,newLearnableLayer);
tmpLayer = lgraphSqz.Layers(end);
newClassLayer = classificationLayer('Name','new_classoutput');
lgraphSqz = replaceLayer(lgraphSqz,tmpLayer.Name,newClassLayer);

options = trainingOptions('sgdm',...
    'MiniBatchSize',15,...
    'MaxEpochs',10,...
    'InitialLearnRate',1e-4,...
    'Shuffle','every-epoch', ...
    'ValidationData',TestImages,...
    'ValidationFrequency',10,...
    'Verbose',0,...
    'Plots','training-progress');

trainSN = trainNetwork(TrainImages,lgraphSqz,options);

YPred = classify(trainSN,TestImages);
YValidation = TestImages.Labels;
accuracy=sum(YPred == YValidation)/numel(YValidation)
```

APPENDIX-II

LIST OF PUBLICATION

1. "A Deep Neural Network Approach to Automatic Multi-Class Classification of Electrocardiogram Signals". IEEE International Conference on Intelligent Technologies (CONIT), 2021.

2. “A Comparative Analysis of Deep Neural Network Models using Transfer Learning for Electrocardiogram Signal Classification”. 6th IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (IRTEICT), 2021.