



KADIR HAS UNIVERSITY
SCHOOL OF GRADUATE STUDIES
DEPARTMENT OF ENGINEERING AND NATURAL SCIENCES

**CLASSIFICATION OF HEART DISEASES WITH
CONVOLUTIONAL NEURAL NETWORKS**

BEKIR YAVUZ KOÇ
ASSOC.PROF.DR. TANER ARSAN

MASTER'S DEGREE THESIS


ISTANBUL, FEBRUARY, 2021

BEKİR YAVUZ KOÇ

MASTER'S DEGREE THESIS

2021

CLASSIFICATION OF HEART DISEASES WITH CONVOLUTIONAL NEURAL NETWORKS



BEKIR YAVUZ KOÇ
ASSOC.PROF.DR. TANER ARSAN

MASTER'S DEGREE THESIS

SUBMITTED TO THE SCHOOL OF GRADUATE STUDIES
WITH THE AIM TO MEET THE PARTIAL REQUIREMENTS REQUIRED TO
RECEIVE A MASTER'S DEGREE IN THE PROGRAM OF COMPUTER
ENGINEERING

ISTANBUL, FEBRUARY, 2021

NOTICE ON RESEARCH ETHICS AND
PUBLISHING METHODS

I, BEKIR YAVUZ KOÇ;

- hereby acknowledge, agree and undertake that this PhD / Master's Degree Thesis / Dissertation that I have prepared is entirely my own work and I have declared the citations from other studies in the bibliography in accordance with the rules;
- that this PhD / Master's Degree Thesis / Dissertation does not contain any material from any research submitted or accepted to obtain a degree or diploma at another educational institution;
- and that I commit and undertake to follow the "Kadir Has University Academic Codes of Conduct" prepared in accordance with the "Higher Education Council Codes of Conduct".

In addition, I acknowledge that any claim of irregularity that may arise in relation to this work will result in a disciplinary action in accordance with the university legislation.

BEKIR YAVUZ KOÇ

3 FEBRUARY,2021 AND SIGNATURE

ACCEPTANCE AND APPROVAL

This study, titled **CLASSIFICATION OF HEART DISEASES WITH CONVOLUTIONAL NEURAL NETWORKS**, prepared by the **BEKIR YAVUZ KOÇ**, was deemed successful with the **UNANIMOUS VOTING** as a result of the thesis defense examination held on the **FEBRUARY 3RD, 2021** and approved as a **MASTER'S DEGREE THESIS** by our jury.

JURY:

SIGNATURE:

Assoc.Prof.Dr. Taner Arsan (Advisor) (Kadir Has University) _____

Emeritus.Prof.Dr. Önder Pekcan (Kadir Has University) _____

Assoc.Prof.Dr. Osman Kaan Erol (Istanbul Technical University) _____

Asst.Prof.Dr. Figen Özen (Haliç University) _____

Asst.Prof. Ayşe Bahar Delibaş (Kadir Has University) _____

I confirm that the signatures above belong to the aforementioned faculty members.

Prof.Dr. Emine Füsün Alioğlu

Director of the School of Graduate Studies

APPROVAL DATE: Day/Month/Year

TABLE of CONTENTS

| | |
|---|-----|
| LIST OF TABLES | i |
| LIST OF FIGURES | ii |
| LIST OF ABBREVIATIONS | iv |
| ABSTRACT | v |
| ÖZET..... | vi |
| ACKNOWLEDGEMENT..... | vii |
| 1. INTRODUCTION..... | 1 |
| 1.1 PROBLEM DEFINITION AND RESEARCH GAP | 2 |
| 1.2 RESEARCH GOAL | 3 |
| 2. LITERATURE REVIEW..... | 4 |
| 2.1 SAMPLE PROJECT | 10 |
| 3. METHODOLOGY..... | 12 |
| 3.1 THIRD DERIVATE TAYLOR SERIES | 12 |
| 3.2 MACHINE LEARNING | 13 |
| 3.2.1 History of Machine Learning..... | 14 |
| 3.2.2 Supervised Learning | 16 |
| 3.2.3 Unsupervised Learning..... | 16 |
| 3.2.4 Semi-Supervised Learning | 17 |
| 3.2.5 Reinforcement Learning..... | 17 |
| 3.3 DEEP LEARNING | 17 |
| 3.3.1 Biological Neurons | 18 |
| 3.3.2 Artificial Neural Networks | 19 |
| 3.3.2.1 Feedforward..... | 20 |
| 3.3.2.2 Backpropagation | 23 |
| 3.3.3 Convolutional Neural Networks | 25 |
| 3.3.3.1 Convolutional Layer | 26 |
| 3.3.3.2 Activation Functions and ReLU | 31 |
| 3.3.3.3 Pooling Layer..... | 32 |
| 3.3.3.4 Fully Connected Layers..... | 34 |

| | |
|---|-----------|
| 3.3.3.5 Model Training and Loss Function | 36 |
| 3.3.3.6 Preventing Overfitting in Convolutional Neural Networks | 37 |
| 4. CONVOLUTIONAL NEURAL NETWORKS MODELS AND RESULTS..... | 39 |
| 4.1 DATABASE | 39 |
| 4.2 MODEL TRAINING | 43 |
| 4.3 CNN MODELS WITH SAMPLE PROJECT | 44 |
| 4.3.1 ECG Model & Results in Sample Project Dataset | 45 |
| 4.3.2 Phase Space Model & Results in Sample Project Dataset..... | 49 |
| 4.4 CNN MODELS WITH ALTERED TRAINING AND TEST SET | 52 |
| 4.4.1 ECG Model & Results in Altered Training and Test Set | 53 |
| 4.4.2 Phase Space Model & Results in Altered Training and Test Set..... | 55 |
| 5. CONCLUSION..... | 59 |
| 5.1 COMPARISON CNN RESULTS WITH SAMPLE PROJECT DATASET | 59 |
| 5.2 COMPARISON CNN RESULTS WITH ALTERED DATASET | 61 |
| 5.3 MOST ACCURATE MODEL | 62 |
| 5.4 SUGGESTIONS | 63 |
| BIBLIOGRAPHY | 64 |
| CURRICULUM VITAE..... | 72 |

LIST of TABLES

| | |
|---|----|
| Table 4.1 Training Set and Test Set | 45 |
| Table 4.2 Model Training | 52 |



LIST of FIGURES

| | |
|---|----|
| Figure 1.1 Number of Deaths by cause in the World, 2017..... | 1 |
| Figure 1.2 Cardiac cycle of an ECG | 2 |
| Figure 3.1 Connection of neurons..... | 18 |
| Figure 3.2 A neural network block diagram example..... | 19 |
| Figure 3.3 Activation Function | 20 |
| Figure 3.4 Sigmoid Function | 21 |
| Figure 3.5 Basic structure of a neural network with hidden layer | 22 |
| Figure 3.6 Backpropagation in an artificial neural network | 24 |
| Figure 3.7 General Structure of Convolutional Neural Networks | 25 |
| Figure 3.8 Feature map in the convolution layer | 27 |
| Figure 3.9 Input Matrix and Filter Matrix..... | 27 |
| Figure 3.10 Steps of matrix Y calculation | 30 |
| Figure 3.11 Output Matrix after feature detection | 30 |
| Figure 3.12 Stride Lengths example | 31 |
| Figure 3.13 Rectified Linear Unit (ReLU)..... | 32 |
| Figure 3.14 Max Pooling Process in CNN..... | 33 |
| Figure 3.15 Average Pooling Process in CNN..... | 34 |
| Figure 3.16 Flattening in CNN..... | 35 |
| Figure 3.17 Fully connected layers | 35 |
| Figure 3.18 Dropout Function..... | 37 |
| Figure 3.19 Data Augmentation Function..... | 38 |
| Figure 4.1 ECG Line Diagram of Healthy Records..... | 40 |
| Figure 4.2 Phase Space Diagram of Healthy Records | 40 |
| Figure 4.3 ECG Lines of Unhealthy Records | 41 |
| Figure 4.4 Phase Space Diagram of Unhealthy Records | 42 |
| Figure 4.5 Architecture of CNN model..... | 44 |
| Figure 4.6 ECG lines accuracy comparison of training set and testing set in sample project dataset..... | 46 |

| | |
|--|----|
| Figure 4.7 ECG lines model loss comparison of training set and testing set in sample project dataset..... | 47 |
| Figure 4.8 ECG lines model outputs of training set and testing set in sample project dataset..... | 48 |
| Figure 4.9 Phase space model accuracy comparison of training set and testing set in sample project dataset | 49 |
| Figure 4.10 Phase Space model loss comparison of training set and testing set in sample project dataset..... | 50 |
| Figure 4.11 Phase Space model outputs of training set and testing set in sample project dataset..... | 51 |
| Figure 4.12 ECG lines accuracy comparison of training set and testing set in altered dataset..... | 53 |
| Figure 4.13 ECG lines model loss comparison of training set and testing set in altered dataset..... | 54 |
| Figure 4.14 ECG lines model outputs of training set and testing set in altered dataset.. | 55 |
| Figure 4.15 Phase Space model loss comparison of training set and testing set in altered dataset..... | 56 |
| Figure 4.16 Phase Space accuracy comparison of training set and testing set in altered dataset..... | 57 |
| Figure 4.17 Phase Space model outputs of training set and testing set in altered dataset | 57 |
| Figure 5.1 ECG Line Diagram of Healthy Records | 59 |

LIST of ABBREVIATIONS

| | |
|---------|--|
| AAMI: | Association for the Advancement of Medical Instrumentation |
| ACS: | Acute Coronary Syndrome |
| ANN: | Artificial Neural Network |
| ANSI: | American National Standards Institute |
| AUC: | Area Under Curve |
| CNN: | Convolutional Neural Network |
| CVD: | Cardiovascular diseases |
| CWT: | Continuous Time Wavelet Transform |
| db1: | Daubechies wavelet function |
| ECG: | Electrocardiography |
| FC: | Fully Connected |
| HOS: | High Order Statistics |
| IBM: | International Business Machines |
| IHME: | Institute for Health Metrics and Evaluation |
| K-NN: | K-Nearest Neighborhood |
| LBP: | Local Binary Patterns |
| LMN: | Levenberg–Marquardt Neural Network |
| LVH: | Left Ventricular Hypertrophy |
| MIL: | Multiple Instance Learning |
| MLP: | Multilayer Artificial Neural Network |
| MSE: | Mean Squared Error |
| OPF: | Optimum Path Forest |
| RBF: | Radial Based Function |
| RBM: | Restricted Boltzmann Machine |
| ReLU: | Rectified Linear Unit |
| RNN: | Recurrent Neural Networks |
| ROC: | Receiver Operating Characteristics |
| RPROP: | Resilient Backpropagation Neural Network |
| SCGNN: | Scaled Conjugate Gradient Backpropagation Neural Network |
| STFT: | Short-Time Fourier Transform |
| SVM: | Support Vector Machine |
| WAPSYS: | Waveform Separation System |
| WHO: | World Health Organization |
| QGMSVM: | Q-Gaussian Multi-Class Support Vector Machine |

CLASSIFICATION OF HEART DISEASES WITH CONVOLUTIONAL NEURAL NETWORKS

ABSTRACT

Nowadays, the number and frequency of heart diseases is increasing. High amounts of expenses are incurred in order to make improvements in this area. The beats in the electrical conduction of the heart can be recorded by special devices and ECG (Electrocardiogram) can be created. Data generated from ECG can be transformed into phase spaces with Taylor Series algorithm. In order to determine the detection of heart disease, ECG and phase spaces were created from MLII signals based on 44 different records. Both ECG images and phase space images were used to determine the heart conditions of these recordings. The heart status of the recordings was measured by applying Convolutional Neural Networks (CNNs) method to the images and results compared with the SVM (Support Vector Machine) algorithm. In addition, the success rates of different models were compared by changing the training and test set over the same records. The success rate between ECG and phase space was also determined.

Keywords: Electrocardiogram, Phase Space, Convolutional Neural Networks, Support Vector Machine

EVRIŐİMLİ SİNİR AĐLARI İLE KALP RAHATSIZLIKLARININ SINIFLANDIRILMASI

ÖZET

Günümüzde kalp hastalıklarının sayısı ve sıklığı artmaktadır. Bu alanda iyileřtirmeler yapılabilmesi için yüksek miktarda harcama yapılmaktadır. Kalbin elektriksel iletimindeki atımlar özel cihazlarla kaydedilebilir ve EKG (Elektrokardiyogram) oluşturulabilir. EKG'den üretilen veriler, Taylor Series algoritması ile faz uzaylarına dönüřtürülebilir. Kalp hastalığının tespiti için 44 farklı kiřiden alınan verilerle MLII sinyallerinden EKG ve faz uzayları oluşturuldu. Bu kayıtların kalp durumunu belirlemek için hem EKG görüntüleri hem de faz uzayı görüntüleri kullanıldı. Kayıtların kalp durumu görüntülere ve sonuçlara Convolutional Neural Networks (CNNs) yöntemi uygulandı ve SVM (Support Vector Machine) algoritması ile karşılaştırılarak başarı oranı ölçüldü. Ayrıca aynı kayıtlar üzerinden eğitim ve test seti deđiřtirilerek farklı modellerin başarı oranları karşılaştırıldı. EKG ile faz uzayı görüntülerine CNN algoritmasının verdiđi sonuçlardaki farklılık tespit edildi.

Anahtar Sözcükler: Elektrokardiyogram, Faz Uzayı, Evriřimli Sinir Ađları, Destek Vektör Makinesi

ACKNOWLEDGEMENT

I would like to thank my supervisor Assoc.Prof.Dr. Taner ARSAN for all his help and advice with this master thesis. I would also like to thank my sister, whom without this would have not been possible. I also appreciate all the support I received from the rest of my family.

1. INTRODUCTION

The conduction system of the heart has important role in human health. Any blockage in the arteries as a result of the blood being pumped from the arteries to the body through the heart, possibly cause fatal heart attacks [1]. World Health Organization (WHO), specified that cardiovascular diseases (CVDs) are the biggest cause of death in the world.

A group of disorders of the heart and blood vessels can be called as cardiovascular diseases. It includes coronary rheumatic heart, cerebrovascular disease and heart disease with other diseases. CVDs caused eighty percent of deaths due to strokes and heart attacks, and thirty-three percent of the deaths occur in people who is younger than 70 years old. Some of the major causes of heart diseases are obesity, diabetes smoking, hypertension. Also, physiological factors, HIV or AIDS and vitamin D/B12 deficiencies can be the factors. Heart rate disturbances may require immediate measures and special attention [2]. Number of deaths by cause can be observed from Figure 1.1, which shows that CVDs cause nearly 18 million people to die in each year [3].

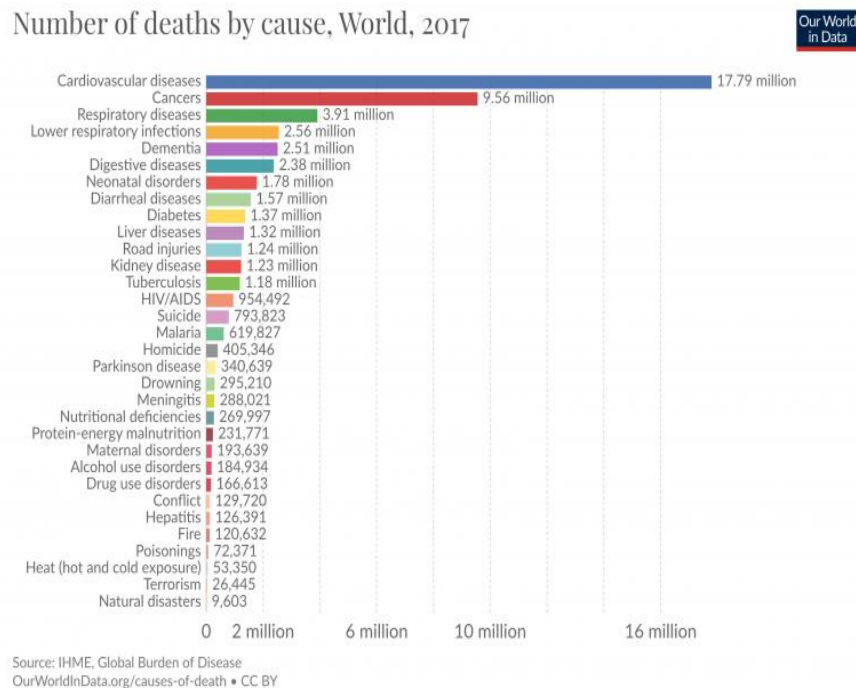


Figure 1.1. Number of Deaths by cause in the World, 2017

1.1 PROBLEM DEFINITION AND RESEARCH GAP

Electrocardiogram (ECG) is a detailed measurement method useful in diagnosing heart disease [4]. ECG can be recorded by twelve different electrodes. These electrodes are used to measure and analyze a standard cardiac activity from the patient. The 12-lead ECG consists of three bipolar limb leads (I, II and III), unipolar limb leads (AVR, AVL, and AVF), as well as precordial or V leads (V1, V2, V3, V4, V5 and V6) [5].

Among the all leads, most commonly used lead is lead II, because it provides a good view of P, Q, R, S and T waves, which are the most important waves [5]. Each beat of the heart reflects voltage versus the time revolution of electrical activity in the heart. It produces deviations or waves away from the baseline and this is measured on the ECG. A standard ECG recording includes a P wave, QRS section followed by the T wave sequence [6]. Depolarization of the atria can be represented with the P wave. Depolarization of the ventricles can be represented with the QRS complex. The repolarization of the ventricles can be represented with the T wave. PQRST cardiac cycle is indicated with Figure 1.2.

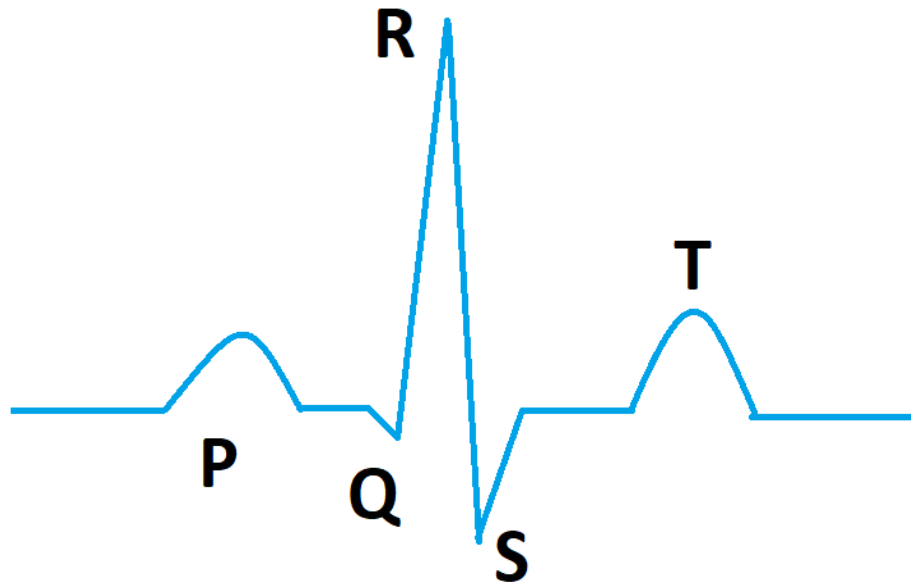


Figure 1.2. Cardiac cycle of an ECG

PR interval starts in the ending phase of P wave and finishes at beginning of the Q wave. The ST phase begins after ending of S wave and finishes at the starting of the T wave [7].

ECG data can be investigated in more detail to observe normal and irregular cardiac activities. There are many studies using different approaches to classify ECG data with machine learning or deep learning models.

1.2 RESEARCH GOAL

The first goal of this research is to calculate the classification accuracy differences between ECG signals and the phase space diagrams with the related data used for images generated by Taylor-Series algorithm.

The second goal of this research is to calculate the classification accuracy differences between Convolutional Neural Networks (CNNs) and Support Vector Machine (SVM) algorithms with using same dataset, which is MIT-BIH Arrhythmia Database published on 24 February 2005 on Physio.net.

2. LITERATURE REVIEW

Different studies have been carried out in attribute extraction and classification methods in the classification problem of ECG lines.

In 1983, the ECG signal is shown in the form of and / or graphic and the classification process is done in the form of graphic research by Stockman. A software system called WAPSYS (waveform separation system) has been defined that implements this structural analysis paradigm [8].

In 1990, two signals collected in Holter ECG systems and signal compression applied. A three-layer ANN with a hidden layer with several units was used to extract the attributes of the ECG waveform as a function of the activation levels of the hidden layer units by Iwata. The number of exit and entry units was the equal. The backpropagation algorithm was applied in purpose of model to learn. The artificial neural network was set up by supervised signals that are the equal with input signals. One network was used for data compression and the other network was used for learning with available signals [9].

In 1992, Xue's study is about the QRS detection area. An artificial neural network adaptive whitening filter was applied to model nonlinear and non-stationary ECG low frequencies. In order to detect QRS, the signal containing mostly higher frequency QRS complex energy was then passed through a linear matched filter. An algorithm was developed to adapt the matching filter template from the detected QRS complex in the ECG signal, so the template could be personalized to the individual [10].

In 1993, ECG signal is defined as if-then rules. In this study by Kundu, the decision-making mechanism of the system is based on the verification method. The control module is designed to be distributed between the two factors, taking into account production level parallelism and state level parallelism in this unifying-decoding-process cycle. It works with a special field dependent algorithm to speed up the inference procedure for online interpretation of the signal [11].

In 1997, the use of neural networks in detecting ventricular late potentials has been studied by Xue and Reddy. High frequency low amplitude signals obtained from signal averaged ECGs is called ventricular late potentials. This data is useful in identifying patients prone to natural or inducible ventricular tachycardia in the electrophysiology testing. Supervised and self-organizing artificial neural network models were developed to detect patients with positive electrophysiology tests for inducible ventricular tachycardia in patients with negative electrophysiology testing using ventricular late potentials. The morphology information of the vector magnitude waveform was added to an original set of three time domains of late potentials, with, high frequency low amplitude signal duration, total QRS duration and root-mean-square voltage [12].

In the study conducted by Lena Biel in 2001, a 12-lead ECG records of 20 different people were measured. The ages of these people varied between 20 and 55. 30 different features were created from the digitized ECG signal. Classification has been applied using SIMCA (Soft independent modelling of class analogies). As a result, it is shown that a person can be identified with a single lead ECG, with only three electrodes can be attached on the body [13].

In 2004, Jekova and Krasteva proposed an algorithm for detecting ventricular fibrillation and tachycardia using a band pass digital filter with integer coefficients, which was very simple to implement in real time study. The wave detection class has been activated for auxiliary parameter and heart rate measurement calculation. ECG recordings from recognized databases of the American Heart Association and Massachusetts Institute of Technology tested with the method. It achieved a sensitivity of 95.93% and a specificity of 94.38% [14].

In 2005, the peaks of the P, QRS and T waves, the PR and QT intervals, the size of the P, QRS complex and the T waves, their frequencies, as well as the features of the ST and PQ segments are classified by decision trees in Rodriguez study. Accuracy is 97.95% for arrhythmias requiring medical attention in less than an hour and 95% for generally bad arrhythmias [15].

In 2010, Ye proposed a new technique to arrhythmia classification. The technique was dependent on dynamic features and morphological features combination. RR interval information characterizing the rhythm around the corresponding heartbeat providing dynamic features was obtained. SVM was used to divide heartbeat classes into 15 classes. The proposed method has been tested with the MIT-BIH Arrhythmia Database, providing an overall accuracy of 99.66% [16].

In 2011, by Shivajirao M Jadhav modular neural network (MNN) is proposed. This model was used for classification purposes of normal and abnormal hearts of arrhythmia. UCI Arrhythmia is tested with this study as a dataset. From four hundred and fifty-two patients, ambiguous and incomplete bio-signal data is collected. A neural network was constructed with one, two and three different hidden layers. The models trained by data set partitions. The models performance was tested with six measures; mean squared error (MSE), area under curve (AUC), receiver operating characteristics (ROC), specificity, classification accuracy and sensitivity. The results showed that the models had more than 82% success [17].

In 2012, Sun's study presented a technique for detecting myocardial infarction. With this technique, with the ECG lines of the patients, an automatic detection could be carried out. Multiple Instance Learning(MIL) was used in this purpose. A latent topic of MIL is proposed and this was a new strategy in 2012. Electrocardiogram lines mapped to into a topic space. ECGs were defined by a number of topics identified. Support vector machine algorithm is applied with the unlabeled heartbeats of training data to the ECG-level topic vectors. For testing results PTB diagnostic database is used [18].

In 2013, E. J. da S., Nunes applied and analyzed optimum-path forest (OPF) classifier which is graph based pattern recognition technique. The classifier specificity, sensitivity, accuracy and time performance was compared to multilayer artificial neural network (MLP), Bayesian and support vector machine (SVM). For testing purposes, MIT-BIH Arrhythmia Database was used [19].

In 2014, Huanhuan conducted a study with deep belief networks. In Huanhuan study, this method was used classifying to electrocardiogram beats. Several classifiers are tested with MIT-BIH Database with extraction of attributes from the original electrocardiogram signals. Nonlinear support vector machine with Gaussian kernel achieved the optimum accuracy, with 98.49% [20].

In 2015, for two-lead heart beat classification purposes, a machine learning algorithm which is, restricted Boltzmann machine was proposed by Chauhan. Restricted Boltzmann machine is an unsupervised learning algorithm. It was one of the best algorithm that mining of very large un-labelled ECG wave beats. Heart healthcare monitoring applications collected these beats. RMB based algorithm was applied for classification. RMB based algorithm was tested on the two-lead ECG dataset of MIT-BIH. It had accuracy with nearly 99% [21].

In 2016, with the spread of machine learning studies, various artificial intelligence solutions, creation and classification of attribute templates from ECG signs had successful results in solving the problem. Feature vectors were obtained from ECG signals obtained from 100 people by using Convolutional Neural Network architecture and 1 frequency domain feature extraction module by Lei's study. The classification performance with the nonlinear SVM classifier was calculated as 99.33% using 26 combined feature vectors with 1x160 length [22].

In 2017, a study was carried out with CNN architecture using different ECG databases ranging from 18 to 47 classes in 2017 by Zhang. After the application of wavelet transform and reclamation function to ECG signals, N parallel connections of four convolutional layers and one fully connected layer were provided as an architectural input, and the average person recognition rate was calculated as 93.5% [23].

In 2017, a study conducted by Eduardo which is the features of the ECG sign were extracted with a deep auto encoder using a database of 709 people, and the K-Nearest Neighborhood (KNN) method was used in classification purposes. Recognition error reached 0.91% at the end of the study [24].

In the study presented by Luz in 2017, the results obtained with 2 different models were combined with the score-level combining technique, and an equal error rate of 14% was obtained. In this study, 1-dimensional QRS signs were obtained by aligning the ECG mark according to the R points, on the other hand, the spectrogram of these QRS signs was taken and these input data were given to 1B-ESA and 2B-ESA models, respectively [25].

In the same year, a study conducted by Salloum, Recurrent Neural Networks-RNNs were used as the attribute and classification method and the system test was performed on MIT-BIH Arrhythmia (MITDB) and ECG-ID databases. For 47 and 90 classes in these databases, the test performance of the model reached almost 100% [26].

In 2018, a new community learning based method has been proposed by Yakut. This method could automatically classify heartbeats with arrhythmia based on the category and patient-based evaluation plan. Power spectral density-based feature extraction technique has been used in this study. Hybrid sub-attribute sets were created using the wrapper attribute selection algorithm. A new community learning-based method has been proposed, which is created using the stacking algorithm of basic learners. Multi-layer perceptron and random forest, meta learner linear regression classifiers. As a result, the average performance values of the proposed new method for the category-based classification of heartbeats with arrhythmias was 99.88% [27].

In 2018, a technique has been developed for diagnosing acute myocardial ischemia. This technique could automatically detect changes in the ST / T part of the ECG signal., ECG records in the STAFF III database are used in this study. By using support vector machines (SVM) with linear and radial based function (RBF) cores, classifiers that are using two and four attributes with the highest discrimination of the ST / T part of the ECG signal have been designed. As a result of the application of the developed technique on ECG signals in the STAFF III database, the results obtained from a significant number of patients showed a successful result in order to detect acute myocardial ischemia [28].

In the study conducted by Labati, QRS segments were obtained from the ECG mark for 52 people, as in some previous methods, and the reciprocal coefficients between these signs were calculated. With reference to these coefficients, some QRS signals were selected and added consecutively, and a one-dimensional autocorrelation vector was obtained. This vector, which was used as an introduction, was given as an introduction to the developed 1-D ESA architecture and a person recognition performance of 100% was achieved [29].

In the study conducted by Abdeldayem in 2018, an 2D-ESA architecture developed by applying Short-Time Fourier Transform (STFT) and Continuous Time Wavelet Transform (CWT) to the ECG signal. For STFT and CWT images, the average recognition rate of the system is 97.85% and 97.50%, respectively [30].

In 2019, in order to realize the robust detection of acute coronary syndrome, a technique has been developed that detects anomalies in the ST segment and T wave of the ECG signal using current signal processing and machine learning techniques. For this purpose, a unique feature extraction technique has been developed to obtain the highest differential ECG features for the diagnosis of ACS by using the broadband recordings in the STAFF III database. With the obtained critical features, a supervised learning technique with SVM and kernel functions that performs detection of ACS has been developed [31].

In the study conducted by Hammad in 2019, the performance of the biometric recognition system was tested using the ECG mark and fingerprint images. In this study, feature extraction was performed by using 1-D ESA architecture with one-dimensional ECG signals and 2D-ESA architecture was used for fingerprint marks. The feature vectors obtained from both biometric signs were classified using the Q-Gaussian Multi-Class Support Vector Machine (QGMSVM), combined with score merging method and the system accuracy rate was calculated as 99.99% [32].

In the study conducted by Jothiramalingam in 2020, in order to achieve method for detecting Left Ventricular Hypertrophy (LVH). This method was applied to detect LVH

from multi-lead ECG lines, with KNN, Ensemble of Bagged Tree, AdaBoost and SVM classifiers. Results are compared with different neural network classifiers which are Scaled Conjugate Gradient Backpropagation Neural Network (SCGNN), MLP, Resilient Backpropagation Neural network (RPROP) and Levenberg–Marquardt Neural Network (LMNN). The accuracy for detection of LVH is 75.6%, 84.4%, 86.6%, 88.9%, 93.3%, 95.6%, 97.8%, 97.8%, using AdaBoost, KNN, SVM, RPROP, Ensemble of Bagged Tree, MLP, SCGNN and LMNN classifiers, respectively [33].

2.1 SAMPLE PROJECT

This thesis's sample project is "Heartbeat classification fusing temporal and morphological information of ECGs via ensemble of classifiers" study which is published in 2019. In Mondéjar-Guerra study, ECG based automatic classifications are combined with multi-ended SVMs. This methodology is based on the time interval of subsequent beats and beats morphology in electrocardiogram characterization. Different descriptors based on wavelets, local binary patterns (LBP) and high order statistics (HOS) values were used in the study. Rather than the concatenating process of all features in a single support vector machine model, these features are used in a different way. They are used as an input in order to teach the different support vector machine models for different features. Before getting the ultimate classifying result, these models are joint with the sum, majority and product methodologies.

The main goal of study was testing the efficiency of implementing multiple support vector machine models as a classification in arrhythmias. Different and more than one features are created in ECG signals. These features are created and used with wavelets, R-R intervals, LBP and HOS.

MIT-BIH database has forty-eight ECG records which are record in thirty minutes, from forty-seven different people. Only the MLII signal is provided by almost all the records. It totally includes nearly a hundred ten thousand beats. The beats are grouped as SVEB (Supraventricular ectopic beat), VEB (Ventricular ectopic beat), F (Fusion), normal and unknown. Record 102, 104, 107 and 217 did not used in the study, due to

they have pace hearts. Approximately ninety percent of the beats are normal beats. One percent of beats composed of fusion, three percent beats composed of supraventricular ectopic and six percent of the remaining beats composed of ventricular ectopic beats. Test and training set consist of twenty-two different records.

In preprocessing step, 200-ms and 600-ms consecutive filters were used to calculate baseline of electrocardiogram signal. After this process, it was removed from the real electrocardiogram signal. All features extracted and calculated in a smaller area of the signals.

In the study, the Daubechies wavelet function (db1) is used and made a twenty-three dimensional descriptor with three levels of decomposition. With the division of each beat in five intervals, a ten dimensional attribute is created in order to use in HOS. With LBP which includes eight neighbor uniform code, fifty-nine dimensional descriptor used with 1D-LBP. Also, a morphological descriptor is implemented in the study. It is dependent on the distance of sample and amplitude by calculating Euclidean distance in the R-peak and the other four waves of the beat.

In the study, it is reported that the best parameters were adjusted and the models were trained again with training set and tested with testing set. It is specified that because of unbalance dataset, the mean or overall accuracy did not show the true performance of classifier. Also it is indicated that if the classifier assigned to a normal testing class, it is expected that it would classify with at least eight-nine percent overall accuracy [34].

3. METHODOLOGY

3.1 THIRD DERIVATE TAYLOR SERIES

Taylor series mathematics is the expansion of a function in the form of an infinite sum of the terms of that function calculated from its derivative values at a single point. It is named after British Mathematician Brook Taylor. Using a finite number of terms of a series is a general method for converging that series to a function. The Taylor series can also be seen as the limit of the Taylor polynomial. Taylor series can be used to find the numerical values of functions at a given point. In addition, derivative or integral operations can be opened to series and operations can be done more easily.

The simple approximation of the first derivative of a function f at a point x is specified as the limit of a difference quotient as follows:

$$f'(x) = \lim_{h \rightarrow 0} \frac{f(x+h) - f(x)}{h} \quad (3.1)$$

If $h > 0$, meaning that h is a finite positive number, then

$$f'(x) = \frac{f(x+h) - f(x)}{h} \quad (3.2)$$

is called the first-order forward difference approximation of $f'(x)$. The approximation of $f'(x)$ can be obtained by combining Taylor series expansions [35, 36]. If the values (x_{i+1}, f_{i+1}) , (x_{i+2}, f_{i+2}) and (x_{i+3}, f_{i+3}) are known, the first order derivative of $f_i(x)$ can be calculated as shown by $f'_i(x)$.

$$f_{i+1} = f_i + \frac{f'_i}{1!} * h + \frac{f''_i}{2!} * h^2 + \frac{f'''_i}{3!} * h^3 + \dots + \frac{f_i^n}{1!} * h^n + R_n \quad (3.3)$$

$$(x_{i+1}, f_{i+1}) \rightarrow f_{i+1} = f_i + \frac{f_i'}{1!} * h + \frac{f_i''}{2!} * h^2 \quad (3.4)$$

$$f_{i+1} = f_i + h * f_i' + \frac{h^2}{2!} * f_i''$$

$$(x_{i+2}, f_{i+2}) \rightarrow f_{i+2} = f_i + \frac{f_i'}{1!} * 2h + \frac{f_i''}{2!} * (2h)^2 \quad (3.5)$$

$$f_{i+2} = f_i + 2 * h * f_i' + 2 * h^2 * f_i''$$

$$(x_{i+3}, f_{i+3}) \rightarrow f_{i+3} = f_i + \frac{f_i'}{1!} * 3h + \frac{f_i''}{2!} * (3h)^2 \quad (3.6)$$

$$f_{i+3} = f_i + 3 * h * f_i' + \frac{9}{2} * h^2 * f_i''$$

The second derivative f_i'' is canceled by adding Equations 3.4, 3.5 and 3.6 after multiplying them by 18, -9 and 2, respectively. After this step we obtain:

$$18 * f_{i+1} - 9 * f_{i+2} + 2 * f_{i+3} = 11 * f_i + 6 * h * f_i' \quad (3.7)$$

Finally, third-order forward difference Taylor series approximation of $f'(x)$ is obtained as follows:

$$f_i' = \frac{1}{6h} (-11 * f_i + 18 * f_{i+1} - 9 * f_{i+2} + 2 * f_{i+3}) \quad (3.8)$$

3.2 MACHINE LEARNING

Machine Learning is the science of teaching computers to learn and act like humans. Its aims computers to improve their learning rate by input data and information, autonomously [37]. With machine learning, it is ensured that the experiences obtained from previous examples are taught to the computer. Due to that, this event can be described as learning from experience. Machine learning is used in different fields as automotive, entertainment, marketing industry, science and medicine industries [38].

There are certain basic steps to be followed in the face of a problem to be solved in machine learning. It is important to take these steps in order to successfully solve the existing problem at the desired time [38]. These steps are:

- Defining the problem
- Data analysis
- Data preparation
- Establishing the model
- Evaluating the model
- Usage of the model

Machine learning methods are divided into two as supervised and unsupervised learning, then the classification of machine learning methods has been formed by adding semi-supervised and reinforcement learning.

3.2.1 History of Machine Learning

1943 → The first example of neural networks created by Warren McCulloch and Walter Pitts. They specified about neurons and neurons working system in their paper. They build a model with an electrical circuit and created the neural network which is basis of machine learning.

1950 → Alan Turing created Turing Test. The purpose of this test is designed in a simple logic. In order to passing this test, computer had to convince a person that it is a computer.

1952 → A computer program that plays checkers developed by Arthur Samuel. This program was playing the checkers with itself and learning from the its previous checker games.

1958 → Perceptron was designed by Frank Rosenblatt which was first artificial neural network. Its main purpose was shape recognition.

1959 → Bernard Widrow and Marcian Hoff designed two models. The first was called ADELIN which could detect binary patterns. As an example, is to predict which bit will come after one bit in a bit stream. This modeling was developed and named as the new generation MADELINE. With this modeling, elimination of echo in telephone lines could be possible and this modeling is still being used today.

1982 → John Hopfield tried to build a bidirectional network similar to the logic of neurons. Also this year, Japan announced its focus on more advanced neural networks with more research in the region.

1986 → The algorithm, which was created in 1962 by three researchers from the Stanford Psychology Department, was expanded and the back propagation of advanced neural networks was discovered.

1997 → Deep Blue, the IBM computer which is a chess playing computer, defeated the world champion.

2012 → Deep neural network created by Google's Jeff Dean. The model focused on pattern detection in videos and images. With using Google's resources, it has become incomparable to much smaller neural networks. Also, the model used to detect objects in YouTube for a period of time.

ImageNet competition winner was AlexNet. This situation led to usage of GPUs and CNN in machine learning. One of the biggest reason that AlexNet won this competition that they created the ReLU activation function and increased the efficiency of CNN algorithm.

2014 → In order to recognition of people with the same precision as a human, Alexnet, a deep neural network is developed by Facebook.

DeepMind was acquired by Google. It could play simple video games as much as humans.

2015 → OpenAI is created by Elon Musk, in order to create safe artificial intelligence that can benefit humanity.

2016 → Google's AlphaGo could defeat a professional Go player, which is one of the hardest board games in the world [39].

2019 → Google's AlphaStar played anonymously against human players in StarCraft II which is a real-time strategy video game. This games are played on Battle.net servers and AlphaStar ended up ranked in the top 200 players. This situation made AlphaStar to be placed in the top 0.2 % of all the players. With this success, AlphaStar could be able to play StarCraft II better than almost all human players [40].

3.2.2 Supervised Learning

It is a learning method based on loading the training data set and test data set into the system, making the necessary groupings on each data and thus establishing a connection between the input data set and the output data set. Its main purpose is to find predictions on the data set with unknown results by classifying the data set with known results. Decision trees, support vector machines, nearest k-neighbor, artificial neural networks, genetic algorithms, bayes classifiers can be given as examples of supervised learning methods [41].

There are two type of supervised learning [42].

Classification: Predicting the outcome of the situation in which there are output variables in categories.

Regression: Predicting the result of a particular example where the given variable is in the form of real values.

3.2.3 Unsupervised Learning

Unsupervised learning contains input variables, not output variables. Data sets without categories are used to model data into this structure. Correlation analysis, factor

analysis, cluster analysis, self-regulating maps, k-mean unsupervised learning can be given as examples. Especially, unsupervised learning is used for clustering. Clustering can separate samples from a set that are unrelated and put them together with samples that are close to each other [38].

3.2.4 Semi-Supervised Learning

It is a learning class that tries to increase efficiency compared to unsupervised learning by using unlabeled data and slightly tagged data for education. According to the data obtained as a result of the studies, significant improvements were observed in learning accuracy when unsupervised learning data were associated with classified data. Semi-supervised learning is between supervised and unsupervised learning [38].

3.2.5 Reinforcement Learning

Reinforcement Learning is a method where machine learning operations are performed by determining the most effective process in the current situation using more than one method. Written codes learn the best process with trial and error methods. It is generally used in the robotic field. As an example can be given by the process of learning that area after a robot hits an obstacle and avoiding hitting that area in the next operations is an example of reinforcement learning.

3.3 DEEP LEARNING

The ideas put forward by Alan Turing about machine learning in the mid-twentieth century continue to progress with speed. While the Turing test assumes that an artificial system will be smart if it interacts with a human, its success is related to the unawareness of the interacting person [43]. In this way, machine learning systems can be used in many areas from online searches to personalized marketing purposes, financial and economic forecasts, integrated into the society. Moreover, machine learning has begun to enter the medical world [44]. In the past, the object classification problem was solved in two stages. In the first stage, a smart feature extraction algorithm would be designed, and features were extracted from the data. Because of this feature extraction process was problem dependent, it required expert knowledge. In the second step, the features obtained from the data were given to a classifier and the process was

completed. However, with deep learning, systems that learn the features required to solve the problem from data without requiring any expert knowledge began to emerge [45].

3.3.1 Biological Neurons

It is believed by researchers that the basic processing unit is the neurons, but the details are still not fully elucidated yet. The figure 3.1 shows the parts of biological neurons called dendrites and axons communicate with each other. The signal perceived by the dendritic part of a neuron is evaluated in the neuron, and the exit signal obtained is transmitted to the dendritic part of another neuron by the axon. In other words, a neuron can be thought of as a simple system whose input channels are dendrites and the exit channel is axon. The junction points between these dendrite and axon branches are called synapses. The most important function of the synapse is that it can scale the incoming x_i signal. This scaling is modeled as w_i weighting coefficients. It is accepted that the human brain can be able to learn as a result of changes in the weights at the synapse points in these nerve cells.

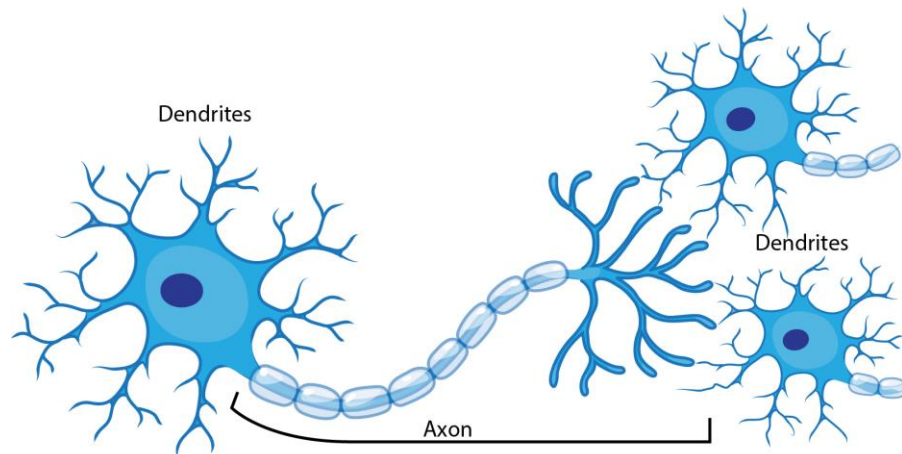


Figure 3.1. Connection of neurons [46]

It has been stated by researchers that the human brain has approximately 86 billion neurons and more than 500 trillion connections. Neurons, the main structure of the

artificial neural networks, are formed by modeling the structure and functioning of biological neurons in the brain. These artificial neurons have inputs and transformation functions just like biological neurons [47].

3.3.2 Artificial Neural Networks

A simple neural network can be developed with input layer, output layer and one or more hidden layers. The weighted features from this hidden layer or layers are transmitted to the output layer. With the large number of hidden layers in the network, deep neural networks are capable of learning higher-level features with higher complexity [48]. Figure 3.2 shows an example of basic neural network block diagram.

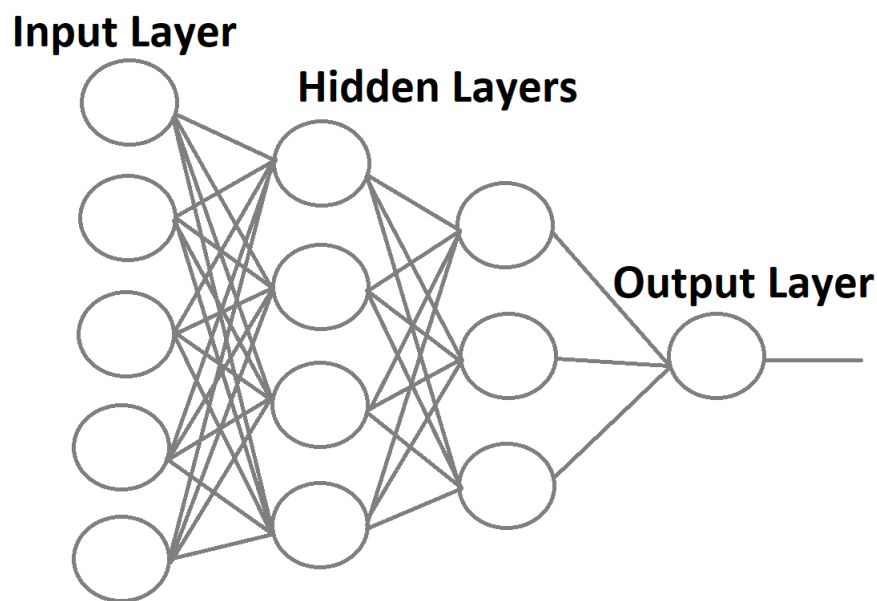


Figure 3.2. A neural network block diagram example

Neurons in the brain and their functioning are simulated in artificial neural networks. These designed neurons form the artificial neural network by establishing a connection between them in a plane and coming together in layers [49]. There are two ways to provide a neural network to learn. These are feedforward and backpropagation.

3.3.2.1 Feedforward

After the initial values are determined randomly, the information sent by the cells in the input layer (x_i) is multiplied by the weight value (w_i) of all the connections in the neuron to which it reaches, and the net input is found through the addition function. The function that plays a role in calculating the net value in the neuron may vary according to the mathematical operation performed in the designed artificial neural network. At this point, the addition function does not have a definite formula and it is aimed to achieve the best result by using trial and error method for this situation. The general formulation of the updating layers can be displayed as an equation.

$$\sum_{i=1}^n x_i w_i + b \quad (3.9)$$

The neuron reaches the net value by using the sum function as a result of various mathematical operations, and reaches the output to be produced by the activation function. Generally, the activation function is a nonlinear function. While applying this function is to choose a function that can be calculated more easily in order not to waste time during the operations performed due to the active use of derivation methods of propagation algorithms.

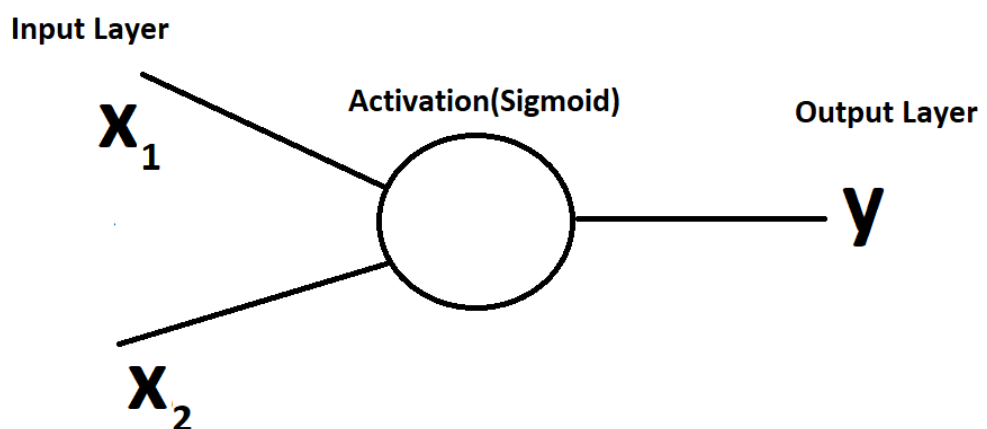


Figure 3.3. Activation Function

To give a simple example, each input is multiplied by a random weight.

$$x_1 \rightarrow x_1 * w_1$$

$$x_2 \rightarrow x_2 * w_2$$

Then, all variables with newly calculated weights are summed with a bias coefficient called a bias (b) and inserted into an activation function.

$$y = f(x_1 * w_1 + x_2 * w_2 + b)$$

The activation function is used to transform unlimited input data into a predictable form of output data. Sigmoid function is commonly used as an activation function. With Sigmoid function, input data can take place in the range of $(-\infty, \infty)$, while the output data can take place in the range of $(0,1)$ [50].

Sigmoid Function

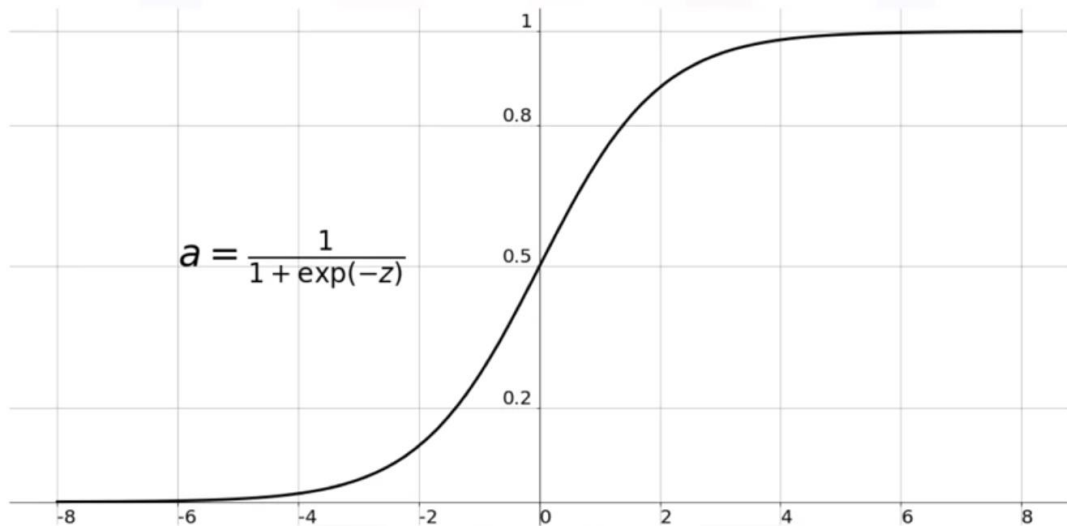


Figure 3.4. Sigmoid Function [51]

$$a = \frac{1}{1 + \exp(-z)} \quad (3.10)$$

For a mathematical example,

$$w = [w_1 \ w_2] = [0.2 \ 0.8]$$
$$b = 2$$

Suppose a neuron with two inputs has the parameters above equations and uses the sigmoid function as the activation function, when $x = [4,5]$ input is given to neuron:

$$y = f(w * x + b) = f(6.8)$$
$$f(6.8) = 0.9988$$

Multiple layers can be designed in artificial neural networks. In Figure 3.5, a basic artificial neural network structure with a input layer, hidden layer and output layer can be seen.

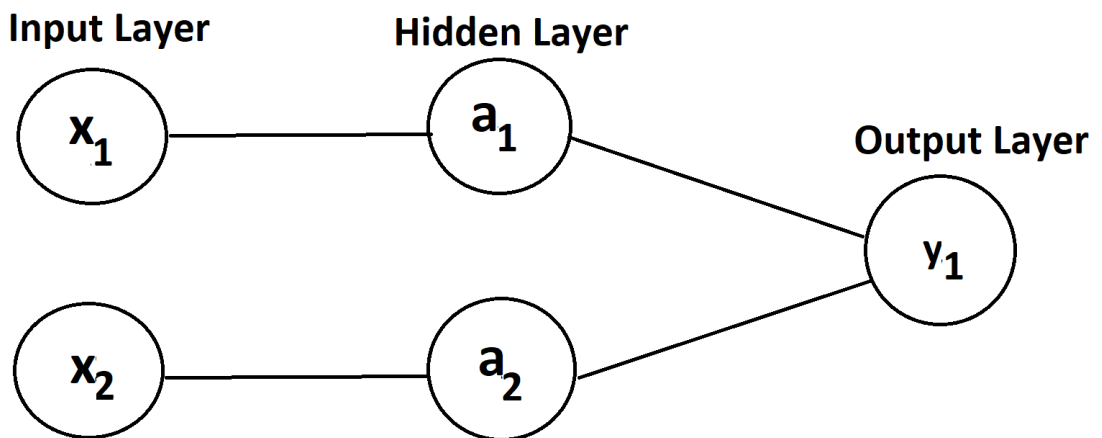


Figure 3.5. Basic structure of a neural network with hidden layer

Using the structure of artificial neural networks, assuming that all neurons are the same weight:

$$w = [w_1 \ w_2] = [0.3 \ 0.7]$$

$$b = 0.1, x = [1,3]$$

$$\begin{aligned}
 a_1 &= a_2 = f(w * x + b) \\
 &= f((0.3 * 1) + (0.7 * 3) + 0.1)
 \end{aligned}$$

$$= f(2.5) = \frac{1}{1 + e^{-2.5}} = 0.9241$$

$$y_1 = f(w * [a_1, a_2] + b)$$

$$= f((0.3 * a_1) + (0.7 * a_2) + 0.1)$$

$$= f(1.0241) = \frac{1}{1 + e^{-1.0241}} = 0.7357$$

The system can have any number of layers and any number of neurons within those layers. With interaction of neuron structures with each other, the system achieves output by assigning certain weights to each neuron.

3.3.2.2 Backpropagation

In ANN, backpropagation networks move in the opposite direction of feed forward networks. Except for the first output iteration, the difference between the actual value and the value produced by the neural network at the end of each iteration is accepted as the amount of error and is memorized. In the next iteration, the weights of the neurons are updated with the amount of error. This cycle ends when the specified number of iterations is reached or the amount of error is reduced to the desired level. In backpropagation networks, just like feed forward networks, the output value of their neurons depends on the input values of the previous neurons [52].

Backpropagation networks are a network structure where weights are updated by returning to the input layer. These types of neural networks have memory because they will update the previous weight value. The output value of neurons in this algorithm also depends on previous input values.

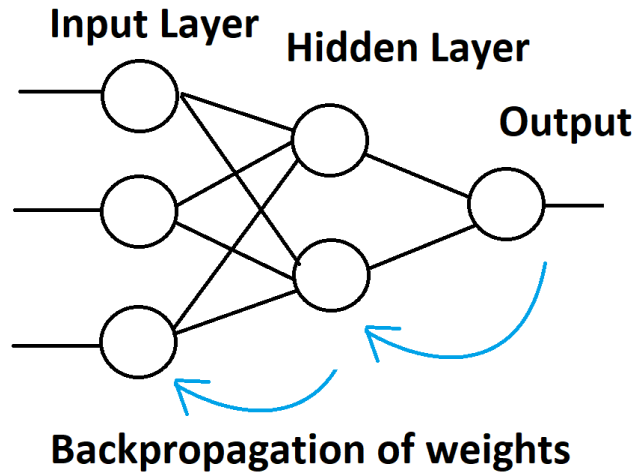


Figure 3.6. Backpropagation in an artificial neural network

Backpropagation is calculated as follows in n iterations for the neuron in the output layer.

$$e_j(n) = d_j(n) - y_j(n) \quad (3.11)$$

The error obtained by the neural network in each iteration during the learning process continues until it reaches the smallest value with the backward propagation algorithm.

$$E(n) = \frac{1}{2} \sum_{j \in C} e_j^2(n) \quad (3.12)$$

The average of all errors at the end of n iterations is shown above. In the relevant equation, C represents the number of neurons in the output layer of the network. Updating of the existing weights is done step by step as follows.

$$\Delta w_{ij} = -\eta \frac{\partial E(w)}{\partial w_{ij}}, \eta = \text{learning rate} \quad (3.13)$$

In the feedforward part, an output is obtained against the information generated by the input parameters. It needs to be updated as much as the difference between these outputs and the actual values. This equation is shown below.

$$\Delta w_{ij} = \eta \delta_j y_i \quad (3.14)$$

$$w_{n+1} = w_n + \Delta w_n \quad (3.15)$$

To update the weights in the hidden layer, as shown in 3.14, the total weights are derived from the weights in the hidden layer. The error change in between will be as much as shown in 3.15. In the next iteration, the current weight is updated by adding Δ . In this equation, η represents the learning rate. The biggest problem in the backpropagation algorithm is that the training time gradually increases depending on the data set and the learning rate [53].

3.3.3 Convolutional Neural Networks

To overcome the problems encountered in the field of computer vision, researchers have discovered Convolutional Neural Networks, inspired by human visual perception [54]. Convolutional Neural Networks can have one or more hidden layers and thus generate high-level features. These high-level attributes are called feature maps. Convolutional Neural Networks have weight coefficients capable of learning, as in classical neural networks [55]. Convolutional Neural Networks, which have emerged in recent years, have achieved great success, especially in problems with images. General Structure of Convolutional Neural Networks can be examined in Figure 3.7.

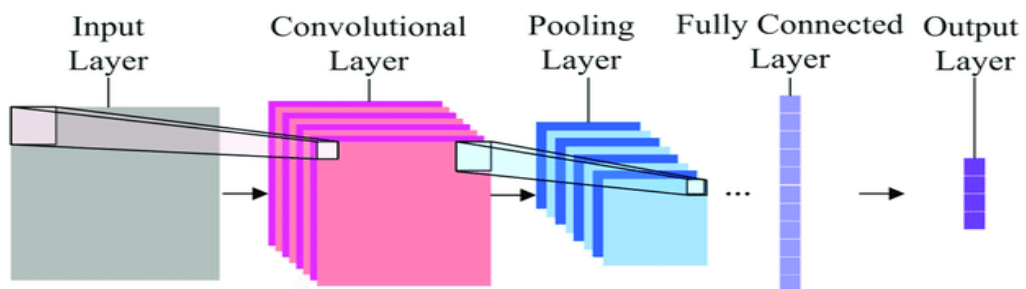


Figure 3.7. General Structure of Convolutional Neural Networks [56]

3.3.3.1 Convolutional Layer

A mathematical model was created based on the logic that the cells in the center of vision were divided into sub-regions in a way to cover the whole visual, and the simple cells have edge and corner-like features and complex cells concentrate on the whole visual with larger receptors. In this model, the convolution process can be thought of as a neuron's response to stimuli from its own half-field. In a normal neural network, in order to input an image to the network, it is necessary to turn this image into vector. Spatial relations between shapes are lost in the images that are vectorized. This may negatively affect the classification result in image-based problems. Convolutional layers designed with inspiration from the visual cortex of the brain and the spatial relationships of the images given as input do not disappear, thus obtaining positive results especially in solving problems whose input is images.

Dataset are fed into the first layer of CNN, the convolution layer. Whether the size of the input images is large or small is important in the performance of the model to be designed. While the use of large-sized input images can increase the success of the network, it can increase the training time due to the high memory requirement due to the large number of parameters to be used, and it can also extend the test time per image. If small images are used, the depth of the network will decrease, and the training time will be shortened and the memory need will be reduced. In this case, the performance of the network may be poor. Input image size should be chosen appropriately for network depth, hardware computing cost and network success [57].

The purpose of convolution is to extract images, video or sequential information from input data. Convolution works by learning image attributes and taking into account or preserving spatial relationships between pixels. If each image considered to be a pixel matrix, convolution reduces blocks of data to a smaller size than the given matrix. In CNN terminology, a 3x3 matrix can be called as "filter" or "feature detector". The matrix which is composed of moving the same filter over the image is called "Evolved Feature" or "Feature Map". This process is repeated until the input image is transformed

into a set of feature maps [58]. A figure of the feature map can be observed in the Figure 3.8.

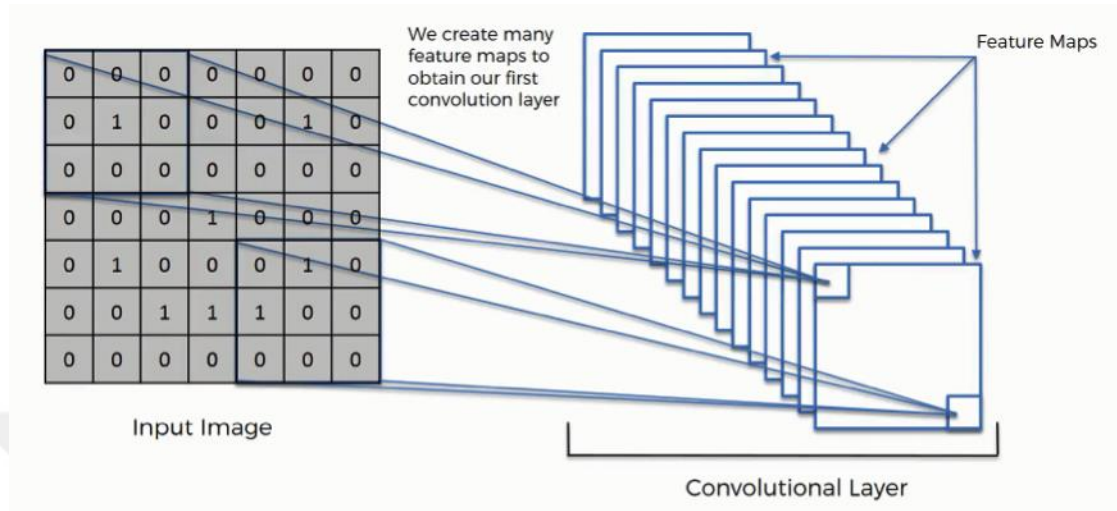


Figure 3.8. Feature map in the convolution layer [59]

In order to explain the functioning of convolution more clearly, this process will be explained in detail with an example. Let's assume that the pixel values of an image consist of only 0s and 1s, the image dimensions are 5×5 and we have a filter matrix of 3×3 dimensions. In the convolution process, the image input matrix X with dimensions $(5 \times 5 \times 1)$ (usually carrying the values from the previous layer), and the K filter matrix (also called kernel), which are usually smaller than the input $(3 \times 3 \times 1)$ and the output is a new matrix called Y. Figure 3.9 shows a sample input matrix and filter matrix.

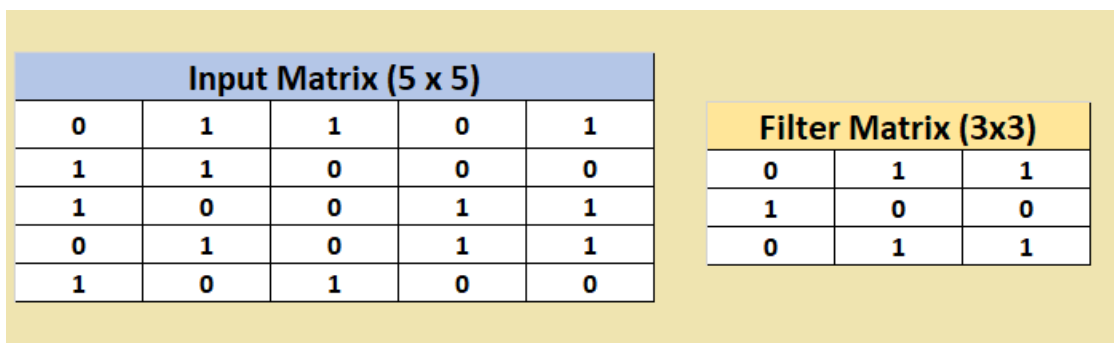
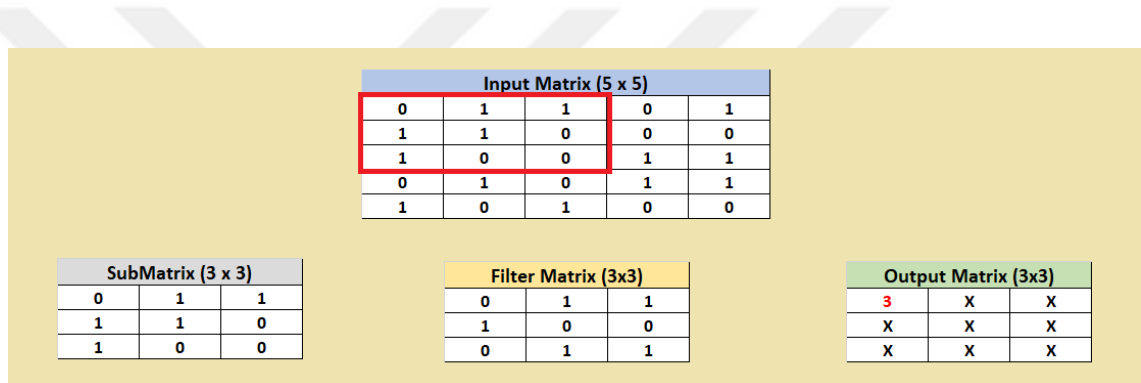


Figure 3.9. Input Matrix and Filter Matrix

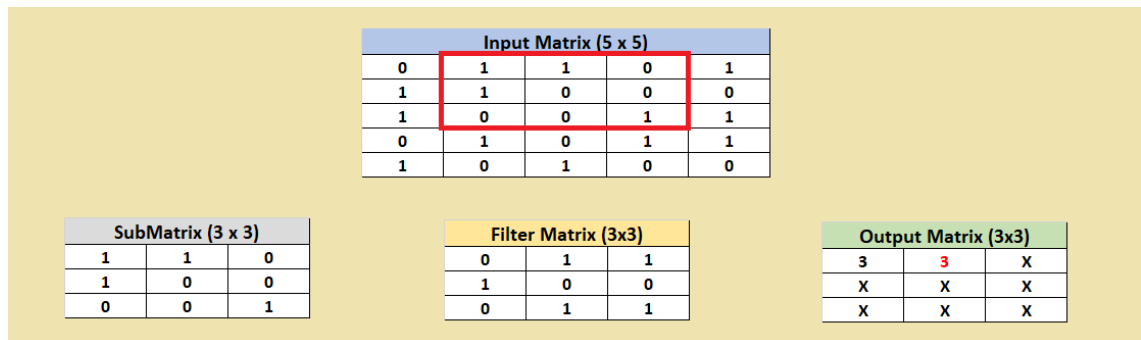
The first entry of matrix Y is calculated as follows:

- The first 3×3 sub-matrix of X is selected.
- Each entry is multiplied by its weight in the corresponding K matrix value.
- All outputs are summed and generated at the corresponding value in Matrix K.

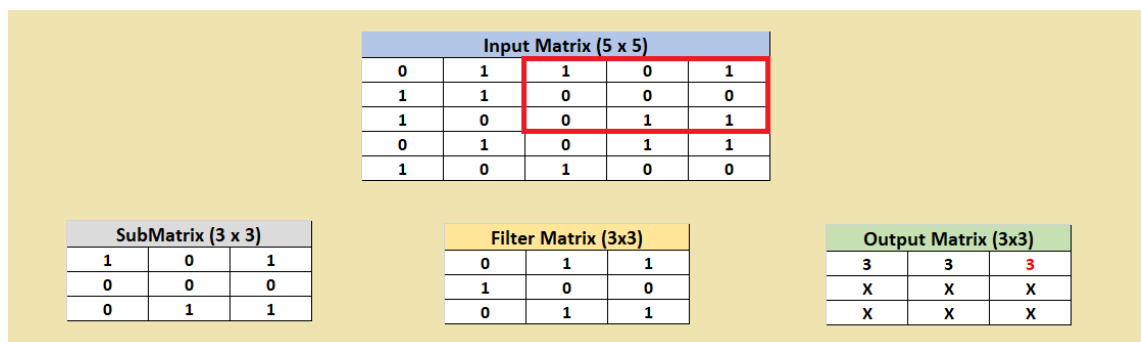
These last two steps are equal to the dot product of the sub-matrix of X and K, in fact as an integer value, is obtained from two multidimensional vectors. The matrix K is then shifted to the right to get the next element, and so the process is repeated with each submatrix of X. When there is not any element is right, new sub-matrix starts from the second row. Figure 3.10 explains the steps of matrix Y calculation in details.



First Step $\rightarrow (0 \times 0) + (1 \times 1) + (1 \times 1) + (1 \times 1) + (1 \times 0) + (0 \times 0) + (1 \times 0) + (0 \times 1) + (0 \times 1) = 3$



Second Step $\rightarrow (1 \times 0) + (1 \times 1) + (0 \times 1) + (1 \times 1) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (0 \times 1) + (1 \times 1) = 3$



Third Step $\rightarrow (1 \times 0) + (0 \times 1) + (1 \times 1) + (0 \times 1) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (1 \times 1) + (1 \times 1) = 3$

| Input Matrix (5 x 5) | | | | |
|----------------------|---|---|---|---|
| 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 |

| SubMatrix (3 x 3) | | |
|-------------------|---|---|
| 1 | 1 | 0 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |

| Filter Matrix (3x3) | | |
|---------------------|---|---|
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 1 |

| Output Matrix (3x3) | | |
|---------------------|---|---|
| 3 | 3 | 3 |
| 3 | X | X |
| X | X | X |

Fourth Step $\rightarrow (1 \times 0) + (1 \times 1) + (0 \times 1) + (1 \times 1) + (0 \times 0) + (0 \times 0) + (0 \times 0) + (1 \times 1) + (0 \times 1) = 3$

| Input Matrix (5 x 5) | | | | |
|----------------------|---|---|---|---|
| 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 |

| SubMatrix (3 x 3) | | |
|-------------------|---|---|
| 1 | 0 | 0 |
| 0 | 0 | 1 |
| 1 | 0 | 1 |

| Filter Matrix (3x3) | | |
|---------------------|---|---|
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 1 |

| Output Matrix (3x3) | | |
|---------------------|---|---|
| 3 | 3 | 3 |
| 3 | 1 | X |
| X | X | X |

Fifth Step $\rightarrow (1 \times 0) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (0 \times 0) + (1 \times 0) + (1 \times 0) + (0 \times 1) + (1 \times 1) = 1$

| Input Matrix (5 x 5) | | | | |
|----------------------|---|---|---|---|
| 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 |

| SubMatrix (3 x 3) | | |
|-------------------|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 0 | 1 | 1 |

| Filter Matrix (3x3) | | |
|---------------------|---|---|
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 1 |

| Output Matrix (3x3) | | |
|---------------------|---|---|
| 3 | 3 | 3 |
| 3 | 1 | 2 |
| X | X | X |

Sixth Step $\rightarrow (0 \times 0) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (1 \times 0) + (1 \times 0) + (0 \times 0) + (1 \times 1) + (1 \times 1) = 2$

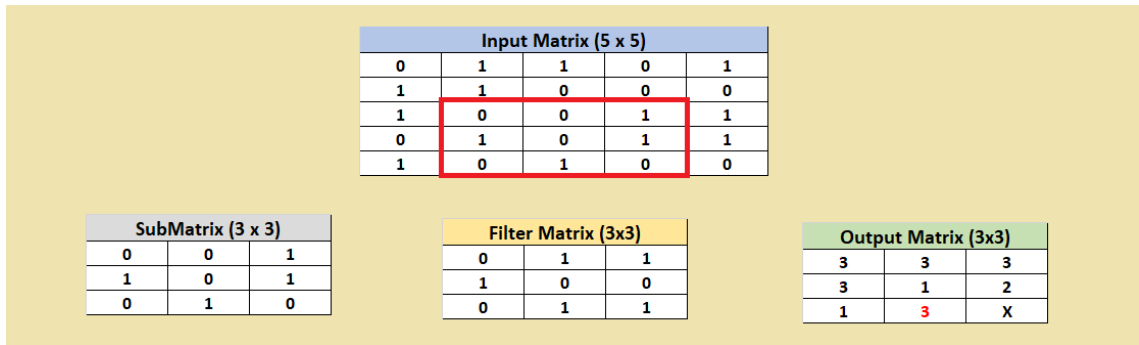
| Input Matrix (5 x 5) | | | | |
|----------------------|---|---|---|---|
| 0 | 1 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 0 |

| SubMatrix (3 x 3) | | |
|-------------------|---|---|
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |

| Filter Matrix (3x3) | | |
|---------------------|---|---|
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 1 |

| Output Matrix (3x3) | | |
|---------------------|---|---|
| 3 | 3 | 3 |
| 3 | 1 | 2 |
| 1 | X | X |

Seventh Step $\rightarrow (1 \times 0) + (0 \times 1) + (0 \times 1) + (0 \times 1) + (1 \times 0) + (0 \times 0) + (1 \times 0) + (0 \times 1) + (1 \times 1) = 1$



Eighth Step $\rightarrow (0 \times 0) + (0 \times 1) + (1 \times 1) + (1 \times 1) + (0 \times 0) + (1 \times 0) + (0 \times 0) + (1 \times 1) + (0 \times 1) = 3$

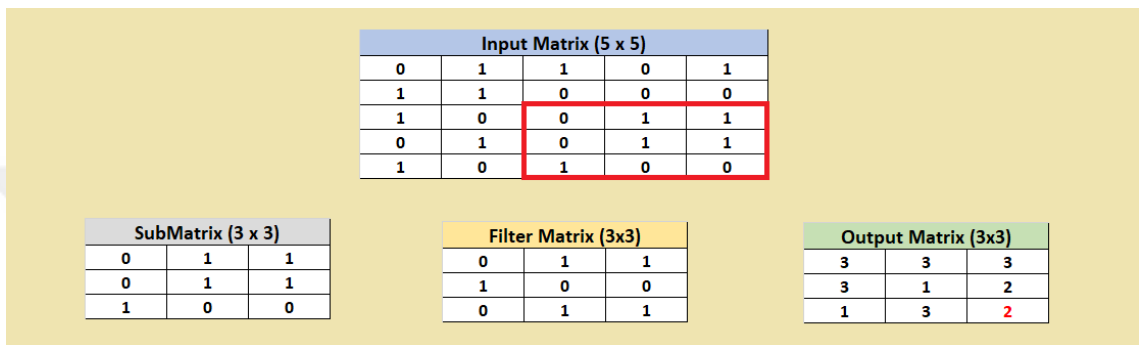


Figure 3.10. Steps of matrix Y calculation

Ninth Step $\rightarrow (0 \times 0) + (1 \times 1) + (1 \times 1) + (0 \times 1) + (1 \times 0) + (1 \times 0) + (1 \times 0) + (0 \times 1) + (0 \times 1) = 2$

After multiplying the image matrix by the filter matrix, the result of the convolution process is the following matrix (Y) as can be seen in Figure 3.11.

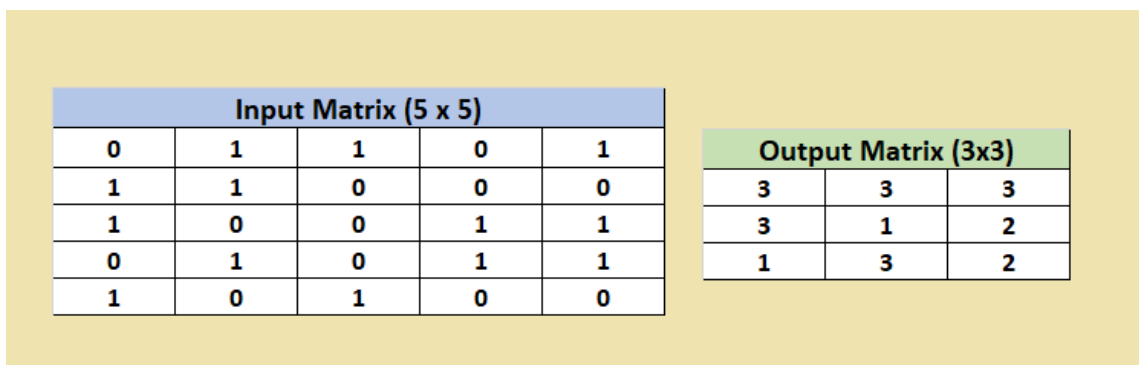


Figure 3.11. Output Matrix after feature detection

During the training process, CNN learns the principles of these filters on its own. When more filters are used, more object attributes can be extracted and model performance

will be improved by identifying hidden patterns in images. Depth, zero-padding and stride parameters determine the convolution attribute. These parameters must be determined before the convolution step takes place.

The number of filters used for the convolution process is called the depth. Depth is used for visual recognition and it translates to the third dimension or corresponds to the different color channels of an image. Zero-padding fills the periphery of the input matrix with zeros, feature detector can be applied to the boundary values of the input image matrix. Stride length as shown in Figure 3.12 is determined how many pixels the filter matrix will jump to after one step on the input matrix. If stride length or stride is 1, filters move only one pixel per step. When the step length is 2, the filters skip 2 pixels at a time.

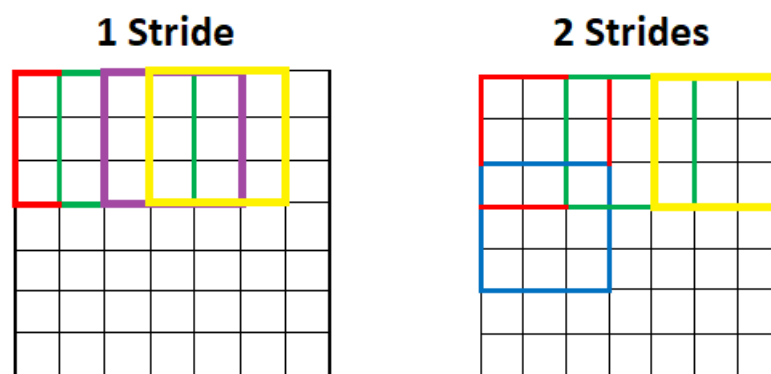


Figure 3.12. Stride Lengths example

3.3.3.2 Activation Functions and ReLU

The simplest known form of activation functions are linear activation functions with no change applied. Although training a network with these activation functions is fairly easy, they cannot discover the complex features in the data. Unfortunately, nonlinear activation functions such as sigmoid and tanh can better identify complex relationships in data. However, these functions have disadvantages such as not being able to update the weights since their derivatives towards the advancing layers of the network are zero. For this reason, the activation function that is frequently preferred in convolutional

neural networks is the corrected linear unit function, known as Rectified Linear Unit (ReLU). When the function gets a negative value, it produces a zero result, and when it gets a positive value, it returns the received value. The formula of ReLU is as follows with graphic in Figure 3.13.

$$f(x) = \max(0, x) \quad (3.16)$$

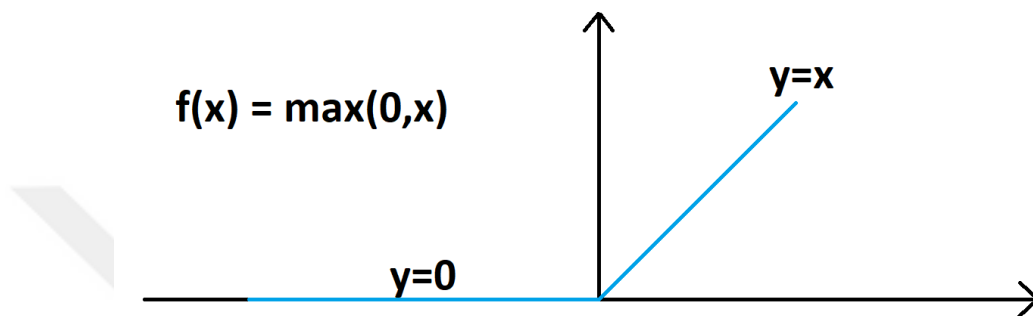


Figure 3.13. Rectified Linear Unit (ReLU)

This activation function was first introduced into a dynamic network in 2000 for strong biological motivations and mathematical justifications. ReLU is the stage after the convolution process. It applies to each element individually and replaces all negative pixel values in the attribute map to zero. The purpose of ReLU is to include nonlinearity in training.

3.3.3.3 Pooling Layer

When the input image is too large, the number of trainable parameters and the computational load are also too large. The main purpose of the pooling layer is to reduce the width and height by keeping the depth of the input data constant for the next layer. The reduction in size as a result of the pooling layer causes information loss and prevents the system from memorizing [60]. In this layer, certain filters are defined and the process is carried out by taking the values. There are two different types are used, maximum pooling and average pooling, in the industry. The maximum pooling type is

preferred as it generally performs well. Pooling process is performed for all images as many filters are formed as a result of the previous layer [57].

Max pooling is created by taking the highest, which the maximum value, in each specified part of the attribute map [61].

$$S_i = \max a_i (i \in R_j) \quad (3.17)$$

In 3.17 equation, S_i is output matrix, a_i is feature map size, R_j is the unit area where the pooling process is executed in the feature map. Figure 3.14 shows the max pooling process for a 2×2 filter, stride length = 2 and depth = 0.

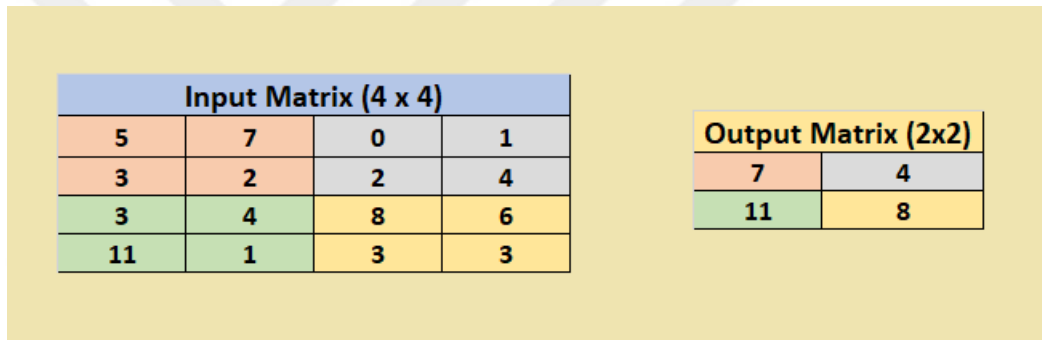


Figure 3.14. Max Pooling Process in CNN

$$S_1 = \max (5,7,3,2) = 7$$

$$S_2 = \max (0,1,2,4) = 4$$

$$S_3 = \max (3,4,11,1) = 11$$

$$S_4 = \max (8,6,3,3) = 8$$

Average Pooling process is performed by calculating the average value of each part of the feature map, that is, the part of the image covered by the core [61].

$$S_i = 1/ R_j \sum_{i \in R_j} a_i \quad (3.18)$$

In 3.18 equation, S_i is output matrix, a_i is feature map size, R_j is the unit area where the pooling process is executed in the feature map. Figure 3.15 shows the average pooling process for a 2×2 filter, stride length = 2 and depth = 0.

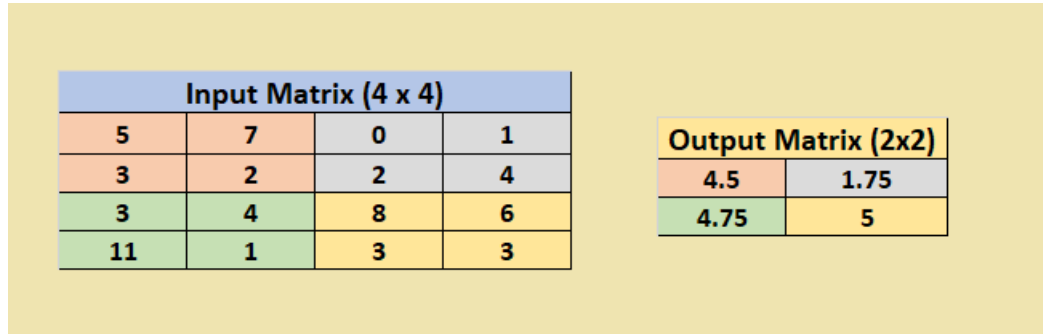


Figure 3.15. Average Pooling Process in CNN

$$S_1 = (1 / 16) \times 18 \times 4 = 4.5$$

$$S_2 = (1 / 16) \times 7 \times 4 = 1.75$$

$$S_3 = (1 / 16) \times 19 \times 4 = 4.75$$

$$S_4 = (1 / 16) \times 20 \times 4 = 5$$

3.3.3.4 Fully Connected Layers

This step is the last phase of the CNN. The fully connected layers is very similar to the way a conventional neural network arranges neurons. The main disadvantage of a fully connected layers is that training examples require various parameters that require complex computation. However, in conventional CNN models, the fully connected (FC) layer is widely used [23].

The flattened output of the convolutional layer or final merging is transmitted as input to the fully connected layer. Flattened means that the concatenation and the final (or any) output of the convolutional neural network, i.e. a 3-dimensional matrix, will be transformed into a vector. Figure 3.16 shows flattening in CNN.

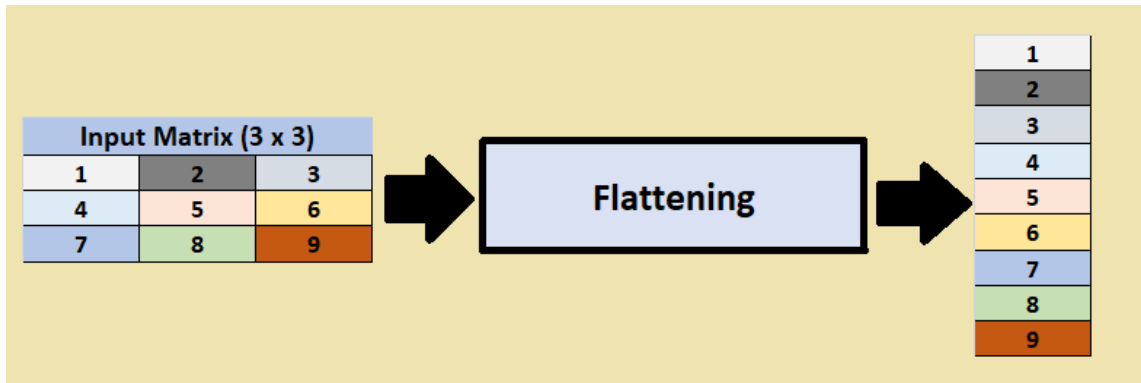


Figure 3.16. Flattening in CNN

The fully connected layer is a conventional multi-layer layer and uses a softmax or sigmoid activation function in the output layer. Thus, after the output matrix has been obtained and flattened, the activation function is used to estimate the output classes. Activation function will compress the outputs of each group between 0 and 1. Ideally, the activation function is used at the classifier output layer where each input class is attempted to be determined. After the output matrix is obtained and this matrix is flattened, the activation function is the final stage of the fully connected layer in convolutional neural network application. An example structure for fully connected layer can be examined in Figure 3.17.

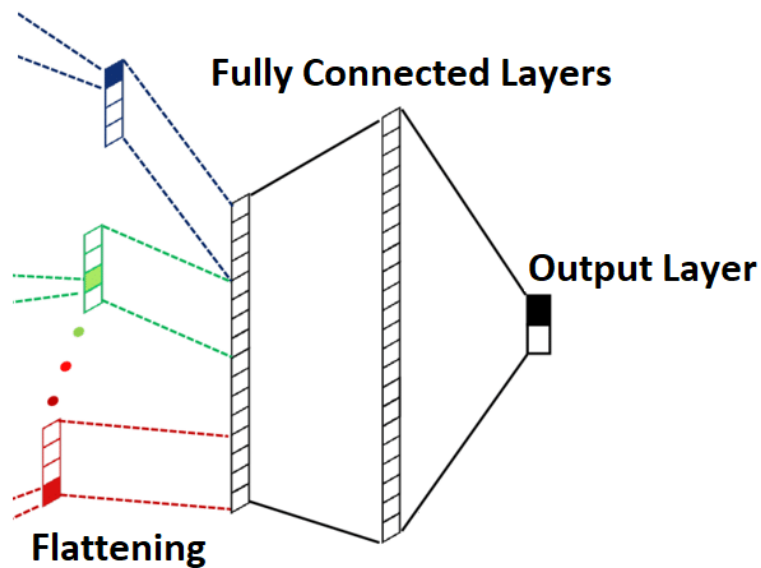


Figure 3.17. Fully connected layers

3.3.3.5 Model Training and Loss Function

The result class probabilities are calculated from the original images through the convolutional neural networks layers. In convolutional layers, 3-dimensional layer outputs are obtained from layer inputs with 3-dimensional volume. Convolutional and fully connected layers have parameters. The parameters of neurons in convolutional and fully connected layers are updated with the gradient descent algorithm. Thus, the class scores of the convolutional network can be calculated according to the tags in the training set for each image. The parameters of the convolutional layer in CNN are learnable parameters [55].

CNN architecture scores of the correct classes can be kept in the training set at maximum and the scores of the wrong classes can be kept at the minimum. This task is accomplished by improving the weight coefficients (w_i). Since the correct tags of training examples are known while training the network, the difference between the tags estimated by CNN (with the current weight coefficients) and the actual class tags defined in the training set is calculated with the loss function. Thus, the weight coefficients to minimize the average loss value of the whole training set are found. The gradient descent algorithm is an iterative algorithm, and the updating of the weight coefficients is performed as specified in below.

$$w_{ij}^{t+1} = w_{ij}^t - \alpha (dL/ dw_{ij}) \quad (3.19)$$

w_{ij} is the weight coefficient of the j neuron in layer i , t is the iteration step, α is the learning rate and L is the loss function. Gradient descent algorithm aims to find the best weight coefficients to minimize the loss function. Updating the parameters in each iteration is performed by subtracting the derivative of the loss function by a small number (learning rate) from the value of the current weight coefficient. The learning rate parameter controls the learning speed between steps and determines best parameters in the model. Usually small values are being chosen for this parameter. Otherwise, instead of reaching the minimum point of the loss function (convergence), the possibility of divergence arises [62].

3.3.3.6 Preventing Overfitting in Convolutional Neural Networks

One of the biggest problems of deep networks is overfitting. The problem of overfitting is that the convolutional neural network that achieves high accuracy rates during training gives low accuracy during testing, in short, the network memorizes the data. This situation usually arises in problems where there is not enough data for training. Studies are continuing to achieve high accuracy with low amount of data. It is a difficult and laborious process to obtain large-scale data, especially in medical fields. Therefore, various data replication methods are needed and used in deep learning. Dropout method and data augmentation is summarized as follows.

Dropout method is one of the methods of avoiding overfitting. This method can be examined in Figure 3.18 is the exclusion of some random neurons at a previously determined rate in the weight update process during the re-propagation process during training.

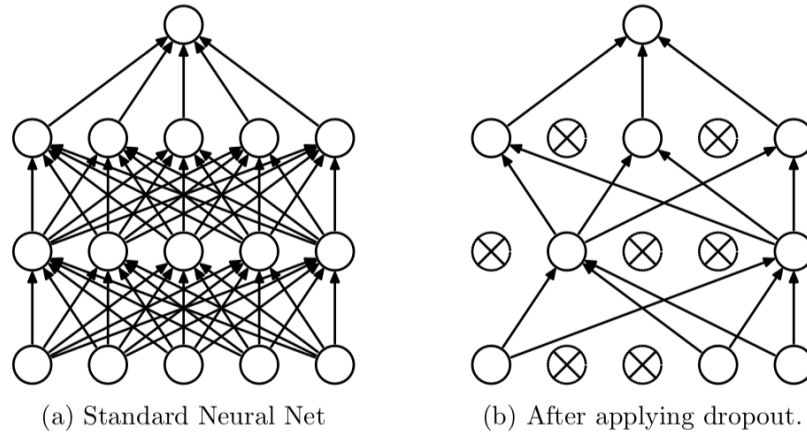


Figure 3.18. Dropout Function [63]

Data Augmentation is a method that achieves very successful results, especially in neural networks with images. With this method, the data set used for training is expanded by producing new images with various image processing methods such as rotation, cutting, cropping and contrast enhancement. Data-specific replication methods can be developed, as well as there are ready-made libraries that combine these methods. One of these libraries is Keras ImageDataGenerator as being one of the most popular

libraries due to its ease of use and advantageous features. The content and usage details of the library are available on the website. Figure 3.19 shows an example of data augmentation function.



Figure 3.19. Data Augmentation Function [64]

4. CONVOLUTIONAL NEURAL NETWORKS MODELS AND RESULTS

This thesis study is based on as a research on ECG classification. It includes data collection and analysis, preprocessing, model training and classification sections.

In the thesis study, the data in the MIT-BIH Arrhythmia Database are divided into test and training as in the Mondéjar-Guerra et al (2019) used in SVM classification. ECG signal images and phase space images from the data set is used also these images are repurposed to generate the training set. In addition, the dataset was converted into a version which includes seventy-five percent of the training set and twenty-five percent of the test set and the model is trained. Four different models are trained in this thesis.

Models are trained with computer hardware specifications as below:

- Intel® Core™ i7-7700HQ @ 2.80 GHz CPU
- NVIDIA Geforce GTX 1050 4 GB GPU
- Ramaxel 16 GB RAM

4.1 DATABASE

Database contains two-channel half-hour 48 extracts of ambulatory ECG recordings taken from 47 people by the BIH Arrhythmia Laboratory between 1975 and 1979. Each of the 48 recordings is nearly over 30 minutes. In 46 of the 48 recordings, the measurement of the subjects was made with the MLII signal obtained by placing the electrodes on the subject's chest. Record 102 and record 104, could not measure with MLII signal, they measured with V5 signal. Due to this reason, these two records are excluded from the database in this thesis. Also, the records with paced beats were not considered, which are record 107 and record 217. In order to increase the success rate of the classification, the Q wave which is connected to the highest R wave, was drawn with a single line in the phase space images. The subsections of this chapter discuss these steps in more detail and conveys an in-depth view of the enhancements.

After all the procedures, 11 subjects were found to be healthy. 33 subjects had an unhealthy heart. For healthy records, ECG line diagrams are shown in Figure 4.1 and ECG phase space diagram is shown in Figure 4.2. For unhealthy records, ECG line diagrams are shown in Figure 4.3 and ECG phase space diagram is shown in Figure 4.4.

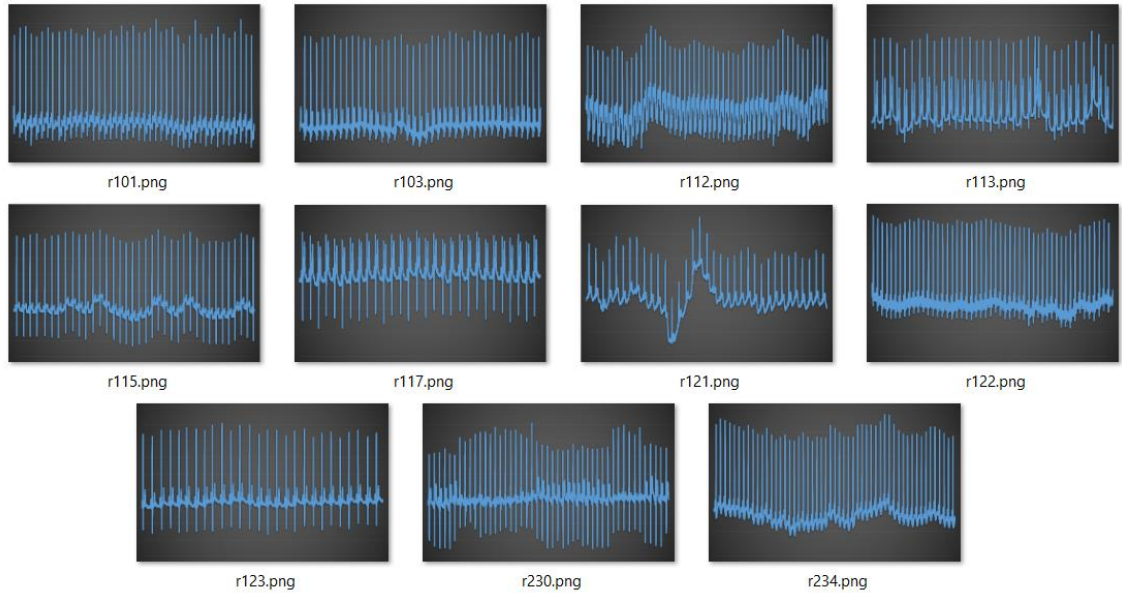


Figure 4.1. ECG Line Diagram of Healthy Records

Healthy Records, ECG Lines: 101, 103, 112, 113, 115, 117, 121, 122, 123, 230 and 234.

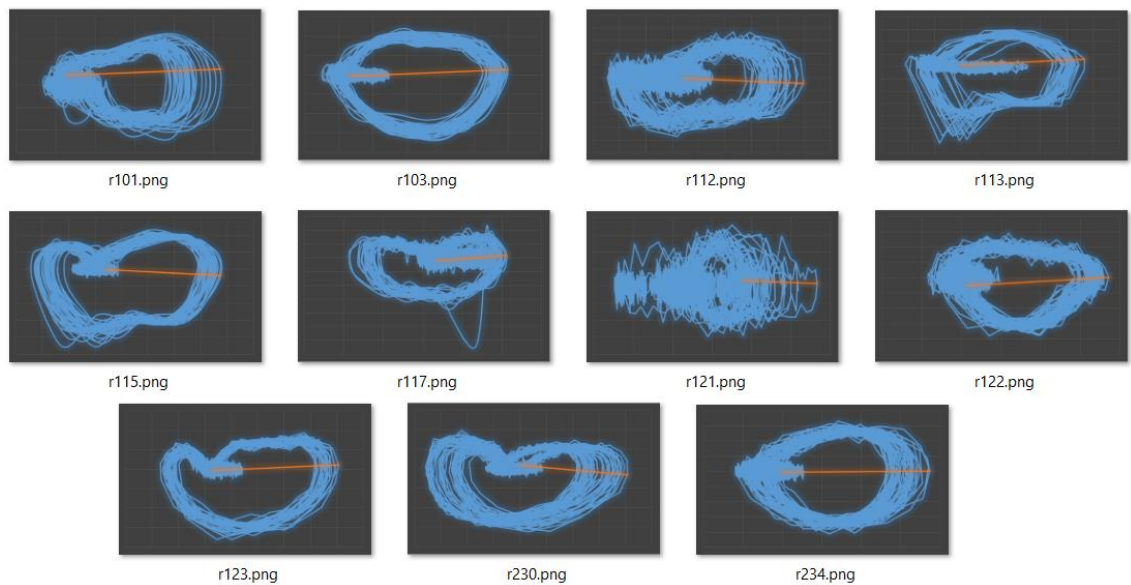


Figure 4.2. Phase Space Diagram of Healthy Records

Healthy Records, Phase Space Diagram: 101, 103, 112, 113, 115, 117, 121, 122, 123, 230 and 234.

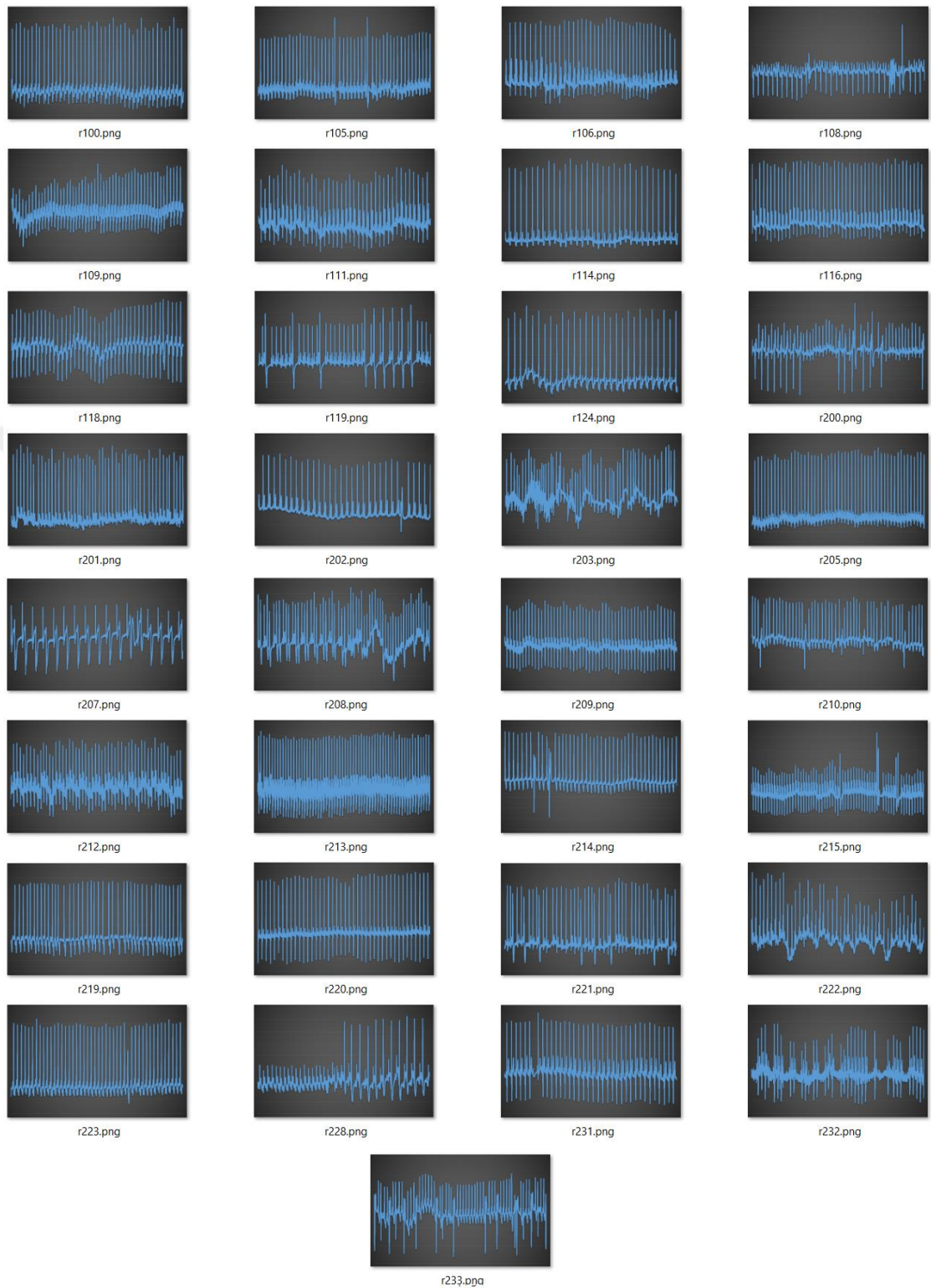


Figure 4.3. ECG Lines of Unhealthy Records

Unhealthy Records, ECG lines: 100, 105, 106, 108, 109, 111, 114, 116, 118, 119, 124, 200, 201, 202, 203, 205, 207, 208, 209, 210, 212, 213, 214, 215, 219, 220, 221, 222, 223, 228, 231, 232 and 233.

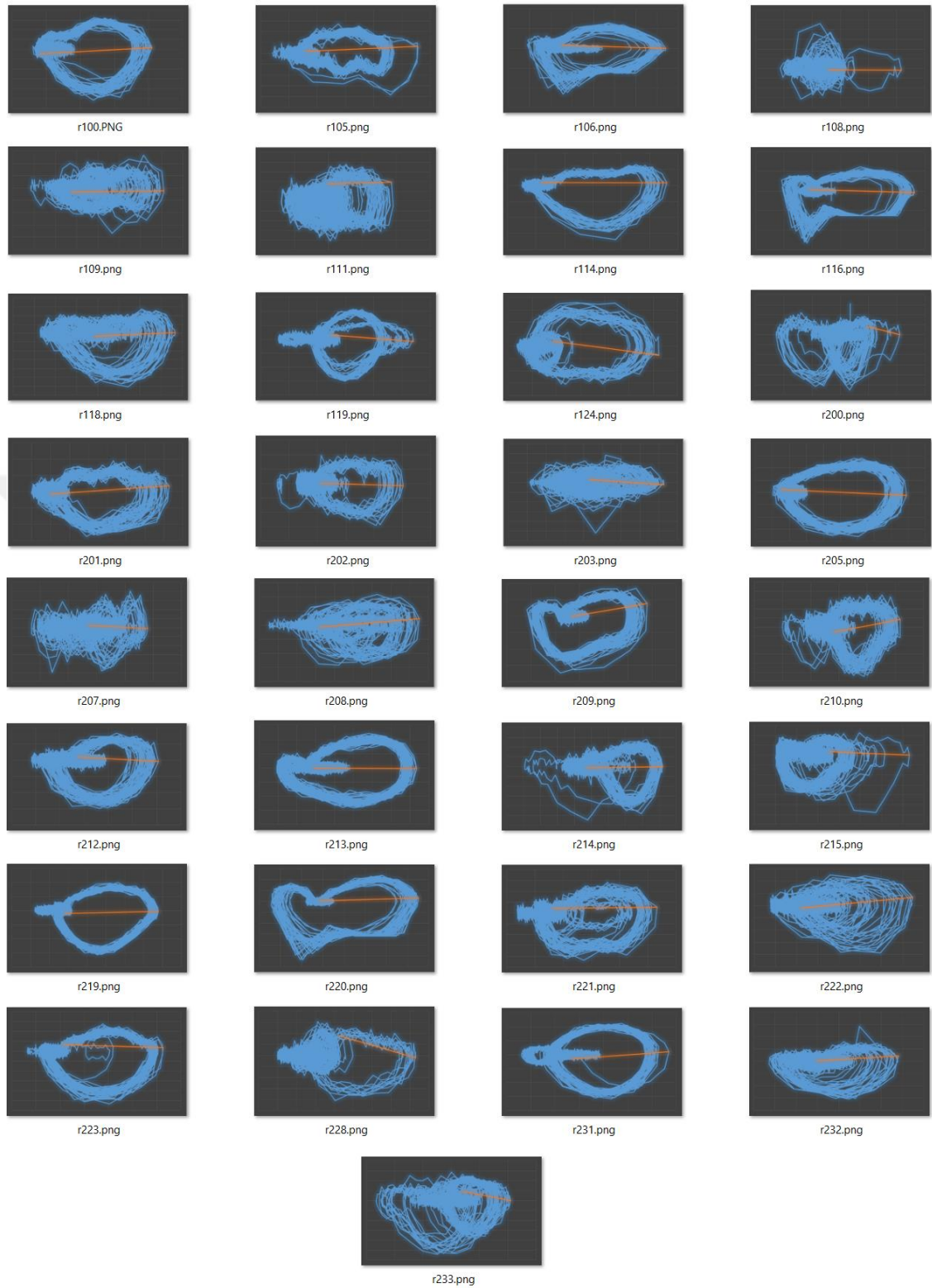


Figure 4.4. Phase Space Diagram of Unhealthy Records

Unhealthy Records, Phase Space Diagram: 100, 105, 106, 108, 109, 111, 114, 116, 118, 119, 124, 200, 201, 202, 203, 205, 207, 208, 209, 210, 212, 213, 214, 215, 219, 220, 221, 222, 223, 228, 231, 232 and

233

4.2 MODEL TRAINING

Keras and Tensorflow libraries are used in Python in order to implement the CNN model. With input shape of 64x64x3 feature vectors were fed to the convolution layers followed by pooling layer where the padding features were kept same as input to the pooling layer. In order to minimize the size of the output vector gradually, dense layers are used. Unit value of dense layer specified with one neuron due developing a binary CNN model purpose.

There are only 44 records to be used in the dataset. In a usual convolutional neural network model, 1000 or more records are recommended to use in the dataset. Due to this issue, data augmentation method is used to create more pictures. With Keras library; shear range, zoom range and horizontal flip parameters are adjusted. Shear range is used for shear intensity (shear angle in counter-clockwise direction in degrees), zoom range is used for zooming the parts of the images, horizontal flip is used for flipping the images horizontally.

In this convolutional neural network model, two convolutional layers are used related with ReLU activation function. Two pooling layers with using max pooling function after ReLU activation function. Flattening process is executed in order to process fully connected layers section. Afterwards, the CNN model is transformed a normal neural network. 128 neurons are used in hidden layer and one neuron is used for the output layer, due to binary CNN classification. Figure 4.5 shows the steps in the architecture of CNN model.

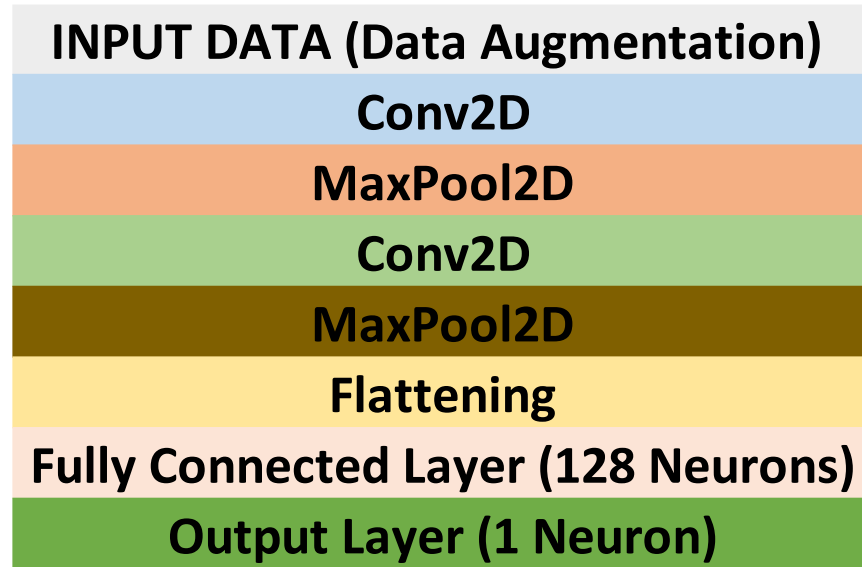


Figure 4.5. Architecture of CNN model

In two models, which training set and test set is the same with the study by Mondéjar-Guerra, 80 epochs are used to train the model. In altered training set and test set, 175 epochs are used to train the model. This is the only difference between training the models.

4.3 CNN MODELS WITH SAMPLE PROJECT

In this thesis, two different CNN models are trained respective of ECG lines. One model is trained with the same test and training dataset of the study of Mondéjar-Guerra et al, (2019). Other model is trained with a dataset which has seventy-five percent training set and twenty-five percent test set ratio. The subsections of this chapter discuss these steps in more detail and conveys an in-depth view of the enhancements.

In MIT-BIH Arrhythmia Database records, ECG data is plotted in order to train the model. Training set and testing set are identified and created to train the CNN. Table 4.1 summarizes the test record data for healthy and unhealthy data which used in training set and testing set. Figure 4.1 and 4.3 can be examined for the images using the data sets.

Training set and testing set includes equally 22 records. Training and testing set ratio is equally composed with fifty percent. Healthy testing set includes different 6 records and training set includes different 5 records. There are 54% healthy records exist in test and 46% healthy records exist in training set. Disease test set includes 16 disease records and training set includes 17 disease records. There are 48% disease records exist in test and 52% records exist in training set.

Table 4.1. Training Set and Test Set

| SET | Training Set | Testing Set |
|-------------------|--------------|-------------|
| Healthy records | 101 | 103 |
| | 112 | 113 |
| | 115 | 117 |
| | 122 | 121 |
| | 230 | 123 |
| | | 234 |
| Unhealthy records | 106 | 100 |
| | 108 | 105 |
| | 109 | 111 |
| | 114 | 200 |
| | 116 | 202 |
| | 118 | 210 |
| | 119 | 212 |
| | 124 | 213 |
| | 201 | 214 |
| | 203 | 219 |
| | 205 | 221 |
| | 207 | 222 |
| | 208 | 228 |
| | 209 | 231 |
| | 215 | 232 |
| | 220 | 233 |
| | 223 | |

4.3.1 ECG Model & Results in Sample Project Dataset

After the model is trained with this specific properties, it had 95.45% success rate in training set and 63.63% in testing set as seen in Figure 4.6. When the epochs performance is analyzed, after 30th epoch training accuracy gets higher but test accuracy gets lower. The model can even classify the training model successfully, it classified

new ECG records unsuccessfully. With this training and test set rate, model had an overfitting problem.

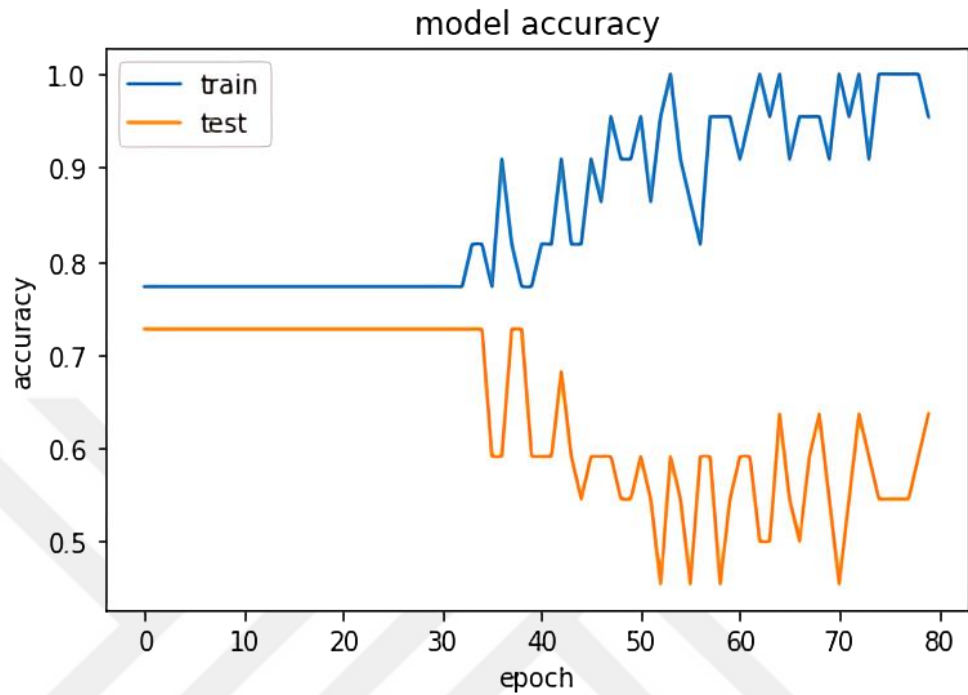


Figure 4.6. ECG lines accuracy comparison of training set and testing set in sample project dataset

Figure 4.7 shows model loss comparison between sets. It can easily be seen that the training and testing data is related and parallel with other. Moreover, it also proves that when the model learning, it gets better to classify test data. Especially, after 30th epoch training loss gets lower but test loss gets higher. This means that model can train itself in order to classify training dataset, but it does not train itself for test dataset. Result of that, model accuracy gets better in training set but not in testing set. This situation indicates an overfitting problem with model training.

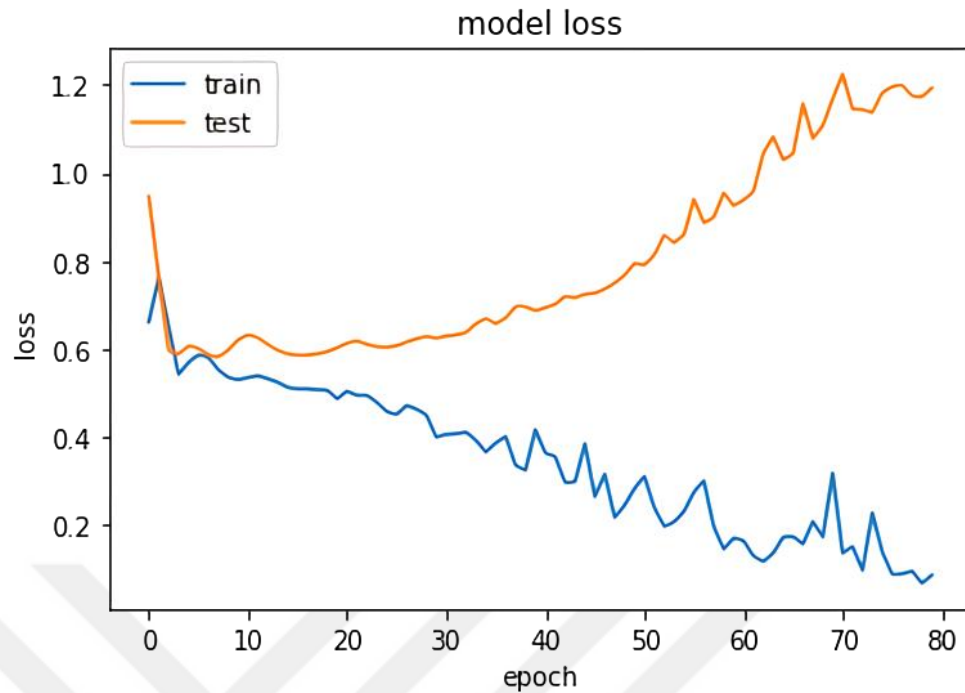


Figure 4.7. ECG lines model loss comparison of training set and testing set in sample project dataset

There is a considerable accuracy result difference in training accuracy and validation(test) accuracy in the results. Among of 22 test records, model can classify 14 of them correctly and 8 of them with fault. Healthy test set accuracy is %16.7. Among of the 6 healthy records, model can classify only one record, r103, correctly. For r113, r117, r121, r123 and r234 model classify as a disease record. Disease test set accuracy is %81.25. Among of 16 disease records, model can classify 13 records; r100, r105, r200, r202, r210, r212, r213, r214, r222, r228, r231, r232 and r233 correctly. Model failed to classify r111, r219 and r221 by classifying these records as healthy records.

| Outputs of ECG Results | | | | | | | |
|------------------------|---------|----------|---------|--------------|---------|---------|---------|
| Epoch Count | | Accuracy | | Val Accuracy | | | |
| 80 | | 95.45% | | 63.63% | | | |
| Healthy | | | | | | | |
| r103 | r113 | r117 | r121 | r123 | r234 | | |
| Healthy | Disease | Disease | Disease | Disease | Disease | | |
| Disease | | | | | | | |
| r100 | r105 | r111 | r200 | r202 | r210 | r212 | r213 |
| Disease | Disease | Healthy | Disease | Disease | Disease | Disease | Disease |
| r214 | r219 | r221 | r222 | r228 | r231 | r232 | r233 |
| Disease | Healthy | Healthy | Disease | Disease | Disease | Disease | Disease |

Figure 4.8. ECG lines model outputs of training set and testing set in sample project dataset

In order to avoid overfitting problem, image augmentation is used to train this model. Although this feature is used, overfitting problem still occurred in the result. This issue is caused by the ratio of the testing set and training set in the dataset. In a CNN model, training ratio would be %75 and %25 or %80 and %20 to the all dataset. However, with these training and testing set ratio, data set is not enough training data for extract features.

Especially, it can be observed that there is a very low accuracy which is %16.7 in healthy test set. Model can only classify one record correctly. There are six test records in healthy test set and five records in healthy training set. Model is being trained with five different healthy records to classify six different healthy records. There are more records in the test set than the training set. Also the number of records in the training set is significantly low for using CNN. Due to this situation, it cannot learn the healthy data in the dataset and cannot classify correctly.

On the other hand, there are 17 records in the training set and 16 records in test set for disease records. Although test set and training set total records are similar, there are more records in training set then the test set. Also, 17 records can be considerably higher than only 5 records in the healthy training set. With 17 records, model can train

itself and extract the features to detect disease records. Model has only three faulty classification result of 16 test records in the disease test set.

4.3.2 Phase Space Model & Results in Sample Project Dataset

After the model is trained with this specific properties, it had 95.45% success rate in training set and 72.72% in test set. When the epochs performance is analyzed, after 30th epoch training accuracy gets higher but test accuracy not changes and it is around 65 – 75%. Even the model can classify the training model successfully, it classified test records unsuccessfully. With this training and test set rate, model had an overfitting problem even with the phase space images. Figure 4.9 compares the accuracy of training and testing set.

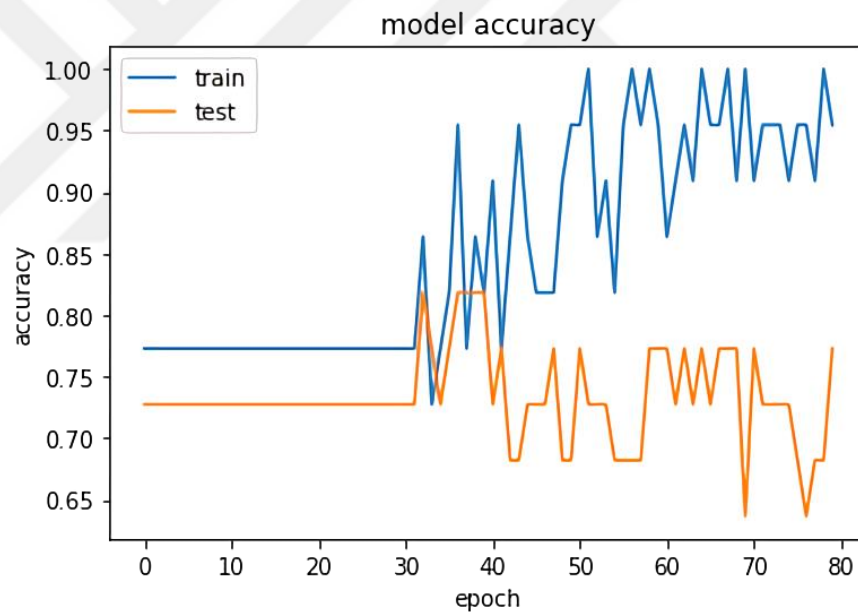


Figure 4.9. Phase space model accuracy comparison of training set and testing set in sample project dataset

Figure 4.10 show that the train and test data is not related and not parallel with other. After 30th epoch training loss gets lower but test loss gets higher and the difference is starts to increase. This means that model can train itself in order to classify training dataset, but it does not train itself for test dataset. Result of that, model accuracy gets better in training set but not in test set. This situation shows there is an overfitting problem in this model training.

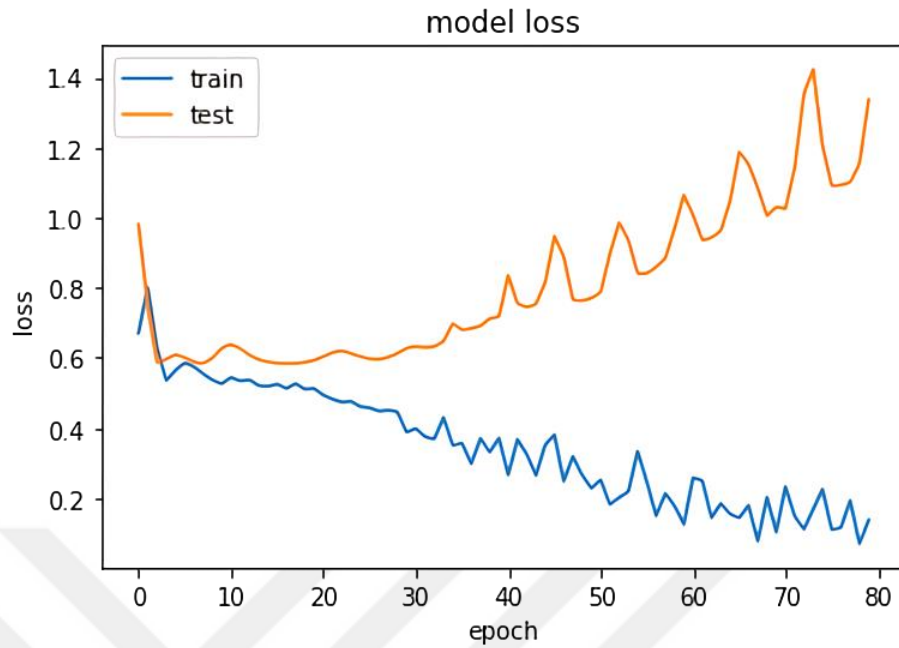


Figure 4.10. Phase Space model loss comparison of training set and testing set in sample project dataset

There is an immense accuracy result difference training accuracy and validation(test) accuracy in the results. Among of 22 test records, model can classify 16 of them correctly and 6 of them faulty. Healthy test set accuracy is 16.7%. Among of the 6 healthy records, model can classify only one record, r234, correctly. For r103, r113, r117, r121 and r123 model classify as a disease record. Disease test set accuracy is 93.75%. Among of 16 disease records, model can classify 15 records; r105, r111, r200, r202, r210, r212, r213, r214, r219, r221, r222, r228, r231, r232 and r233 correctly. Model failed to classify r100 by classifying these records as healthy record.

| Outputs Phase Space Images Results | | | | | | | |
|------------------------------------|----------|--------------|---------|---------|---------|---------|---------|
| Epoch Count | Accuracy | Val Accuracy | | | | | |
| 80 | 95.45% | 72.72% | | | | | |
| Healthy | | | | | | | |
| r103 | r113 | r117 | r121 | r123 | r234 | | |
| Disease | Disease | Disease | Disease | Disease | Healthy | | |
| Disease | | | | | | | |
| r100 | r105 | r111 | r200 | r202 | r210 | r212 | r213 |
| Healthy | Disease | Disease | Disease | Disease | Disease | Disease | Disease |
| r214 | r219 | r221 | r222 | r228 | r231 | r232 | r233 |
| Disease | Disease | Disease | Disease | Disease | Disease | Disease | Disease |

Figure 4.11. Phase Space model outputs of training set and testing set in sample project dataset

In order to avoid overfitting problem image augmentation is used to training this model. But even this feature is used, overfitting problem occurred in the result. This issue is caused by the ratio of the testing set and training set in the dataset. In a CNN model, training ratio would be %75 and %25 or %80 and %20 to the all dataset. But with these training and test set ratio, there are not enough training data to extract features.

Especially, it can be observed that there is a very low accuracy (%16.7) in healthy test set. Model can only classify one record correctly. There are six test records in healthy test set and five records in healthy training set. Model is being trained with five different healthy records to classify six different healthy records. There are more records in the test set than the training set. Also the number of records in the training set is significantly low for using CNN. Due to this situation, it cannot learn the healthy data in the dataset and cannot classify correctly.

On the other hand, there are 17 records in the training set and 16 records in test set for disease records. Although test set and training set total records are similar, there are more records in training set then the test set. Also, 17 records can be considerably higher than only 5 records in the healthy training set. With 17 records, model can train itself and extract the features to detect disease records. Model has only one faulty classification result of 16 test records in the disease test set.

4.4 CNN MODELS WITH ALTERED TRAINING AND TEST SET

Training set and testing set are altered to create and train the CNN. Table 4.2 summarizes the test record data for healthy and unhealthy data. Figure 4.1 to Figure 4.4 can be examined for the images using the data sets.

Table 4.2 Altered Training Set and Test Set

| SET | Training Set | Testing Set |
|-------------------|--------------|-------------|
| Healthy records | 101 | 103 |
| | 113 | 112 |
| | 115 | 234 |
| | 117 | |
| | 121 | |
| | 122 | |
| | 123 | |
| | 230 | |
| Unhealthy records | 106 | 100 |
| | 108 | 105 |
| | 109 | 111 |
| | 114 | 200 |
| | 116 | 202 |
| | 118 | 210 |
| | 119 | 212 |
| | 124 | 213 |
| | 201 | |
| | 203 | |
| | 205 | |
| | 207 | |
| | 208 | |
| | 209 | |
| | 214 | |
| | 215 | |
| | 219 | |
| | 220 | |
| | 221 | |
| | 222 | |
| 223 | | |
| 228 | | |
| 231 | | |
| 232 | | |
| 233 | | |

Training set includes 33 records and test set includes equally 11 records. Training set ratio is 75% and test set ratio is 25%. Healthy test set includes different 3 records and training set includes different 8 records. There are 27.27% healthy records exist in test and 72.72% healthy records exist in training set. Disease test set includes 8 disease records and training set includes 25 disease records. There are 24.24% disease records exist in test and 75.75% records exist in training set. The training set and test ratio is suitable for CNN models.

4.4.1 ECG Model & Results in Altered Training and Test Set

After the model is trained with this specific properties, it had 78.79% success rate in training set and 72.72% in test set as shown in Figure 4.12. When the epochs performance is analyzed, in 140th epoch training there is a peak in test set accuracy which gets up to 90%. However, in other epochs and in general, model is not learning and classification improvement is always stays at 72.72%. Especially after 70th epoch training set accuracy gets higher until and gets lower after 125th epoch. In this period, test set accuracy did not change. It indicates that the model is not learning the correct patterns of features in the training set to classify new data. This issue specifies a problem with ECG lines cannot be determined and classified by the model.

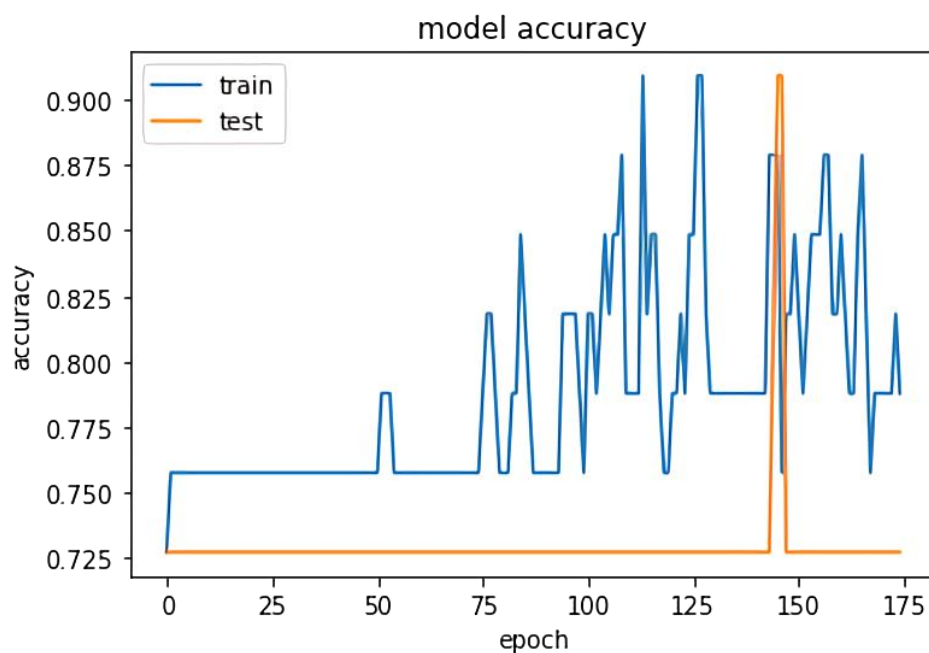


Figure 4.12. ECG lines accuracy comparison of training set and testing set in altered dataset

In Figure 4.13, it can be seen that the test set model loss never drops and stays steady. But in training, model loss is getting lower and lower and that is the reason why the model accuracy is getting better. This means that the model can train itself in order to classify the training dataset, but it does not train itself for the test dataset. Even though the model can classify the training set successfully, the accuracy of classifying the test set almost never changes. As a result, model accuracy gets better in the training set but not in the test set. This situation shows the problem of identifying the test set records.

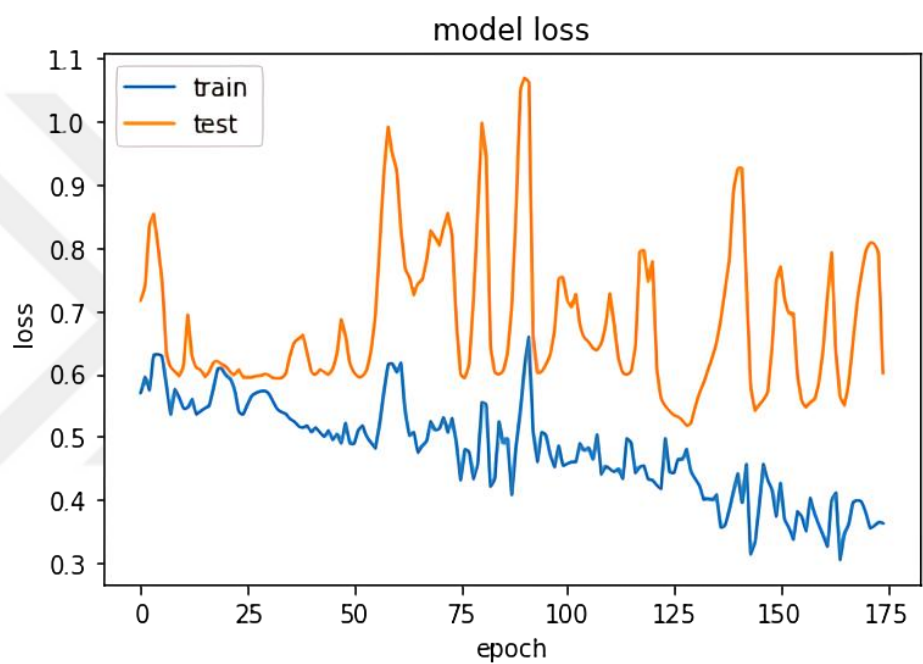


Figure 4.13. ECG lines model loss comparison of training set and testing set in altered dataset

There is not a big accuracy result difference between training accuracy and validation (test) accuracy in the results. Among 11 test records, the model can classify 8 of them correctly and 3 of them faulty. Healthy test set accuracy is 0%. Among the 3 healthy records, the model cannot classify any record correctly. For r103, r112 and r234 records, the model classifies as a disease record. On the other hand, disease test set accuracy is 100%. Among 8 disease records, the model can classify all the records correctly. The model did not classify any record faulty. The testing set and training set ratio is 75% and 25% in the all dataset. With this ratio we had 72.72% test accuracy. Figure 4.14 summarizes the outputs of ECG Results with altered test data selection.

| Outputs ECG Results | | | | | | | |
|---------------------|----------|--------------|---------|---------|---------|---------|---------|
| Epoch Count | Accuracy | Val Accuracy | | | | | |
| 175 | 78.79% | 72.72% | | | | | |
| Healthy | | | | | | | |
| r103 | r112 | r234 | | | | | |
| Disease | Disease | Disease | | | | | |
| Disease | | | | | | | |
| r100 | r105 | r111 | r200 | r202 | r210 | r212 | r213 |
| Disease | Disease | Disease | Disease | Disease | Disease | Disease | Disease |

Figure 4.14. ECG lines model outputs of training set and testing set in altered dataset

Especially, it can be observed that there is a very low accuracy which is 0%, in healthy test set. Model cannot classify any records correctly. There are 3 test records in healthy test set and 8 records in healthy training set. Even though there is a good ratio between the test set and training set, model could not able to classify any records correctly. 8 records can be not enough model to learn the important and essential features of the healthy records. Also, due to the images are presented in ECG form, they can be similar to the model and cannot extract the important differences between disease and healthy records. When all classification results are analyzed it can be seen that model classified all testing records as a disease.

On the other hand, there are 25 records in the training set and 8 records in test set for disease records. There are enough records in the training set to train itself for identifying the disease records. Model has 100% accuracy to classify disease records. As mentioned earlier, model tries to classify any new data as a disease. This situation indicates that model did not learn and improve itself in a healthy way. It shows an overfitting problem to model classifies all the test records as a disease.

4.4.2 Phase Space Model & Results in Altered Training and Test Set

After the model is trained with this specific properties, it had 96.97% success rate in training set and 90.90% in test set as seen in Figure 4.15. When the epochs performance is analyzed, after 75th epoch training set accuracy and testing set accuracy is going higher performance in parallel. Although there is an outlier epoch performance in 105th

epoch, in general, the model gets higher accuracy in training set and testing set. Due to training set and test set accuracy is getting better epoch by epoch, it shows there is not any overfitting problem in the model. With 90% and more accuracy is a very important success for classifying the test data.

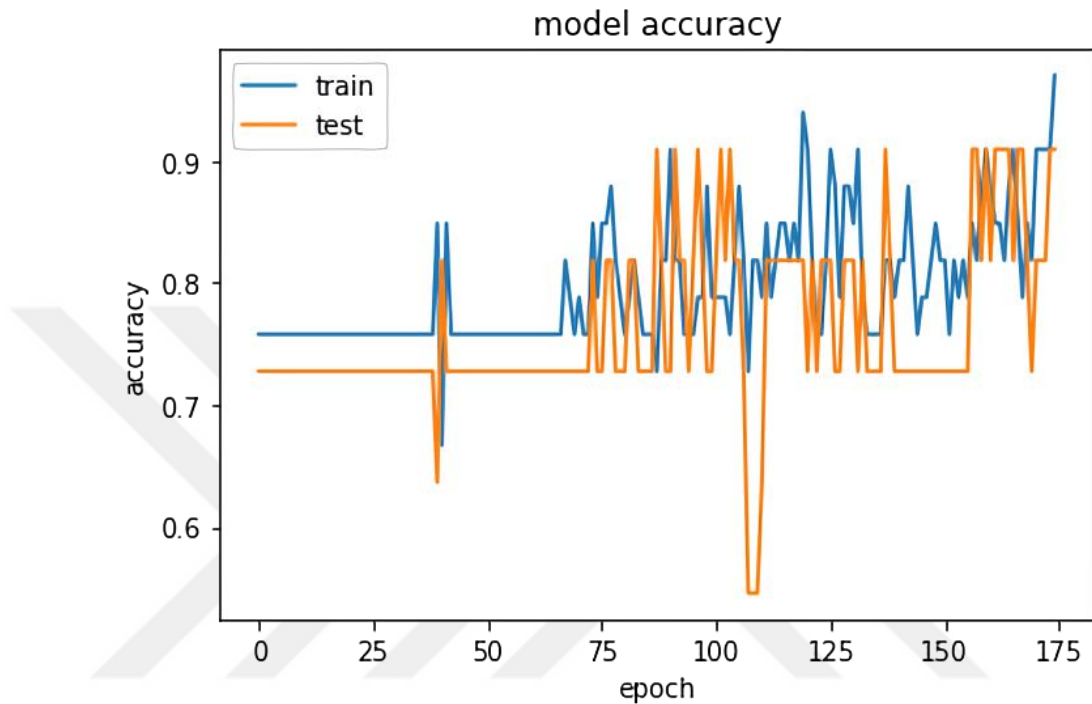


Figure 4.15. Phase Space model loss comparison of training set and testing set in altered dataset

In Figure 4.16, it can be seen that the training and testing loss is parallel. It shows the training ratio and data augmentation prevented the overfitting problem. This means that model can train itself in order to classify training dataset and it can classify test dataset accurately. Result of that, model accuracy gets better in training set and test set. This is a well-trained model to classify healthy heart and disease heart.

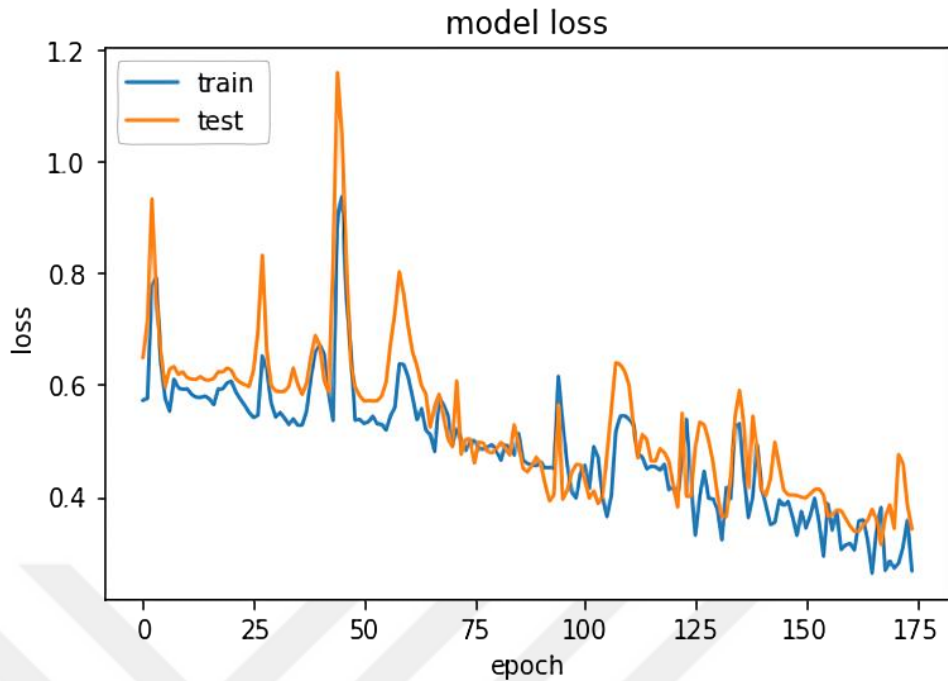


Figure 4.16. Phase Space accuracy comparison of training set and testing set in altered dataset

Among of 11 test records, model can classify 10 of them correctly and only one record faulty. Healthy test set accuracy is 100%. Among of the 3 healthy records, model can classify all the records correctly. On the other hand, disease test set accuracy is 87.5%. Among of 8 disease records, model could classify all the records correctly expect r100. This record is classified as a healthy record even it is a disease health. Figure 4.16 summarizes the output of the phase space results with altered data records.

| Outputs of Phase Space Results | | | | | | | |
|--------------------------------|----------|--------------|---------|---------|---------|---------|---------|
| Epoch Count | Accuracy | Val Accuracy | | | | | |
| 175 | 96.97% | 90.90% | | | | | |
| Healthy | | | | | | | |
| r103 | r112 | r234 | | | | | |
| Healthy | Healthy | Healthy | | | | | |
| Disease | | | | | | | |
| r100 | r105 | r111 | r200 | r202 | r210 | r212 | r213 |
| Healthy | Disease | Disease | Disease | Disease | Disease | Disease | Disease |

Figure 4.17. Phase Space model outputs of training set and testing set in altered dataset

The testing set and training set ratio is 75% and 25% in the all dataset. With this ratio 90.90% test accuracy has obtained. Especially, it can be observed that model can

classify health records with best accuracy almost 100% in test set. There are 3 test records in healthy test set and 8 records in healthy training set. With having a good ratio between the test set and training set, model could able to classify any records correctly in the phase space images. Due to the images are presented in phase space form, healthy and diseases heart differences are become clear and model could able to extract the important differences between disease and healthy records. When all classification results are analyzed it can be seen that model can classify the disease and healthy records successfully.

On the other hand, there are 25 records in the training set and 8 records in test set for disease records. There are enough records in the training set to train itself for identifying the disease records. Model has 87.5% accuracy to classify disease records. There are not any issues related with overfitting in this model training. The training accuracy and testing accuracy are similar and has very high accuracy.

5. CONCLUSION

5.1 COMPARISON CNN RESULTS WITH SAMPLE PROJECT DATASET

In Biomedical Signal Processing and Control, Mondéjar-Guerra et al. 2019 shows hearth beats are classified with %89 accuracy [34]. According to the study, MIT-BIH Arrhythmia Database the data is used with beat per beat in the records heart among 110.000 beats. In this thesis, only 44 records are classified according to their heart health condition.

According to Biomedical Signal Processing and Control training set and test set article, two models are created and trained. In one model, ECG lines of the records are used as inputs. In the other model, phase space graph is used which is created with third derivate Taylor formula. For both models, same model structure is used and they are trained with the 80 epochs. Only difference from these two models are in the input images, one model is trained with ECG lines and the other model is trained with phase space diagrams. Figure 5.1 shows an example of health data record r115 of ECG lines and Phase Diagram of the same patient.

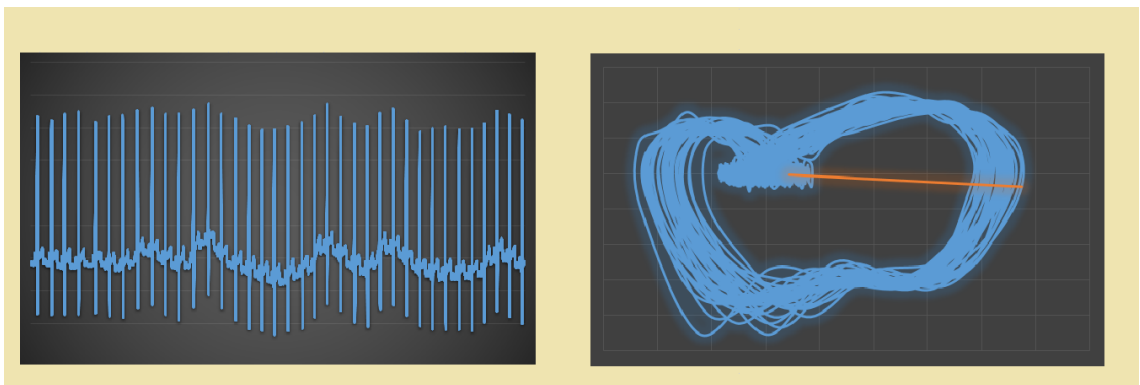


Figure 5.1. ECG lines and Phase Space Diagram of r115 healthy heart record

In the ECG model, the training set accuracy is 95.45% whereas test set accuracy is 63.63%. This can be considered as typical overfitting problem in CNN. Model could able to classify the training set records correctly, but when it faced with new records, it classified correctly 14 records in 22 records. Model learned the wrong or faulty features

in the training set the determine a records heart situation. It could not extract the important details or essential attribute in the images. Especially after 30th epoch, while the performance of the model should have increased, it was observed that it was falling. Although the model started to classify the records with 70% accuracy, it declined to %50 as epochs progressed. In the final epoch, test accuracy was 63.63%. It can be said that this model does not provide a good performance in classifying healthy or disease hearth.

In the phase space diagram model, the training set accuracy is 95.45% but test set accuracy is 72.72%. Like ECG model, this model has also an overfitting problem. Model could able to classify the training set records correctly, but when it faced with new records, it classified correctly 16 records in 22 records. Phase space diagram model has the same problem with ECG Model. It learned the wrong or faulty features in the training set the determine a records heart situation. It could not extract the important details or essential attribute in the images. In phase space diagram model, testing accuracy is between 65-80%. In final epoch, test accuracy output is 72.72%. It can be said that this model has a nearly acceptable good performance in classifying healthy or disease hearth.

When this two models are examined, it easily can be seen that the input of phase space diagram images better than ECG lines. With the same epoch and same model architecture, ECG lines model accuracy dropped significantly to 63.63%. But the phase space model accuracy even got higher up to %80's but finished at 72.72%. Phase space diagram accuracy has never dropped to 63.63%, which is the ECG lines models final accuracy rate. It is an indisputable fact that phase space images could made a difference against the ECG images. CNN could able to extract more important features and attributes in phase space images rather than ECG lines. But in both model could not surpass Biomedical Signal Processing and Control SVM model which has 89% accuracy.

5.2 COMPARISON CNN RESULTS WITH ALTERED DATASET

Biomedical Signal Processing and Control article, dataset proportion for test set and training set is equally distributed to 50%. But in a CNN model, this proportion is generally 75% training set and 25% test set or 80% training set and 20% test set. With 75% training set and 25% test set proportion two different models are trained. Both models are trained with 175 epochs with the same CNN architecture. The only difference between them is the input images. In one model ECG lines of the records given as input images, in the other model phase space diagrams given as input images.

In the ECG model, the training set accuracy is 78.79% but test set accuracy is 72.72%. Even the model can be able to train itself with the training set, it has improved its accuracy very insignificant level. It classified correctly 8 records in 11 records. All the correct classified records are disease records and all the faulty classified records are healthy records. Model could not extract the important differences to identify a health heart or a disease hearth. It classifies all the records as a disease record. When the model training is analyzed, test accuracy is almost never changes. It can be said that this model does not provide a good performance in classifying healthy or disease hearth.

In the phase space diagram model, the training set accuracy is 96.97% but test set accuracy is 90.90%. As there is not much ratio difference is between training set and test set, there is not any overfitting problem in this model. Epoch by epoch model improves itself with training set and get better results and also for testing set it improves itself classification accuracy. As it can be seen the model analysis, both training set and test set accuracy and training loss and test set loss are parallel with each other. Model classified correctly 10 records in 11 records. Only one disease record is classified as a healthy record. Model could able to extract the important features to detect a healthy heart and a disease heart to make classifications. It can be said that this model has a good performance in classifying healthy or disease hearths.

When this two models are examined, it easily can be seen that the input of phase space diagram images accuracy ratio is far better than ECG lines images. With the same epoch and same model architecture, ECG lines model accuracy stayed steady with 72.72%. But the phase space model accuracy got higher up to 90.90%. The biggest different with these two dataset was the phase space diagram model, CNN could able to identify the healthy heart with the given image inputs. As a result of that, it could have classified more records correctly and accurately.

5.3 MOST ACCURATE MODEL

As it is mentioned in the previous parts of the thesis, the model architecture is same for all models. The only difference is the input images that provided to the CNN model.

Among the all models, the most accurate and successful model was the altered data set with phase space diagram with 90.90% accuracy. It is followed by equally 72.72% accuracy with sample project dataset ratio with phase spaces and altered data set with ECG lines. The most unsuccessful model is 63.63% with ECG lines with sample project dataset ratio.

Since there are enough training set and test set ratio in the altered dataset with phase space diagram, the model did not face with any overfitting problem. Also, due to the records images are changed to the phase space diagrams, more clear and understandable images are created. With this update, CNN could easily make detections between healthy heart and disease hearth. When the all models are failed to classify health hearths, this model could able to realize the differences of the healthy heart and disease heart with phase space images. With this update, it could able to surpass other models. Also, even the test set and training set ratio is changed, the model has better accuracy than the Biomedical Signal Processing and Control articles SVM model. When the altered dataset with phase space model classifies 90.90% accuracy, that SVM model could classify with 89%.

5.4 SUGGESTIONS

Even if the almost all records samples are observed with MLII signal, the number of the records can be considered as not enough for deep learning purposes. There are only 44 records in MIT-BIH Arrhythmia Database.

There are two layers in convolutional layer. The number of layers can be increased in order to have more accurate classification results. Finally, in fully connected layer there can be more hidden layers for model to update the weights and increase the performance of the model.



BIBLIOGRAPHY

- [1] Aje, Temilolu Olayinka. 2009. "Cardiovascular Disease: A Global Problem Extending Intro The Developing World". *World Journal Of Cardiology* 1 (1): 3. doi:10.4330/wjc.v1.i1.3.
- [2] Luz, Eduardo José da S., Thiago M. Nunes, Victor Hugo C. de Albuquerque, João P. Papa, and David Menotti. 2013. "ECG Arrhythmia Classification Based On Optimum-Path Forest". *Expert Systems With Applications* 40 (9): 3561-3573. doi:10.1016/j.eswa.2012.12.063.
- [3] Institute for Health Metrics and Evaluation, Findings from the Global Burden of Disease Study 2017.
- [4] Ebrahimi, Zahra, Mohammad Loni, Masoud Daneshtalab, and Arash Gharehbaghi. 2020. "A Review On Deep Learning Methods For ECG Arrhythmia Classification". *Expert Systems With Applications: X* 7: 100033. doi:10.1016/j.eswax.2020.100033.
- [5] Meek, S. 2002. "ABC of clinical electrocardiography: Introduction. I---Leads, rate, rhythm, and cardiac axis", *BMJ*, 324(7334), pp. 415-418. doi: 10.1136/bmj.324.7334.415.
- [6] Tse, Gary. 2016. "Mechanisms Of Cardiac Arrhythmias". *Journal Of Arrhythmia* 32 (2): 75-81. doi:10.1016/j.joa.2015.11.003.
- [7] Atrial Fibrillation with LVH and Strain Michael Rosengarten BEng, MD. McGill, 2012. EKG World Encyclopedia <http://cme.med.mcgill.ca/php/index.php>, courtesy of Michael Rosengarten BEng, MD.McGill.

- [8] Stockman, G. C., and Kanal, L. N., 1983. Problem reduction representation for the linguistic analysis of waveforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3, 287-298.
- [9] Iwata, A., Nagasaka, Y., and Suzumura, N., 1990. Data compression of the ECG using neural network for digital Holter monitor. *IEEE Engineering in Medicine and Biology Magazine*, 9,3, 53-57.
- [10] Xue, Q., Hu, Y. H., and Tompkins, W. J., 1992. Neural-network-based adaptive matched filtering for QRS detection. *IEEE Transactions on Biomedical Engineering*, 39,4, 317- 329.
- [11] Kundu, M., Nasipuri, M., Basu, D. K., and Bhattacharya, A., 1993. A reasoning system for on-line interpretation of ECG signal. In *TENCON'93. Proceedings. Computer, Communication, Control and Power Engineering. 1993 IEEE Region 10 Conference, Ekim, Çin, Bildiriler Kitabı: 626-630.*)
- [12] Xue, Q., and Reddy, B. S., 1997. Late potential recognition by artificial neural networks. *IEEE Transactions on Biomedical Engineering*, 44,2, 132-143.
- [13] Lena Biel., and Ola Pettersson 2001. ECG analysis: A new approach in human identification *IEEE Transactions on Instrumentation and Measurement* 50(3):808 - 812
- [14] Jekova, I., and Krasteva, V., 2004. Real time detection of ventricular fibrillation and tachycardia. *Physiological Measurement*, 25,5, 1167.
- [15] Rodriguez, J., Goni, A., and Illarramendi, A., 2005. Real-time classification of ECGs on a PDA. *Transactions on Information Technology in Biomedicine*, 9,1, 23-34.
- [16] Ye, C., Coimbra, M. T., and Kumar, B. V., 2010. Arrhythmia detection and classification using morphological and dynamic features of ECG signals. In *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference*

- [17] Shivajirao M Jadhav, Sanjay L Nalbalwar, Ashok A Ghatol 2011. Modular neural network based arrhythmia classification system using ECG signal data. *International Journal of Information Technology and Knowledge Management*, 2011, 4,1, 205-209
- [18] Li Sun, Yanping Lu, Kaitao Yang, Shaozi Li, 2012. ECG Analysis Using Multiple Instance Learning for Myocardial Infarction Detection - *IEEE transactions on bio-Medical Engineering* 59(12)
- [19] Luz, E. J. da S., Nunes, T. M., de Albuquerque, V. H. C., Papa, J. P., & Menotti, D. 2013. ECG arrhythmia classification based on optimum-path forest. *Expert Systems with Applications*, 40(9), 3561–3573. doi:10.1016/j.eswa.2012.12.063
- [20] Huanhuan, M., & Yue, Z., 2014. Classification of Electrocardiogram Signals with Deep Belief Networks. 2014 IEEE 17th International Conference on Computational Science and Engineering. doi:10.1109/cse.2014.36
- [21] Chauhan, S., & Vig, L., 2015. Anomaly detection in ECG time signals via deep long short-term memory networks. 2015 IEEE International Conference on Data Science and Advanced Analytics (DSAA). doi:10.1109/dsaa.2015.7344872)
- [22] Lei, X., Zhang, Y., & Lu, Z. 2016, October. Deep learning feature representation for electrocardiogram identification. In 2016 IEEE International Conference on Digital Signal Processing (DSP) (pp. 11-14)
- [23] Zhang, Q., Zhou, D., & Zeng, X. 2017. HeartID: A multiresolution convolutional neural network for ECG-based biometric human identification in smart health applications. *IEEE Access*, 5, 11805-11816
- [24] Eduardo, A., Aidos, H., & Fred, A. L. 2017, February. ECG-based Biometrics using a Deep Autoencoder for Feature Learning-An Empirical Study on Transferability. In ICPRAM (pp. 463-470)

- [25] da Silva Luz, E. J., Moreira, G. J., Oliveira, L. S., Schwartz, W. R., & Menotti, D. 2017. Learning deep off-the-person heart biometrics representations. *IEEE Transactions on Information Forensics and Security*, 13(5), 1258-1270
- [26] Salloum, R., & Kuo, C. C. J. 2017, March. ECG-based biometrics using recurrent neural networks. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp. 2062-2066)
- [27] Yakut, Ö. 2018. EKG işaretindeki aritmilerin yumuşak hesaplama algoritmaları kullanılarak sınıflandırılması, Kocaeli Üniversitesi, Fen Bilimleri Enstitüsü, Available at: <http://dspace.kocaeli.edu.tr:8080/xmlui/handle/11493/1034>
- [28] Arikan, O and Terzi, M. 2018. "Detection of acute myocardial ischemia based on support vector machines", 2018 26th Signal Processing and Communications Applications Conference (SIU). doi: 10.1109/siu.2018.8404733
- [29] Labati, R. D., Muñoz, E., Piuri, V., Sassi, R., & Scotti, F. 2018. Deep-ECG: Convolutional neural networks for ECG biometric recognition. *Pattern Recognition Letters*
- [30] Abdeldayem, S. S., & Bourlai, T. 2018. ECG-based Human Authentication using High-level Spectro-temporal Signal Features. In *2018 IEEE International Conference on Big Data (Big Data)* (pp. 4984-4993). IEEE.
- [31] Arikan, O. and Terzi, M. 2019. "Coronary Artery Disease Detection by using Support Vector Machines and Gaussian Mixture Model", 2019 Medical Technologies Congress (TIPTEKNO). doi: 10.1109/tiptekno.2019.8894953
- [32] Hammad, M., & Wang, K. 2019. Parallel score fusion of ECG and fingerprint for human authentication based on convolution neural network. *Computers & Security*, 81, 107-122

- [33] Jothiramalingam, R. et al. 2020. "Machine learning-based left ventricular hypertrophy detection using multi-lead ECG signal", *Neural Computing and Applications*. doi: 10.1007/s00521-020-05238-2
- [34] Mondéjar-Guerra, V. et al. 2019. "Heartbeat classification fusing temporal and morphological information of ECGs via ensemble of classifiers", *Biomedical Signal Processing and Control*, 47, pp. 41-48. doi: 10.1016/j.bspc.2018.08.007
- [35] Khan IR, Ohba R. Closed form expressions for the finite difference approximations of first and higher derivatives based on Taylor series. *J. Comp. Appl. Math.* 1999; 107: 179–193.
- [36] Ronco E, Arsan T, Gawthrop PJ. Open-loop intermittent feedback control: Practical continuous-time GPC. *IEE Proceedings - Control Theory and Applications*, 1999; 146(5): 426-434.
- [37] Faggella, D. 2020. What is Machine Learning? - An Informed Definition, Emerj. Available at: <https://emerj.com/ai-glossary-terms/what-is-machine-learning/>
- [38] Sarioğlu, C, 2019. Yüksek Lisans Tez Savunması (Cemile Sarıcaoğlu) - Bilgisayar Mühendisliği Bölümü, Mf-bm.gazi.edu.tr
- [39] Mayo, H. 2018 History of Machine Learning. Available at: <https://www.doc.ic.ac.uk/~jce317/history-machine-learning.html>
- [40] Vinyals, O. et al. 2019. "Grandmaster level in StarCraft II using multi-agent reinforcement learning", *Nature*, 575(7782), pp. 350-354. doi: 10.1038/s41586-019-1724-z
- [41] Kızılkaya Y. M., Oğuzlar A., 2018. Bazı Denetimli Öğrenme Algoritmalarının R programlama dili ile kıyaslanması , *Karadeniz*, 37, s:90-98

- [42] Cerebro 2018, Yeni Başlayanlar için Makine Öğrenmesi Algoritmaları: Available at:<https://medium.com/t%C3%BCrkiye/yeni-ba%C5%9Flayanlar-i%C3%A7in-makine-%C3%B6%C4%9Frenmesi-algoritmalar%C4%B1-ae22f794af2f>
- [43] A. M. Turing, "Computing machinery and intelligence," in Parsing the Turing Test: Springer, 2009, pp. 23-65
- [44] P. Kashyap, 2017. "Industrial applications of machine learning," in Machine Learning for Decision Makers: Springer, pp. 189-233
- [45] Y. LeCun, Y. Bengio, and G. Hinton, 2015. "Deep learning," nature, vol. 521, no. 7553, pp. 436-444
- [46] Devin K. Phillips, 2015. Speed of the Human Brain Arizona State University School of Life Sciences Ask A Biologist
- [47] Patterson, J., & Gibson, A. 2017. Deep learning: A practitioner's approach. O'Reilly Media, Inc.
- [48] Sze, V., Chen, Y. H., Yang, T. J., & Emer, J. S. 2017. Efficient processing of deep neural networks: A tutorial and survey. Proceedings of the IEEE, 105(12), 2295-2329
- [49] Bayır, F.,2006. Yapay sinir ağları ve tahmin modellemesi üzerine bir uygulama, Yüksek Lisans Tezi” İstanbul Üniversitesi Sosyal Bilimler Enstitüsü, İstanbul
- [50] Zhou, V. 2019. Machine Learning for Beginners: An Introduction to Neural Networks. Towards Data Science
- [51] Tormod Næs, Knut Kvaal, Tomas Isaksson, Charles Miller, 1993. Artificial Neural Networks in Multivariate Calibration

[52] Aşkın, D., İskender İ. ve Mamızadeh A., 2011. “Farklı yapay sinir ağları yöntemlerini kullanarak kuru tip transformatör sargısının termal analizi”, Gazi Üniv. Müh. Mim. Fak. Der., Ankara, 26(4): 905-913

[53] Keleşoğlu, Ö.,2006. Yapay sinir ağları ile betonarme kiriş kesitlerin analizi, İMO Teknik Dergi, 3935-3942

[54] Krizhevsky, A., Sutskever, I., and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems 25, pages 1097-1105. Curran Associates, Inc.

[55] Karpathy, A., & Li, F. F. 2018. Stanford CS class CS231n: Convolutional Neural Networks for Visual Recognition. Course Notes, Link: <http://cs231n.github.io>.

[56] Peng, M; Wang, C; Chen, T; Liu, G; Fu, X; 2017. Dual Temporal Scale Convolutional Neural Network for Micro-Expression Recognition - UCL Discovery Available at: <https://discovery.ucl.ac.uk/id/eprint/10045102>

[57] İnik, Ö. and Ülker, E. 2017. "Derin Öğrenme ve Görüntü Analizinde Kullanılan Derin Öğrenme Modelleri", Gaziosmanpaşa Bilimsel Araştırma Dergisi, 6(3), pp. 85-104

[58] Bantupalli and Xie, 2019. American Sign Language Recognition Using Machine Learning and Computer Vision

[59] Convolutional Neural Networks (CNN): Step 1- Convolution Operation SuperDataScience. Available at: <https://www.superdatascience.com/blogs/convolutional-neural-networks-cnn-step-1-convolution-operation>

[60] Scherer, D., Müller, A. and Behnke, S. 2010. "Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition", Artificial Neural Networks – ICANN 2010, pp. 92-101. doi: 10.1007/978-3-642-15825-4_10.

[61] Zeiler, M. and Fergus, R. 2013. Visualizing and Understanding Convolutional Networks, arXiv.org. Available at: <https://arxiv.org/abs/1311.2901>

[62] Sze, V., Chen, Y. H., Yang, T. J., & Emer, J. S. 2017. Efficient processing of deep neural networks: A tutorial and survey. Proceedings of the IEEE, 105(12), 2295-2329

[63] Dropout Function (Srivastava, Nitish et al. 2014. Dropout: a simple way to prevent neural networks from overfitting, JMLR 2014

[64] Team, K. Keras documentation: Image data preprocessing, Keras.io. Available at: <https://keras.io/api/preprocessing/image/>

CURRICULUM VITAE

Personal Information

Name and surname: Bekir Yavuz Koç

Academic Background

Bachelor's Degree Education: Industrial Engineer

Foreign Languages: English

Work Experience

Institutions Served and Their Dates:

Huawei – Android Developer – Apr 2020 – Currently working

P.I Works – Software Support Intern – Oct 2019 – Apr 2020

