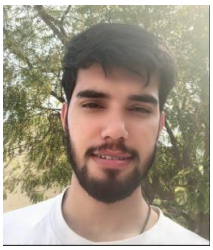
ECG ARRHYTHMIA CLASSIFICATION

by

Putul Siddharth
(17BCE1100)

Rahul Mahajan
(17BCE1063)

Anant Sirohi
(17BCE1170)



a project report submitted to

Prof. PRABHAKAR RAO

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

In partial fulfilment of the requirements for the course

CSE4015 – Human Computer Interaction

in

B.tech COMPUTER SCIENCE ENGINEERING

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

VIT UNIVERSITY, CHENNAI

Vandalur – Kelambakkam

Road Chennai – 600127

ACKNOWLEDGEMENT

We wish to express our sincere thanks and deep sense of gratitude to our project guide, **Prof. Prabhakar Rao**, Senior Professor, School of Computer Science Engineering, for his consistent encouragement and valuable guidance offered to us from time to time in order to rectify our errors and bring professionalism in our projects.

We are extremely grateful to **Dr. R. Jagadeesh Kannan**, Dean of the School of Computer Science Engineering, VIT Chennai, for extending the facilities of the School towards our project and for her unstinting support.

We express our thanks to our Programme Chair **Dr. Justus.** for their support throughout the course of this project.

We also take this opportunity to thank all the faculty of the school for their support and their wisdom imparted to us throughout the course.

We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

TABLE OF CONTENTS

ABSTRACT

In this project, an intellectual based electrocardiogram (ECG) signal classification approach utilizing Deep Learning (DL) is being developed. ECG plays important role in diagnosing various Cardiac ailments. The ECG signal with irregular rhythm is known as Arrhythmia such as Atrial Fibrillation, Ventricular Tachycardia, Ventricular Fibrillation, and so on. The main aspire of this task is to screen and distinguish the patient with various cardio vascular arrhythmia. This examination encourages us to recognize diverse kinds of arrhythmia utilizing Deep Learning algorithm. Here we use Convolutional Neural Network (CNN) a DL algorithm which is efficient in classifying signals. Utilizing CNN, features are learned automatically from the time domain ECG signals which are acquired from MIT-BIH Database from Physiobank.com. The feature adapted specifically replaces manually extracted features and this analysis will help the Cardiologists in screening the patient with Cardiac illness effectively. The CNN is trained, tested using ECG Dataset obtained from MIT-BIH Database and from the signal 7 of arrhythmia were classified. The proposed system is compared for Various Activation function by varying the number of epochs. From the result obtained we came to know that ELU activation function gives better result with an accuracy of 90%.

INTRODUCTION

According to the World Health Organization (WHO), cardiovascular diseases (CVDs) are the number one cause of death today. Over 17.7 million people died from CVDs in the year 2017 all over the world which is about 31% of all deaths, and over 75% of these deaths occur in low and middle income countries. Arrhythmia is a representative type of CVD that refers to any irregular change from the normal heart rhythms. There are several types of arrhythmia including atrial fibrillation, premature contraction, ventricular fibrillation, and tachycardia. Although single arrhythmia heartbeat may not have a serious impact on life, continuous arrhythmia beats can result in fatal circumstances. For example, prolonged premature ventricular contraction (PVCs) beats occasionally turn into a ventricular tachycardia (VT) or ventricular fibrillation (VF) beats which can immediately lead to heart failure. Thus, it is important to periodically monitor the heart rhythms to manage and prevent the CVDs. Electrocardiogram (ECG) is a non-invasive medical tool that displays the rhythm and status of the heart. Therefore, automatic detection of irregular heart rhythms from ECG signals is a significant task in the field of cardiology. Arrhythmia is an important group of diseases in cardiovascular disease. Arrhythmia can occur on its own or with other cardiovascular diseases. The diagnosis of arrhythmia mainly depends on the ECG (electrocardiogram). ECG (electrocardiogram) is an important modern medical tool that records the process of cardiac excitability, transmission, and recovery. Automatic detection of irregular heart rhythms from ECG signals is a significant task for the automatic diagnosis of cardiovascular disease. We classify ECG into seven categories, one being normal and the other six being different types of arrhythmia using deep two-dimensional CNN with grayscale ECG images. By transforming one-dimensional ECG signals into two-dimensional ECG images, noise filtering and feature extraction are no longer required. This is important since some of ECG beats are ignored in noise filtering and feature extraction. In addition, training data can be enlarged by augmenting the ECG images which results in higher classification accuracy. Data augmentation is hard to be applied in 1-d signals since the distortion of 1-d ECG signal could downgrade the performance of the classifier. However, augmenting two-dimensional ECG images with different cropping methods helps the CNN model to train with different viewpoints of the single ECG images. Using ECG image as an input data of the ECG arrhythmia classification also benefits in the sense of robustness.

PROBLEM STATEMENT

The process of identifying and classifying arrhythmias can be very troublesome for a human being because sometimes it is necessary to analyze each heartbeat of the ECG records, acquired by a holter monitor for instance, during hours, or even days. In addition, there is the possibility of human error during the ECG records analysis, due to fatigue. Automatic detection and classification of life-threatening arrhythmia plays an important part in dealing with various cardiac conditions. So in this project, a novel method for classification of various types of arrhythmia is presented.

OBJECTIVE

The objective of the work is to detect and classify cardiac arrhythmias of patients from the ECG signal. The detection of the QRS complex (characteristic wave of the ECG) is gaining momentum and many algorithms have been reported in the literature for R-peak detection.

EXISTING SYSTEM

The process of identifying and classifying arrhythmias can be very troublesome for a human being because sometimes it is necessary to analyze each heartbeat of the ECG records, acquired by a holter monitor for instance, during hours, or even days. In addition, there is the possibility of human error during the ECG records analysis, due to fatigue. An alternative is to use computational techniques for automatic classification. The heart is a muscle that contracts in a rhythmical manner, pumping blood throughout the body. This contraction has its beginning at the atrial sine node that acts as a natural pacemaker, and propagates through the rest of the muscle. This electrical signal propagation follows a pattern. As a result of this activity, electrical currents are generated on the surface of the body, provoking variations in the electrical potential of the skin surface. These signals can be captured or measured with the aid of electrodes and appropriate equipment.

On equipment belonging into the on-the-person category, three or more electrodes are used to obtain the signal, in which one of them serves as a reference for the others. Usually, the reference electrode is placed near the right leg. As such, there can be different visions of the ECG signal, depending on the pair of electrodes chosen to construct the signal. These differentiated visions are given the name of leads.

A widely used configuration of electrodes is one composed of 5 electrodes : one of the electrodes is positioned on the left arm (LA), one on the right arm (RA), one on the left leg (LL), one on the right leg (RL) and one on the chest, to the right of the external (V or V1). Another widely employed setup uses 10 electrodes, where 5 extra electrodes (besides V or V1 on the chest and LA, RD, LL and RA on legs and arms) are positioned on the chest (V2 to V6) allowing a formation of 12 leads. From these configurations, several different leads can be constructed to visualize the ECG signal. Lead II is one of the most utilized for diagnosing heart diseases. It highlights various segments within the heartbeat, besides displaying three of the most important waves: P, QRS and T. These waves correspond to the field induced by the electrical phenomena occurring on the heart surface, denominated atrial depolarization (P wave), ventral depolarization (QRS complex wave) and repolarization (T wave). The patterns provoked by arrhythmias can deeply change these waves. Meanwhile, lead V and

its correlate leads (V1, V2) favor the classification of ventricular related arrhythmias, since there are electrodes positioned on the chest, improving the registry of action potentials on ventricular muscle. Heartbeat segmentation methods (*i.e.*, detection of the R peak or the QRS complex) have been studied for more than three decades, and the generations of these algorithms and newly developing methods reflect the evolution of the processing power of computers. With the facility of using faster processing computers, authors stopped worrying about computational cost and started concentrating on the heartbeat segmentation accuracy. Two measures are usually considered for evaluating the accuracy of heartbeat segmentation: sensitivity and positive predictively. For a fair comparison of the methods focusing on the heartbeat segmentation, a standard database needs to be used.

The most utilized, and recommended by ANSI/AAMI for the validation of medical equipment, is the MIT-BIH database for arrhythmia analysis – in this case, used for heartbeat segmentation. The feature extraction stage is the key to the success in the heartbeat classification of the arrhythmia using the ECG signal. Any information extracted from the heartbeat used to discriminate its type maybe considered as a feature. The features can be extracted in various forms directly from the ECG signal's morphology in the time domain and/or in the frequency domain or from the cardiac rhythm.

The RR interval is the time between the R peak of a heartbeat with respect to another heartbeat, which could be its predecessor or successor. With exception of patients that utilize a pacemaker, the variations perceived in the width of the RR interval are correlated with the variations in the morphology of the curve, frequently provoked by arrhythmias. Thus, the features in the RR interval have a great capacity to discriminate the types of heartbeats and some authors have based their methods only on using the RR interval features. Variations of this feature are used to reduce noise interference and are very common, *e.g.*, the average of the RR interval in a patient for a certain time interval. Among these intervals, the QRS interval, or the duration of the QRS complex, is the most utilized. Some types of arrhythmias provoke variations in the QRS interval, making it a good discriminating feature. The simplest way to extract features in the time domain is to utilize the points of the segmented ECG curve, *i.e.*, the heartbeat, as features.

Results presented in literature usually use the MIT-BIH database (also known as MIT-BIH ARR DB) that is extremely unbalanced. We believe that one major obstacle to achieving advances in the research focusing on fully-automatic classification of heartbeats (arrhythmias) in ECG is the reduced number of available databases. Therefore, we suggest to the research community dedicated to study the heartbeat classification problem that they encourage/stimulate the extension of databases dedicated for this end. We also suggest the use of new trends to capture the ECG signal, such as *off-the-person* approaches, for the elaboration of new databases.

Nonetheless, we believe that the creation of such databases would be a great challenge because, besides the financial costs involved, they would have to be incorporated into standards such as AAMI standards to reach the desired audience. Efforts to create new databases or even to increase the size of existing ones, as well as creating standard evaluation protocols, should be made in several research areas involving pattern recognition, specially to avoid unfair comparisons between methods and heartbeat segmentation.

FUTURE SCOPE

Automatic heartbeat classification is essential for real-time applications in detection of cardiac arrhythmias. The obtained results of this thesis suggest that there is a possibility growth of future in automatic ECG classification systems. The systems must include four decisive steps: pre-processing, QRS complex detection, features extraction and classification of heartbeats. Further effort of this work should move towards proposing new feature extraction and classification methods. Future progresses can be directed as follows, various other transforms like discrete cosine transform (DCT), and other time frequency methods can be used to extract the features that may improve the classification accuracy. In order to improve detection rate, varying ECG beat length may be chosen for the feature extraction and classification. To modify the network structure of the classifier to achieve better classification accuracy compared to existing heartbeat classifiers. As an extension of three feature selection method, the application of other dimensionality reduction techniques can be carried out as a future direction. The Supervised classification system are used for ECG beat detection, so the unsupervised methods may be adapted for classifying real-time signal analysis. Five methods were proposed in this thesis for ECG beat classification, each will have misclassification patterns in supervised classification system. Therefore an approach that fuses all these methods can be adapted to improve the performance of the system by reducing misclassification. 127 currently, there are many machine learning (ML) techniques are used for classifying ECG data. However, the main disadvantages of these ML results is the use of heuristic features with low feature learning architectures. In an alternate approach, the use of deep learning architectures is emphasized for making final decision about ECG classes. This deep learning architectures approach reduces the task of developing new feature extractor for every problem. The learning algorithms is inspired by the structure and function of the brain called artificial neural networks. Due to these advantages, deep learning is highly demand classifiers that may be used for future work in classifying ECG beats.

PROPOSED SYSTEM

1. Getting Data -

I have used the MIT-BIH arrhythmia database for the CNN model training and testing as has been mentioned in the paper. The MIT-BIH arrhythmia Database contains 48 half-hour excerpts of two-channel ambulatory ECG recordings, obtained from 47 subjects studied by the BIH arrhythmia laboratory between 1975 and 1979. For each record there are three files: 1. Annotation file 2. Signals file 3. Header file.

2. Data Augmentation –

Data augmentation means increasing the number of data points. In terms of images, it may mean that increasing the number of images in the dataset. We can achieve both high specificity and sensitivity by augmenting and balancing input data. I have augmented six ECG arrhythmia beats (PVC, PAB, RBB, LBB, APC, VEB) with nine different cropping methods: left top, center top, right top, center left, center, center right, left bottom, center bottom, and right bottom. Each cropping method results in two of three sizes of an ECG image that is 96 x 96. Then, these augmented images are resized to the original size which is 128 x 128.

3. Find the Rpeaks –

To do the above thing, we have used biosppy.signals.ecg.christov_segmenter() which helps in finding the RPeaks which helped in plotting the image. With the help of RPeaks, we can get the graph of each of the newly created augmented 2d data.

With the help of these graphs, the doctors can find whether the patient has normal heart rate or abnormal rate.

4. VGG net Model

It usually refers to a deep convolutional network for object recognition developed and trained by Oxford's renowned Visual Geometry Group (VGG), which achieved very good performance on the ImageNet dataset.

It is quite famous because not only it works well, but the Oxford team have made the structure and the weights of the trained network freely available online.

Application:

- Given image → find object name in the image
- It can detect any one of 1000 images
- It takes input image of size 224 * 224 * 3 (RGB image)

Built using:

- Convolutions layers (used only 3*3 size)
- Max pooling layers (used only 2*2 size)
- Fully connected layers at end
- Total 16 layers

FLOW OF OUR PROJECT

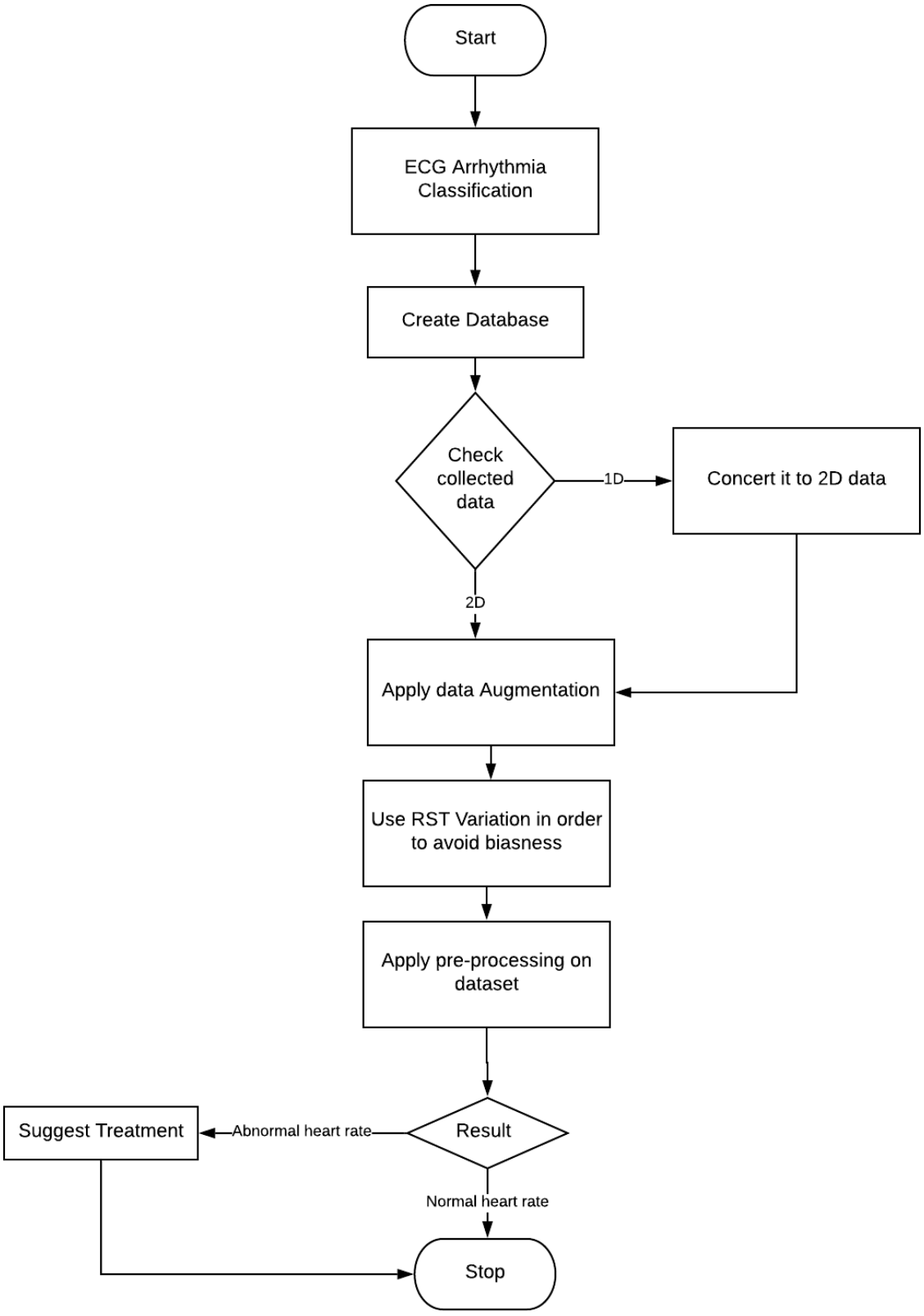
1. We will first create dataset using different csv files available in Kaggle and other resources.

2. After getting the required dataset, we need to do pre-processing as the data have is in 1D, so we need to convert it into 2D form.

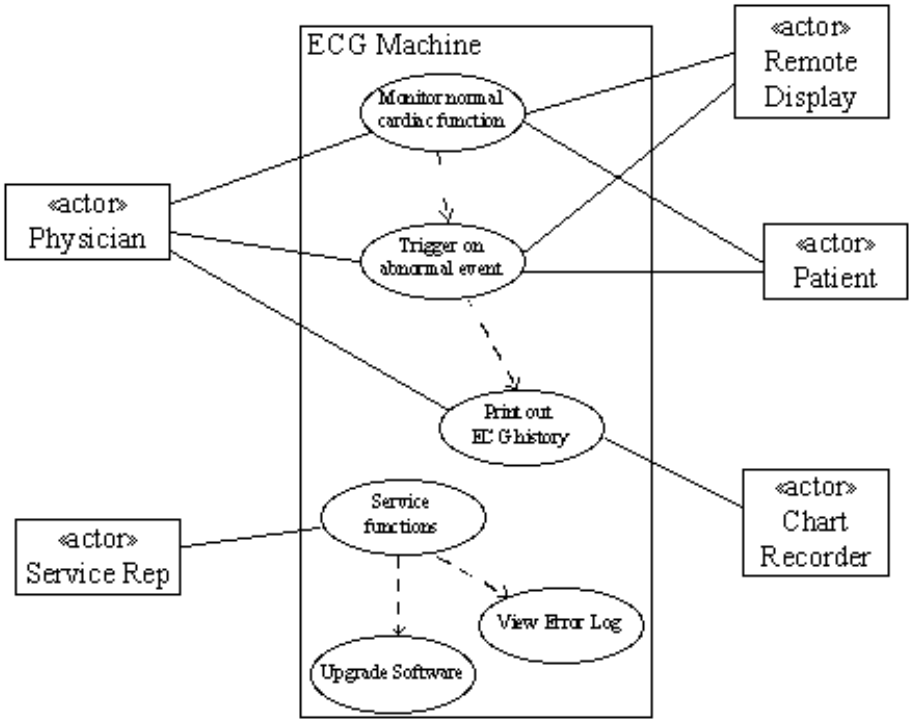
3. As data augmentation is difficult to apply on 1D ECG signal data as it could degrade the performance of the classifier.

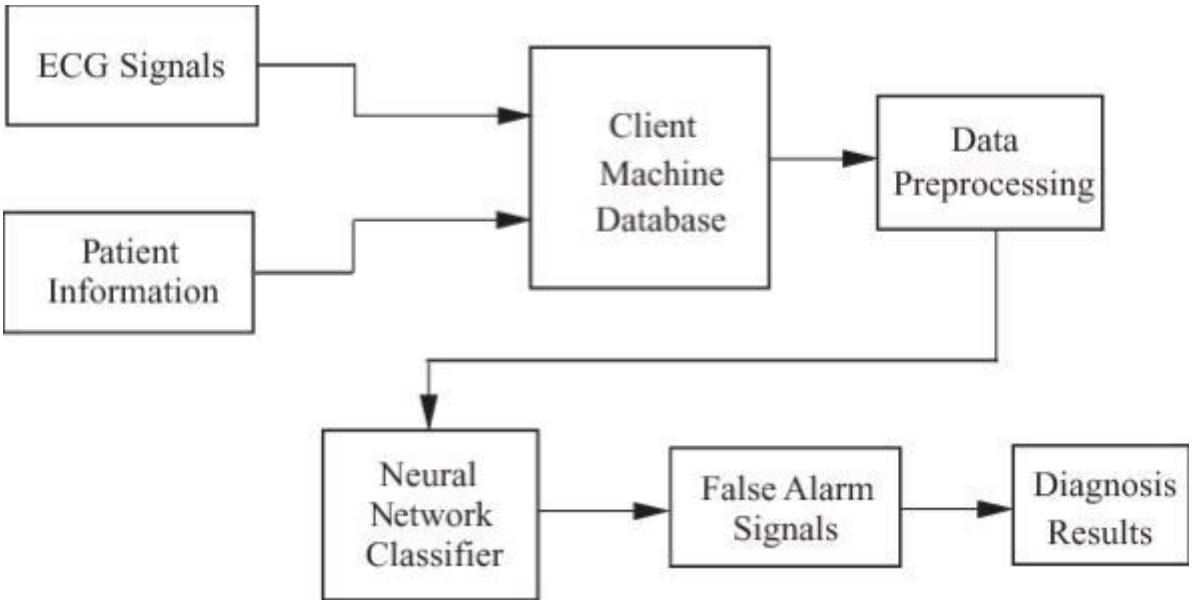
4. Since most of the data we have belong s to normal heart-rate so we need to augment the data using RST variation in order to avoid the biasness of the classifier.

5. Once we have finalized our dataset (after pre-processing and data augmentation), now we will apply deep learning concepts to classify the ECG signal as normal and abnormal.


Start
ECG Arrhythmia Classification
Create Database
Check collected data
1D
Concert it to 2D data
2D
Apply data Augmentation
Use RST Variation in order to avoid biasness
Apply pre-processing on dataset
Result
Abnormal heart rate
Suggest Treatment
Normal heart rate
Stop

USE CASE


ECG Machine
Monitor normal cardiac function
Trigger on abnormal event
Print out ECG history
Service functions
View Error Log
Upgrade Software
«actor» Physician
«actor» Service Rep
«actor» Remote Display
«actor» Patient
«actor» Chart Recorder


ECG Signals
Patient Information
Client Machine Database
Data Preprocessing
Neural Network Classifier
False Alarm Signals
Diagnosis Results

INTRODUCTION TO HCI

Human Computer Interaction (HCI) is a multidisciplinary practice that focuses on both the interaction between the user (humans) and the computer as well as the design of the computer interface. Originally, HCI focused primarily on computers, but since then has expanded to include almost all variations of information technology design after the emergence of personal computers in the late 1970s. Researchers realized that they had to expand the interaction with computers to everyone rather than to only people who were information technology professionals. It studies primarily on how people interact with computers and to what extent the users are able to interact with the computers. The goal is to have a successful interaction between both the computer and the user and to see which areas still need more development. HCI consists of three components: the user, the computer, and how these two works together.

Simply put, the goal of HCI is to create a user-friendly system that is also functional and safe. In order to achieve this goal, developers must be able to Empathize with the users and anticipate how they will use the technology, develop tools and techniques to enable best practices on building a sufficient system and produce an effective, efficient, and safe interaction.

Often times, developers are so worried about writing their codes that they forget to think about the user experience and visual design aspect of their product. This can lead to frustrated users with many pain points and stressed developers because they need to use more time and resources to fix the problem(s). At the end of the day, these products are created to simplify the everyday lives of people. Some examples of interactive products include cell phones, computers, coffee machines, ATM, the web, etc.

SAMPLE CODING

1. Importing the required libraries –

```
# Importing the required libraries
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
import biosppy # it contains the function to calculate the RPeaks which is very much required
import warnings
import matplotlib.pyplot as plt
import cv2
warnings.filterwarnings('ignore') # to suppress the warnings
```

2. Data Augmentation –

Since the data was highly imbalanced because we have majority of normal heart beat and minority of affected or non-normal heart beat so it we don't do data augmentation, the model will be biased.

```python
def cropping(image, filename,ytrain,g):

    #Left Top Crop
    crop = image[:46, :46]
    crop = cv2.resize(crop, (64, 64))
    cv2.imwrite(filename[:-4] + 'leftTop' + '.png', crop)
    ytrain.append(g)

    #Center Top Crop
    crop = image[:46, 16:50]
    crop = cv2.resize(crop, (64, 64))
    cv2.imwrite(filename[:-4] + 'centerTop' + '.png', crop)
    ytrain.append(g)

    #Right Top Crop
    crop = image[:46, 32:]
    crop = cv2.resize(crop, (64, 64))
    cv2.imwrite(filename[:-4] + 'rightTop' + '.png', crop)
    ytrain.append(g)

    #Left Center Crop
    crop = image[16:50, :50]
    crop = cv2.resize(crop, (64, 64))
    cv2.imwrite(filename[:-4] + 'leftCenter' + '.png', crop)
    ytrain.append(g)
```

```
#Center Center Crop
crop = image[16:50, 16:50]
crop = cv2.resize(crop, (64, 64))
cv2.imwrite(filename[:-4] + 'centerCenter' + '.png', crop)
ytrain.append(g)

#Right Center Crop
crop = image[16:50, 32:]
crop = cv2.resize(crop, (64, 64))
cv2.imwrite(filename[:-4] + 'rightCenter' + '.png', crop)
ytrain.append(g)

#Left Bottom Crop
crop = image[32:, :50]
crop = cv2.resize(crop, (64, 64))
cv2.imwrite(filename[:-4] + 'leftBottom' + '.png', crop)
ytrain.append(g)

#Center Bottom Crop
crop = image[32:, 16:50]
crop = cv2.resize(crop, (64, 64))
cv2.imwrite(filename[:-4] + 'centerBottom' + '.png', crop)
ytrain.append(g)

#Right Bottom Crop
crop = image[32:, 32:]
crop = cv2.resize(crop, (64, 64))
cv2.imwrite(filename[:-4] + 'rightBottom' + '.png', crop)
ytrain.append(g)
```

3. This is the main block where the following steps are done –

1) Reading of the csv file of the patient.

2) Now since the dataset was highly imbalanced, so we had to check how many minimum csv files should we concatenate so that imbalances are minimized and value that we got is 20 (from r0 to r20).

3) After concatenating those 20 files into a single csv, use Standard Scaler to normalize the data otherwise training will take time.

4) Normally we could have put this data in a machine learning model to train but let's do something different. We converted these 1D data to a 2D data (image).

5) To do the above thing, we used biosppy.signals.ecg.christov_segmenter() which helps in finding the RPeaks which helped in plotting the image.

6) Once image is found i did data augmentation (to minimize imbalance) and saved those images to be used in the CNN model.

```
import pandas as pd
w=100
h=(str(w))+'annotations.csv'
pks=[]
signals=[]
xtrain=[]
ytrain=[]
imp=[['N'],['L','A','V']] # we have normal and non normal heartbeat .Non noraml hearbeat can be
                          divided into(L-Left branch block beat, A-Atrial premature
                          contraction, V-Ventricular contraction)
gt=['N','NN']
dt=-1
rohancnt=0
data=pd.DataFrame()

while(w<120):# i was hit and trialing how many minimum files to concatenate to avoid data
                                                imbalancy other model will be biased
    h=(str(w))+"annotations.csv"
    r0=pd.read_csv(r"C:\Users\PUTUL SIDDHARTH\Desktop\ECG\ecg_files\txt_files"+'\\'+
                                                    (str(w))+"annotations.csv")
    r1=pd.read_csv(r"C:\Users\PUTUL SIDDHARTH\Desktop\ECG\ecg_files\txt_files"+'\\'+
                                            (str(w+1))+"annotations.csv")
    r2=pd.read_csv(r"C:\Users\PUTUL SIDDHARTH\Desktop\ECG\ecg_files\txt_files"+'\\'+
                                            (str(w+2))+"annotations.csv")
    r3=pd.read_csv(r"C:\Users\PUTUL SIDDHARTH\Desktop\ECG\ecg_files\txt_files"+'\\'+
                                            (str(w+3))+"annotations.csv")
    r4=pd.read_csv(r"C:\Users\PUTUL SIDDHARTH\Desktop\ECG\ecg_files\txt_files"+'\\'+
                                            (str(w+4))+"annotations.csv")
    r5=pd.read_csv(r"C:\Users\PUTUL SIDDHARTH\Desktop\ECG\ecg_files\txt_files"+'\\'+
                                            (str(w+5))+"annotations.csv")
    r6=pd.read_csv(r"C:\Users\PUTUL SIDDHARTH\Desktop\ECG\ecg_files\txt_files"+'\\'+
                                            (str(w+6))+"annotations.csv")
    r7=pd.read_csv(r"C:\Users\PUTUL SIDDHARTH\Desktop\ECG\ecg_files\txt_files"+'\\'+
                                            (str(w+7))+"annotations.csv")
    r8=pd.read_csv(r"C:\Users\PUTUL SIDDHARTH\Desktop\ECG\ecg_files\txt_files"+'\\'+
                                            (str(w+8))+"annotations.csv")
    r9=pd.read_csv(r"C:\Users\PUTUL SIDDHARTH\Desktop\ECG\ecg_files\txt_files"+'\\'+
                                            (str(w+9))+"annotations.csv")
    r10=pd.read_csv(r"C:\Users\PUTUL SIDDHARTH\Desktop\ECG\ecg_files\txt_files"+'\\'+
                                            (str(w+10))+"annotations.csv")
    r11=pd.read_csv(r"C:\Users\PUTUL SIDDHARTH\Desktop\ECG\ecg_files\txt_files"+'\\'+
                                            (str(w+11))+"annotations.csv")
    r12=pd.read_csv(r"C:\Users\PUTUL SIDDHARTH\Desktop\ECG\ecg_files\txt_files"+'\\'+
                                            (str(w+12))+"annotations.csv")
    r13=pd.read_csv(r"C:\Users\PUTUL SIDDHARTH\Desktop\ECG\ecg_files\txt_files"+'\\'+
                                            (str(w+13))+"annotations.csv")
    r14=pd.read_csv(r"C:\Users\PUTUL SIDDHARTH\Desktop\ECG\ecg_files\txt_files"+'\\'+
                                            (str(w+14))+"annotations.csv")
    r15=pd.read_csv(r"C:\Users\PUTUL SIDDHARTH\Desktop\ECG\ecg_files\txt_files"+'\\'+
                                            (str(w+15))+"annotations.csv")
    r16=pd.read_csv(r"C:\Users\PUTUL SIDDHARTH\Desktop\ECG\ecg_files\txt_files"+'\\'+
                                            (str(w+16))+"annotations.csv")
    r17=pd.read_csv(r"C:\Users\PUTUL SIDDHARTH\Desktop\ECG\ecg_files\txt_files"+'\\'+
                                            (str(w+17))+"annotations.csv")
    r18=pd.read_csv(r"C:\Users\PUTUL SIDDHARTH\Desktop\ECG\ecg_files\txt_files"+'\\'+
                                            (str(w+18))+"annotations.csv")
    r19=pd.read_csv(r"C:\Users\PUTUL SIDDHARTH\Desktop\ECG\ecg_files\txt_files"+'\\'+
                                            (str(w+19))+"annotations.csv")
    r20=pd.read_csv(r"C:\Users\PUTUL SIDDHARTH\Desktop\ECG\ecg_files\txt_files"+'\\'+
                                            (str(w+20))+"annotations.csv")
    print((str(w))+'annotations.csv')
    # concat all those files in one file
    r=pd.concat([r0,r1,r2,r3,r4,r5,r6,r7,r8,r9,r10,r11,r12,r13,r14,r15,r16,r17,r18,r19,r20],
                                                                        axis=0)
    rac=StandardScaler() # data normalization
    r.Sample=rac.fit_transform(r.Sample.values.reshape(-1, 1) )
```

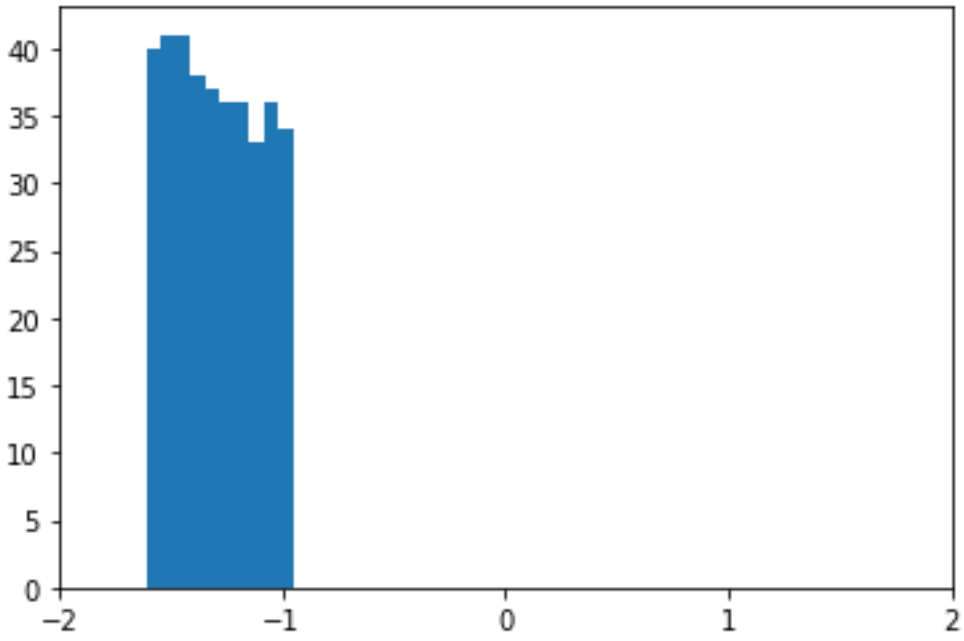
```
    for g in imp:
        for h in g:
            ft=r[(r.Type==h)]
            data=pd.concat([data,ft],axis=0)

        xtrain=data.Sample
        # To find the RPeaks of the each type of signal(N,NN)
        peaks =  biosppy.signals.ecg.christov_segmenter(signal=xtrain, sampling_rate = 200)[0]
        print(g,'peaks are')
        print(peaks)
        pks.append(peaks)

        dt=dt+1
        signals=[]
        b=0
        b1=[]
        count = 1
        for i in (peaks[1:-1]):
            # To convert those 1D Rpeaks to 2D RPeaks(image foramt)
            diff1 = abs(peaks[count - 1] - i)
            diff2 = abs(peaks[count + 1]- i)#
            x = peaks[count - 1] + diff1//2#
            y = peaks[count + 1] - diff2//2#
            signal = xtrain[x:y]
            print(i)
            print(signal)
            fig = plt.figure(frameon=False)
            #plt.figure()
            plt.hist(signal) # plotting the histogram figure of the RPeaks of the signal
            plt.xticks(np.arange(-2,3,1))
            filename = r'C:\Users\PUTUL
                        SIDDHARTH\Desktop\dspimg2'+'\\'+gt[dt]+'\\'+str(count)+'.png'
            fig.savefig(filename)
            # Data augmentation should always be done on gray scale images
            im_gray = cv2.imread(filename, cv2.IMREAD_GRAYSCALE)
            im_gray = cv2.resize(im_gray, (64, 64), interpolation = cv2.INTER_LANCZOS4)
            cv2.imwrite(filename, im_gray)
            cropping(im_gray, filename,ytrain,g)
            ytrain.append(g)
            rohancnt=rohancnt+1


            signals.append(signal)
            count += 1
    print(ytrain)
    w=w+20
```

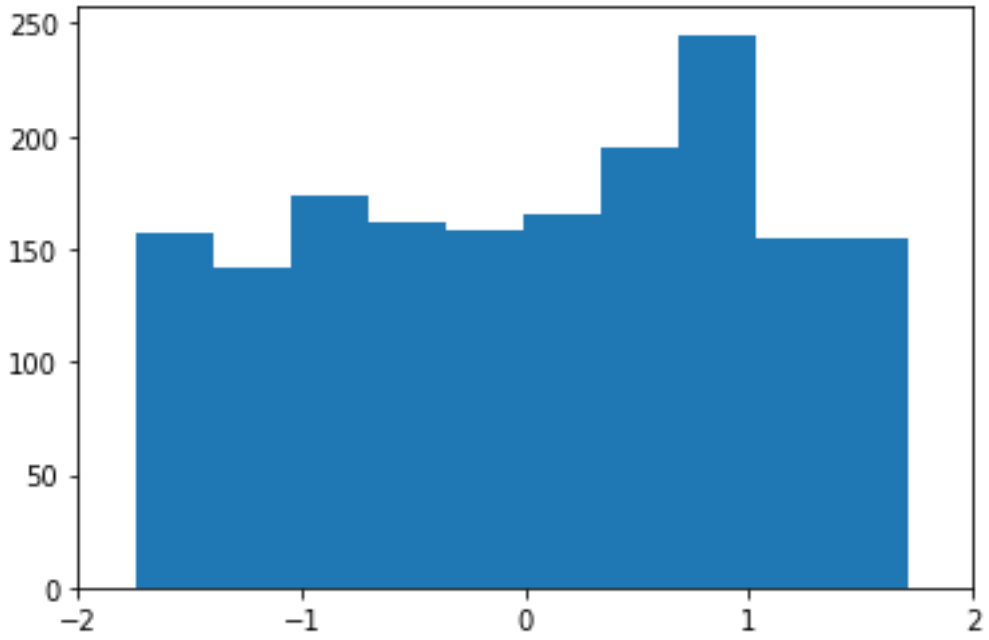
4. Here we have used VGG net Model to train our images using Keras –

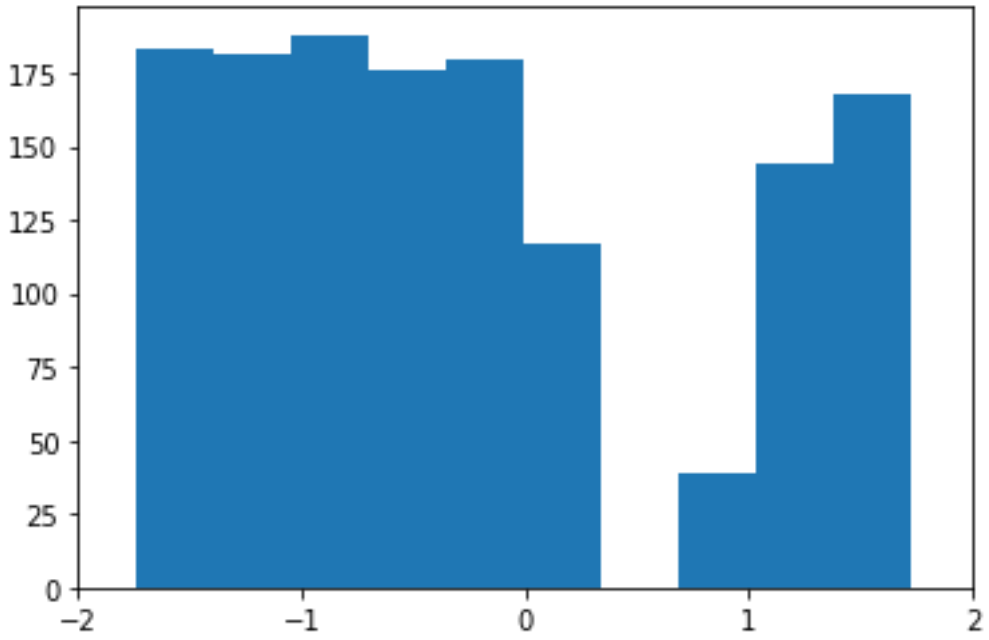
```
from keras.models import Sequential
from keras.layers import Dense,Dropout
import warnings
from keras.wrappers.scikit_learn import KerasClassifier
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import GridSearchCV
from keras.preprocessing.image import ImageDataGenerator
from keras.layers.convolutional import Conv2D
from keras.layers.convolutional import MaxPooling2D
from keras.layers.normalization import BatchNormalization
from keras.layers import Flatten

warnings.simplefilter('ignore')
model = Sequential()
model.add(Conv2D(64, (3,3),strides = (1,1), input_shape =
(64,64,3),kernel_initializer='glorot_uniform'))

#model.add(keras.layers.ELU())
model.add(BatchNormalization())
model.add(Conv2D(64, (3,3),strides = (1,1),kernel_initializer='glorot_uniform'))

#model.add(keras.layers.ELU())
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2), strides= (2,2)))
model.add(Conv2D(128, (3,3),strides = (1,1),kernel_initializer='glorot_uniform'))

#model.add(keras.layers.ELU())
model.add(BatchNormalization())
model.add(Conv2D(128, (3,3),strides = (1,1),kernel_initializer='glorot_uniform'))

#model.add(keras.layers.ELU())
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2), strides= (2,2)))
model.add(Conv2D(256, (3,3),strides = (1,1),kernel_initializer='glorot_uniform'))

#model.add(keras.layers.ELU())
model.add(BatchNormalization())
model.add(Conv2D(256, (3,3),strides = (1,1),kernel_initializer='glorot_uniform'))

#model.add(keras.layers.ELU())
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2), strides= (2,2)))
model.add(Flatten())
model.add(Dense(2048))

#model.add(keras.layers.ELU())
model.add(BatchNormalization())
model.add(Dropout(0.5))
model.add(Dense(1, activation='softmax'))

model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])
train_set=ImageDataGenerator(rescale=1./255,horizontal_flip=True,zoom_range=0.2,shear_range=0.2)
train=train_set.flow_from_directory(r'C:\Users\PUTUL SIDDHARTH\Desktop\dspimg1',target_size=
(64,64),batch_size=1,class_mode='binary')
model.fit_generator(train,steps_per_epoch=10,epochs=4)
```

OUTPUT

```
#OUTPUT -

100annotations.csv
['N'] peaks are
[   27  2205  2435  2949  3917  4413  4484  4494  4507  5954  8254 10337
 13464 14380 15887 17352 20616 21652 23567 25592 28353 29006 31104 32747]
2205
1128   -0.036057
1129   -0.034457
1130   -0.032926
1131   -0.031368
1132   -0.029847
          ...
76     -1.613892
77     -1.612133
78     -1.610447
79     -1.608767
80     -1.607098
Name: Sample, Length: 1204, dtype: float64
2435
81    -1.605418
82    -1.603716
83    -1.601941
84    -1.600186
85    -1.598570
         ...
         ...
         ...
['N'], ['N'], ['N'], ['N'], ['N'], ['L', 'A', 'V'],....
# more ...
```



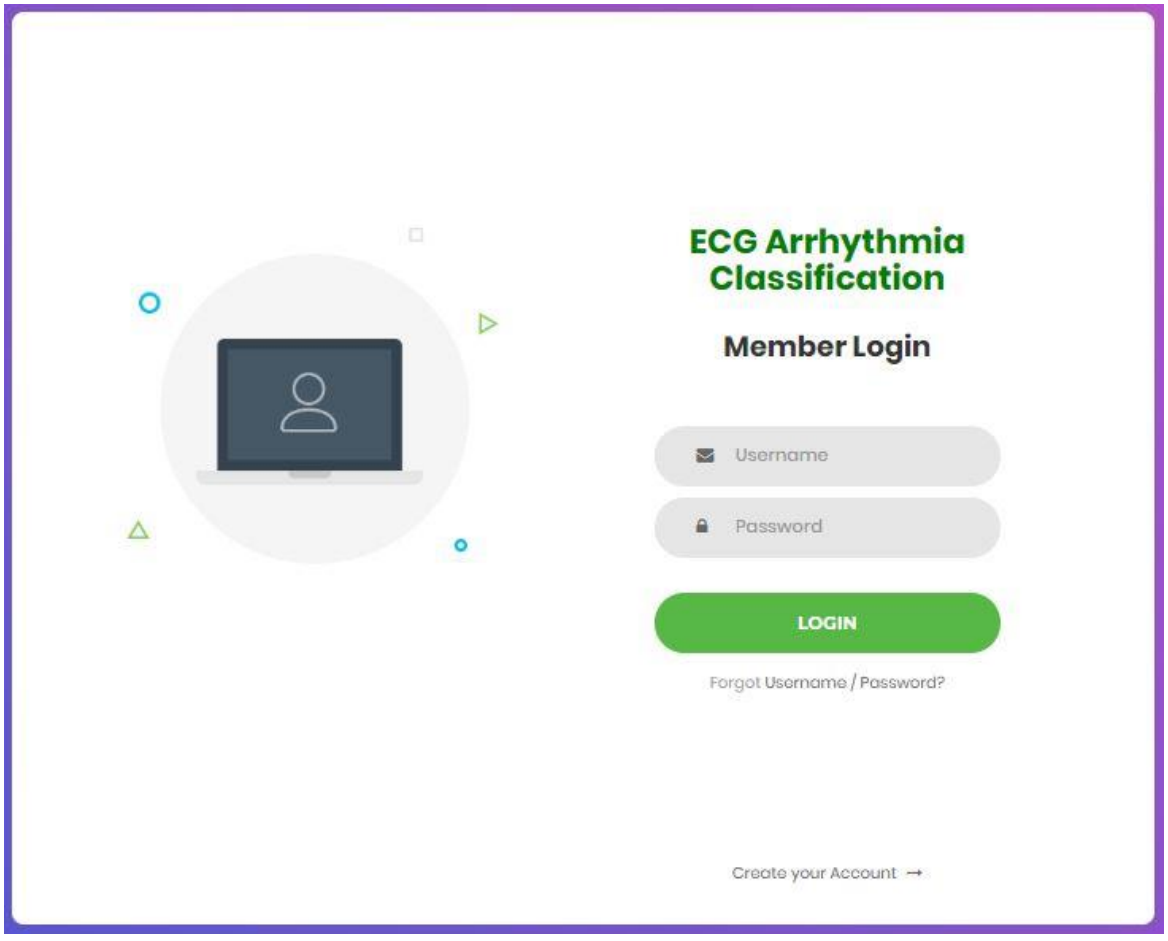

250
200
150
100
50
0
−2
−1
0
1
2


175
150
125
100
75
50
25
0
−2
−1
0
1
2

```
#OUTPUT -

Found 244 images belonging to 2 classes.
Epoch 1/4
10/10 [==============================] - 6s 583ms/step - loss: 1.5333 - accuracy: 0.9000
Epoch 2/4
10/10 [==============================] - 3s 340ms/step - loss: 4.6000 - accuracy: 0.7000
Epoch 3/4
10/10 [==============================] - 3s 339ms/step - loss: 3.0666 - accuracy: 0.8000
Epoch 4/4
10/10 [==============================] - 3s 339ms/step - loss: 1.5333 - accuracy: 0.9000
<keras.callbacks.callbacks.History at 0x19c39ca5088>
```

After doing 4 epochs on 64*64 size images with 10 steps per epoch we got a decent accuracy of 90%.

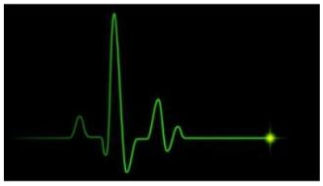
UI SCREENSHOT



ECG ARRHYTHMIA CLASSIFICATION

Introduction

According to the World Health Organization (WHO), cardiovascular diseases (CVDs) are the number one cause of death today. Arrhythmia is a representative type of CVD that refers to any irregular change from normal heart rhythms. There are several types of arrhythmia including atrial fibrillation, premature contraction, ventricular fibrillation, and tachycardia. Although single arrhythmia heartbeat may not have a serious impact on life, continuous arrhythmia beats can result in fatal circumstances. Arrhythmia is an important group of diseases in cardiovascular disease. Arrhythmia can occur on its own or with other cardiovascular diseases. The diagnosis of arrhythmia mainly depends on the ECG (electrocardiogram). ECG (electrocardiogram) is an important modern medical tool that records the process of cardiac excitability, transmission, and recovery.



Upload File

ECG-ARRHYTHMIA-CLASSIFICATION

Full Name :

Address :

Gender : Male Female

State : **City :**

DOB : **Pin Code :**

Email ID :

Upload your file :

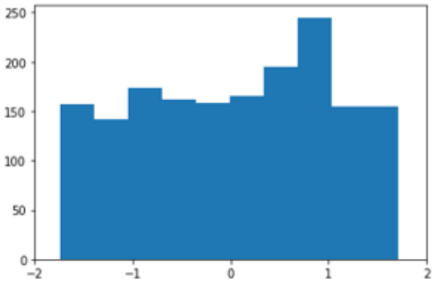
Choose File No file chosen

Predict

ECG Arrhythmia Classification Report

Normal Report

Normal ECG. A normal ECG is illustrated above. Note that the heart is beating in a regular sinus rhythm between 60 - 100 beats per minute (specifically 82 bpm). All the important intervals on this recording are within normal ranges.

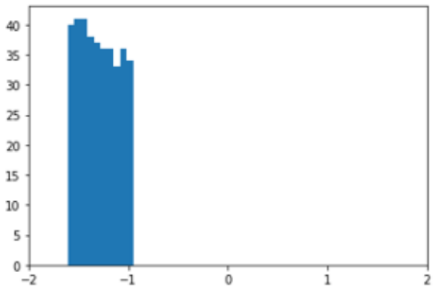


Logout

ECG Arrhythmia Classification Report

Abnormal Report

Many abnormal ECG's are rooted in problems with your heart. Electrolyte imbalances or medication side effects can cause your heart to beat too quickly (tachycardia). Heart defects or abnormalities in your heart's shape or size, blood flow or heart rate will result in an abnormal ECG as well.



Logout

APPLICATION

1. Our project can be used as a fast, efficient and a cost effective way for arrhythmia detection and preliminary detection of cardiovascular diseases.

2. Our project can aid doctors to diagnose tests on the go.

3. It can be deployed in rural and economically backward regions and it can be operated by any adult irrespective of having a medical background.

CONCLUSION

A new technique for automatic heartbeat classification of two types of arrhythmia was presented. An improved hybrid feature representation of heartbeat segments was used. This report highlights the transform techniques used for accurate detection of R-peak and QRS complex with less error rate. Different methodology were proposed for automatic classification of cardiac arrhythmias standard using ECG signals. We have focused mainly the detection and classification ECG beats automatically by analyzing ECG signals

REFERENCES

https://www.sciencedirect.com/science/article/pii/S0169260715003314#:~:text=A%20full%20automatic%20system%20for,(4)%20learning%2Fclassification.

https://www.hindawi.com/journals/cmmm/2018/1380348/

https://ieeexplore.ieee.org/abstract/document/8759878

https://www.sciencedirect.com/science/article/pii/S1877050918307774