PROJECT REPORT

PSG INSTITUTE OF TECHNOLOGY AND APPLIED RESEARCH, NEELAMBUR

Classification of Arrhythmia by Using Deep Learning with 2-D ECG Spectral Image Representation

DEPARTMENT OF ELECTRONICS AND COMMUNICATION DEPARTMENT

FINAL YEAR

BATCH 2019-2023

SUBMITTED BY:

Team ID: PNT2022TMID43406

SWEDHA R	**-715519106054**
SOWMYA S	**-715519106049**
SARVESH VISHWA	**-715519106043**
VEDHA R	**-715519106057**

ABSTRACT

Arrythmia refers to irregular beating of heart and cause be undetected and prone to risk serious cases. Detecting arrhythmia is currently done by monitoring ECG patterns closely for a long period of time can be very time consuming and complicated to detect in some cases due to human errors.

Detecting Arrythmia can be done by using Deep Learning which develop patterns to predict the slightest change in heartbeat. Convolutional Neural Network is a type of Deep Learning that works well with image classification models. ECG image can be converted to 2D for better analysis.

To make this feature possible to all users/patients, a user-friendly website is created to access the model allowing them to easily upload their images and predict their heart rhythm.

This project is created with the fact that such a feature could help save human lives through early detection of arrhythmia that can prevent them for Cardio Vascular Diseases (CVDs) which is the leading cause of death according to WHO.

Project Report Format

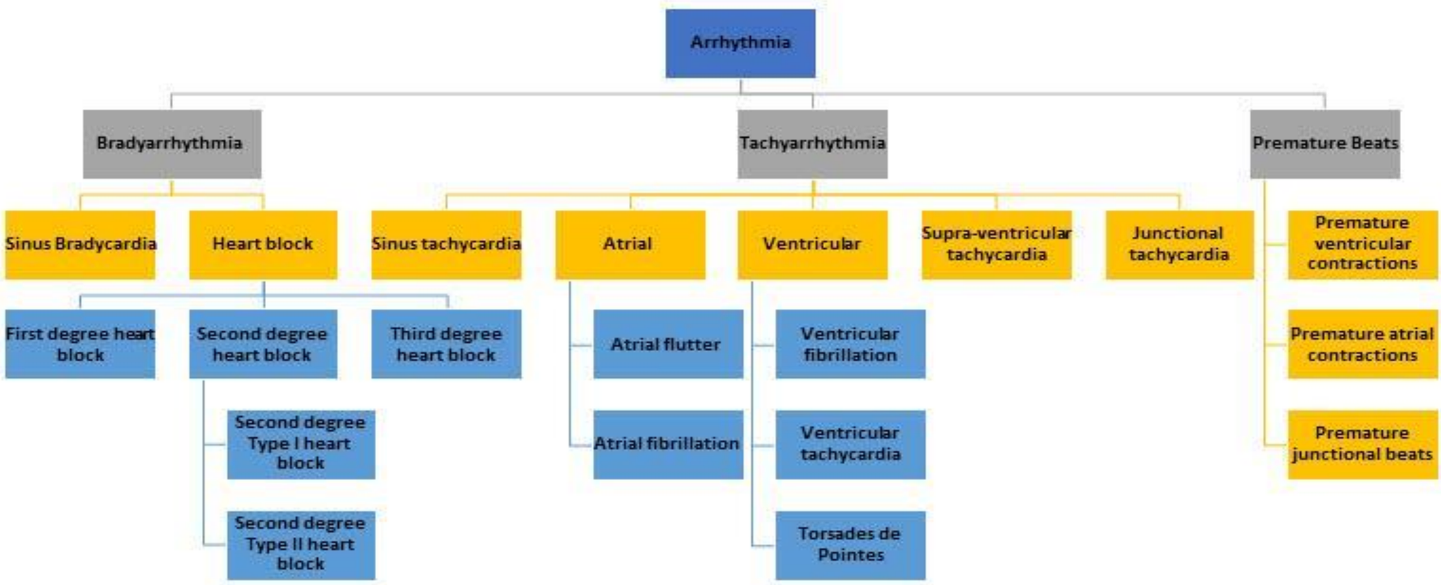
1. **INTRODUCTION**
   1.1 Project Overview
   1.2 Purpose
2. **LITERATURE SURVEY**
   2.1 Existing problem
   2.2 References
   2.3 Problem Statement Definition
3. **IDEATION & PROPOSED SOLUTION**
   3.1 Empathy Map Canvas
   3.2 Ideation & Brainstorming
   3.3 Proposed Solution
   3.4 Problem Solution fit
4. **REQUIREMENT ANALYSIS**
   4.1 Functional requirement
   4.2 Non-Functional requirements
5. **PROJECT DESIGN**
   5.1 Data Flow Diagrams
   5.2 Solution & Technical Architecture
   5.3 User Stories
6. **PROJECT PLANNING & SCHEDULING**
   6.1 Sprint Planning & Estimation
   6.2 Sprint Delivery Schedule
   6.3 Reports from JIRA
7. **CODING & SOLUTIONING (Explain the features added in the project along with code)**
   7.1 Feature 1
   7.2 Feature 2
   7.3 Database Schema (if Applicable)
8. **TESTING**
   8.1 Test Cases
   8.2 User Acceptance Testing
9. **RESULTS**
   9.1 Performance Metrics
10. **ADVANTAGES & DISADVANTAGES**
11. **CONCLUSION**
12. **FUTURE SCOPE**
13. **APPENDIX**
    Source Code
    GitHub & Project Demo Link

CHAPTER 1- INTRODUCTION

1.1 PROJECT OVERVIEW

'Heart Arrhythmia' is the condition of irregular heart rhythms. There are variations in the heartbeat patterns. When electrical signals that were meant to coordinate with the heartbeats falter, this condition takes place. The heart could beat faster, or slower, or any other form of irregularity is usually noticeable.

Continuous arrhythmia beats can lead to deadly situations, even though a single arrhythmia heartbeat may not have a serious influence on life. In this study, we develop an efficient electrocardiogram (ECG) arrhythmia classification method using convolutional neural networks (CNNs). We classify ECG into seven categories, one of which is normal and the other six of which are various types of arrhythmias, using deep two-dimensional CNNs with grayscale ECG images. The user selects the image that will be categorised in the web application we are building. The image is fed into the trained model, and the mentioned class is shown on the webpage.

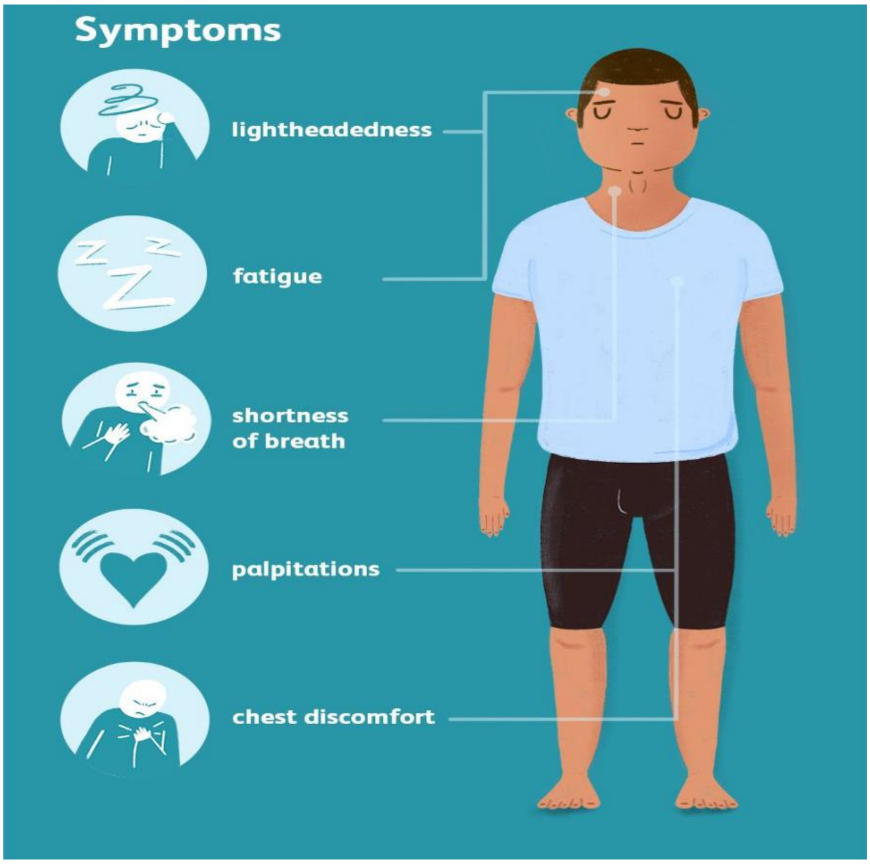


Source: Types of Arrhythmia

1.2 PURPOSE

The World Health Organization (WHO) states that cardiovascular diseases (CVDs) are currently the leading cause of death. Over 17.7 million individuals worldwide, or roughly 31% of all fatalities in 2017, perished from CVDs, and more than 75% of these deaths took place in low- and middle-income nations. Any unusual deviation from the regular cardiac beats is referred to as an arrhythmia, which serves as a representative form of CVD. Atrial fibrillation, premature contraction, ventricular fibrillation, and tachycardia are only a few examples of arrhythmia.

Pointing out the early symptoms of heart disease and ensuring they receive proper treatment can prevent premature deaths. Access to health technologies globally can be beneficial to society. Arrhythmia, a type of CVD, is an irregular pattern of heart beat and is generally complicated to recognise in normal ECG signals.



CHAPTER 2: LITERATURE SURVEY

2.1 EXISTING PROBLEM

The World Health Organization (WHO) recognises cardiovascular diseases (CVDs) as a leading cause of death worldwide. The statistics realize almost 32% of all deaths globally are CVDs. The top risk factors for CVDs are tobacco usage, obesity, unhealthy diets and lifestyle choices, and alcohol misuse.

'Heart Arrhythmia' is the condition of irregular heart rhythms. There are variations in the heartbeat patterns. When electrical signals that were meant to coordinate with the heartbeats falter, this condition takes place. The heart could beat faster, slower, or any other form of noticeable irregularity.

2.2 REFERENCES TABLE

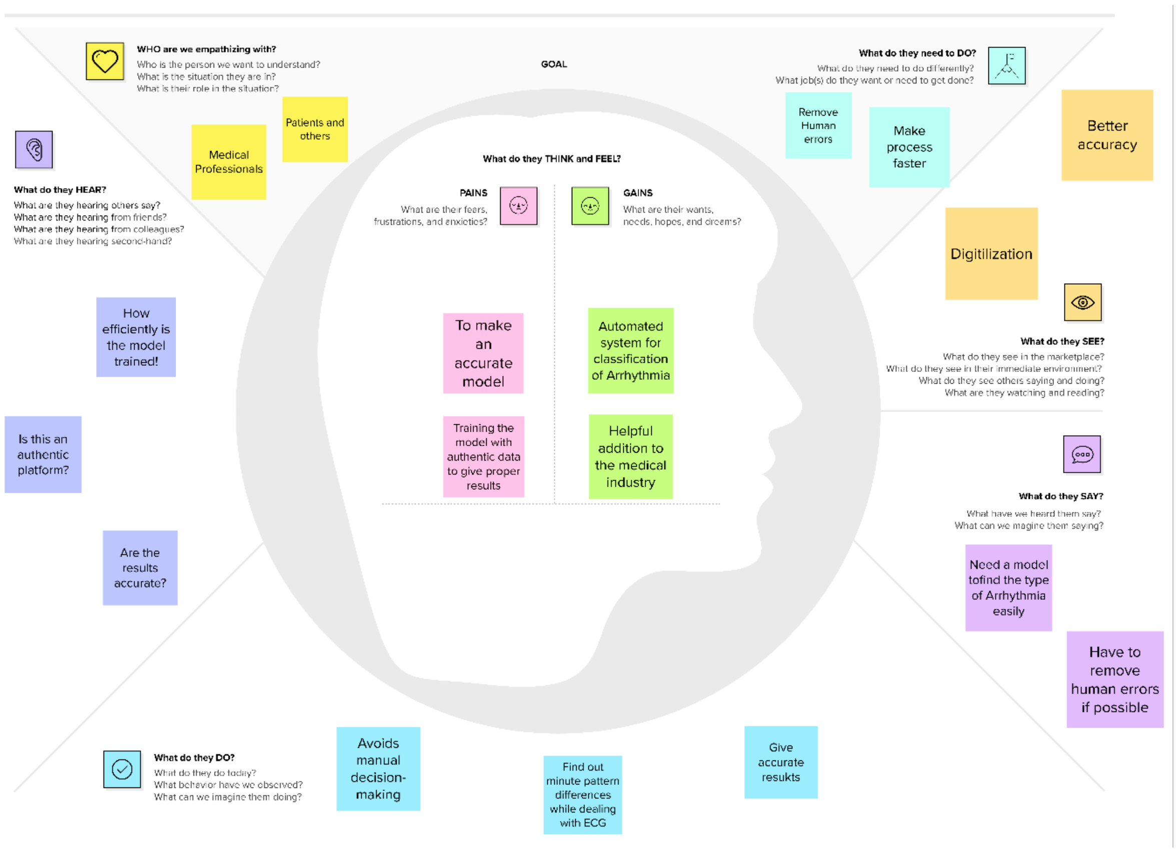
Year	Title	Link	Inference
2020	Classification of Arrhythmia by using Deep Learning with 2-D ECG Spectral Image Representation	Reference Paper 1	One Dimensional ECG time series signals are transformed into 2-D Spectrograms through STFT and fed into a model consisting of four convolutional layers and four pooling layers and reaching the state of art average classification accuracy of 99.11%.
2020	ECG Biometrics Using Deep Learning and Relative Score Threshold Classification	Reference Paper 2	This work proposes two architectures to improve current results in both identification and authentication by using Temporal Convolutional Network and Recurrent Neural Network. It has received an accuracy of almost 96% with an Equal Error Rate of 0.1%.
2019	ECG Arrhythmia Classification By Using Convolutional Neural Network And Spectrogram	Reference Paper 3	Two schools of approach were taken to classify at least 7 types of arrhythmias: One, by time series signal classification where the typical feature-extraction technique is not used, and two, by ECG spectrogram method where STFT (short term Fourier transform) is used to classify the kinds of heartbeat patterns. the 1-D signal was analysed in 4-D domains and around 95% accuracy and sensitivity rates were achieved, especially with the PVC (premature ventricular contractions) type.
2019	ECG Arrhythmia Classification Using STFT-based Spectrogram and Convolutional Neural Network	Reference Paper 4	ECG arrhythmia classification method based on deep learning techniques. ECG signals belonging to five different types were obtained from the MIT-BIH arrhythmia database. The ECG arrhythmia classification experimental results have successfully validated that the proposed 2D CNN can achieve better classification accuracy without manual pre-processing of the ECG signals such as noise filtering, feature extraction, and feature reduction.
2019	Arrhythmia Classification on ECG Using Deep Learning	Reference Paper 5	The proposed system compares various activation function by varying the number of epochs and the result is obtained where ELU function has an accuracy of 93.6 and with a loss of 0.2

2.3 PROBLEM STATEMENT DEFINITION

The current method used regarding Arrhythmias is ECG pattern reading techniques. This is a cumbersome process because it involves separately analysing each heartbeat and every pattern of the ECG records given by the Holter device placed over or around the heart. The procedure being manual, has restricted speed, making the entire process take hours or even days, with the added risk of human errors. This is why it is imperative to use computational techniques over the simple and hefty process of ECG record analysis.

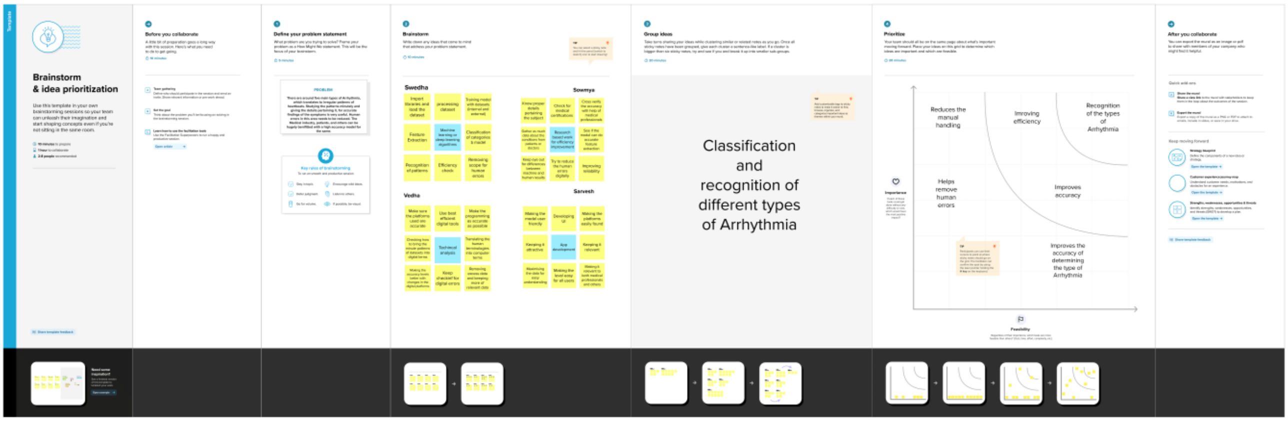
CHAPTER 3: IDEATION AND PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



The empathy map gives us a pictorial representation of the challenges worked on throughout the course of the project and the results received at completion. It also gives a view of the planning process that goes behind the steps taken.

3.2 IDEATION AND BRAINSTORMING



This works on sorting the planning process into an organized way. It gives priority levels to tasks and throws an image of what the main goals of the project are.

3.3 PROPOSED SOLUTION

The project we are working on, "Classification of Arrhythmias using 2-D ECG signals with Deep Learning" involves the two-dimensional convolution of ECG signals for pattern dissemination and the implementation of a CNN deep-learning model for prediction. The proposed model predicts Arrhythmia in images with a high accuracy rate of nearly 96%. The early detection of Arrhythmia gives a better understanding of disease causes, initiates therapeutic interventions, and enables the development of appropriate treatments.

The disadvantages are that the model does not aid in identifying the different stages of Arrhythmia diseases, or motor symptoms.

We can see that with advancing technologies, deep learning has made an effective impact on biomedicine. The industry benefits from the integration of vast datasets and the capability of prediction with varying levels of success. Electrocardiograms are painless methods of monitoring heart conditions and concerning electrocardiograms, deep learning is an artificial intelligence tool that's been entwined together for pattern prediction techniques. This is a big step towards the reduction of human errors in healthcare.

The current method used regarding Arrhythmias is ECG pattern reading techniques. This is a cumbersome process because it involves separately analysing each heartbeat and every pattern of the ECG records given by the Holter device placed over or around the heart. Being manual restricts the speed, making the entire process take hours or even days, with the added risk of human errors. This is why it is imperative to use computational techniques over the simple and hefty process of ECG record analysis. One such computational technique is this project we are dealing with. Its main goals refer to reducing human errors, increasing efficiency, and thus guaranteeing better success rates.

S.No.	Parameter	Description
1.	Problem Statement (Problem to be solved)	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 arrhythmias including atrial fibrillation, premature contraction, ventricular fibrillation, and tachycardia.
2.	Idea / Solution description	An "ambulatory electrocardiogram" or an ECG) about the size of a postcard or digital camera that the patient will be using for 1 to 2 days, or up to 2 weeks. The test measures the movement of electrical signals or waves through the heart. These signals tell the heart to contract (squeeze) and pump blood. The patient will have electrodes taped to your skin. It's painless, although some people have mild skin irritation from the tape used to attach the electrodes to the chest. They can do everything but shower or bathe while wearing the electrodes. After the test period, patient will go back to see your doctor. They will be downloading the information.

3.4 PROBLEM SOLUTION FIT

1.CUSTOMER SEGMENTS	2.JOBS-TO-BE-DONE/PROBLEMS	3.TRIGGERS	4.EMOTIONS: BEFORE/AFTER	5.AVAILABLE SOLUTIONS
Physicians use it to detect arrhythmias and in the middle-aged population.	Not useful for identifying the different stages of Arrhythmia disease. Not useful in monitoring motor symptoms	To detect the heart diseases quickly and efficiently.	Before: Confused, unsure, pain After: Fear, relief, sure	The 12-lead ECG remains the backbone of arrhythmia diagnosis, however, single-lead ECG technology can be incorporated into compact wearable devices. In this proposed model, PPG-identified arrhythmias signal the device to prompt users to perform a single-lead ECG through the same device to confirm an abnormal rhythm.
6.CUSTOMER CONSTRAINTS	**7.BEHAVIOUR**	**8.CHANNELS OF BEHAVIOUR**	**9.PROBLEM ROOT CAUSE**	**10.YOUR SOLUTION**
Lack of affordable and hassle-free technology.	Leads to panic and easy detection can prevent that, earlier diagnosis.	Patients detect arrythmia by running the model and lives are saved.	Unreliable source of detection and going unnoticed.	Building a reliable technology that can address all customer needs and provide long lasting solutions.

CHAPTER 4: REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENTS

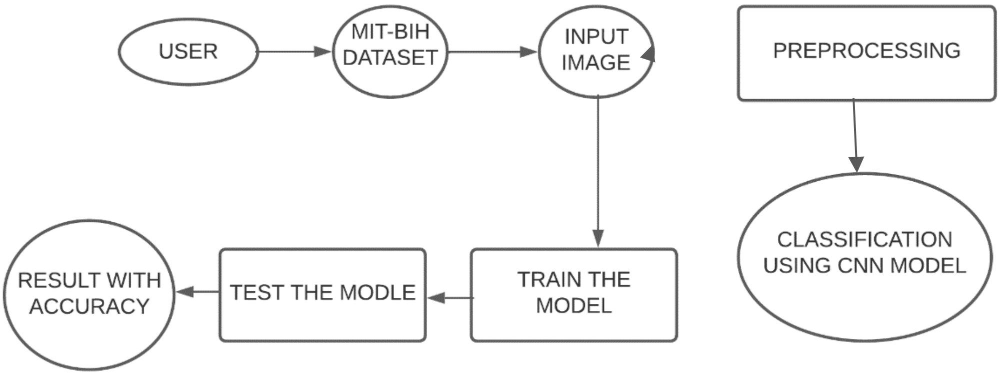
FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	Website Introduction	It encloses the code, graphics, and other learners' information about cardiovascular diseases and Arrhythmia classification.
FR-2	Image selection	Allows users to choose or upload image files from any system being used.
FR-3	Image prediction	Application gives you the classification of Arrhythmia based on the filtering of the given images.
FR-4	Arrhythmia classification model	Tensor Flow
FR-5	MIT-BIH Arrhythmia database	4000 long-term recordings from Beth Israel Hospital. We can also find datasets in kaggle.com, data.gov, UCI machine learning repository, etc.

4.2 NON FUNCTIONAL REQUIREMENTS

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	It should be comfortably usable and easy to understand by all.
NFR-2	**Reliability**	Deep learning and CNN models can be used behind the feature extraction technique. Other tools like Short term Fourier transform (STFT), and ECG spectrogram will deem useful for heartbeat pattern classification. The signals can be analyzed in 4-D domains for the betterment of efficiency.
NFR-3	**Performance**	The minute changes in pattern differences during feature extraction should be carefully noted. The arrhythmia-type classification should be accurate. Also, the delay in giving the desired results should be minimum.
NFR-4	**Availability**	Access is open to medical professionals mainly, but also to patients or others who could understand the fundamentals.
NFR-5	**Scalability**	Should be able to distinguish the three types and all their subtypes of Arrhythmia.

CHAPTER 5: PROJECT DESIGN

5.1 DATA FLOW DIAGRAM



The following data flow explains about the journey of the ECG signal in our proposed project.

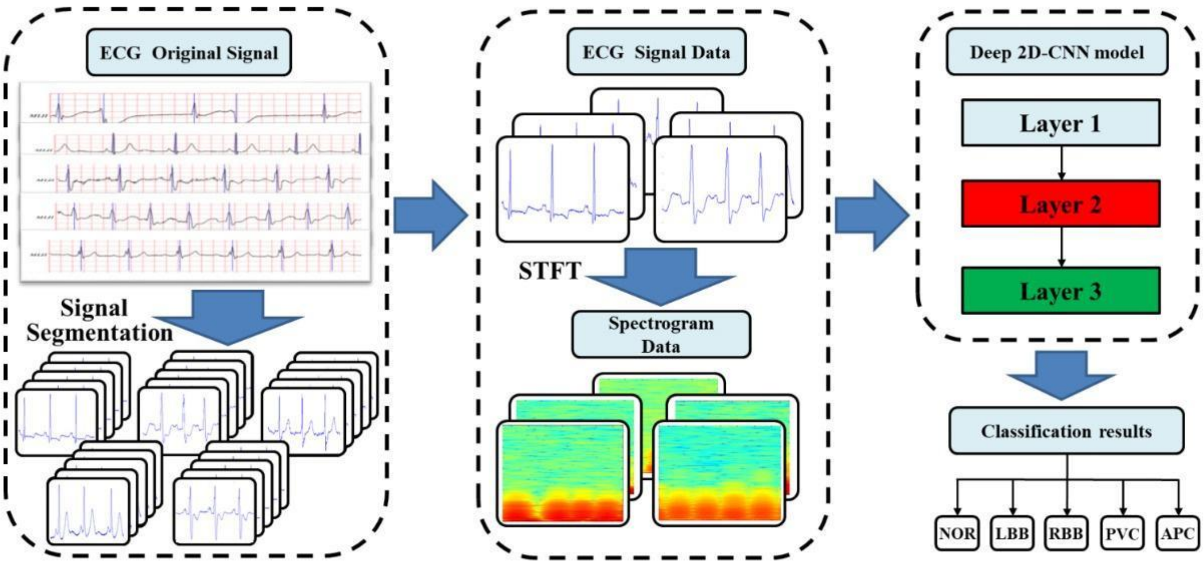


FIGURE 1. Overall procedures in ECG arrhythmia classification based on proposed 2D-CNN

5.2 SOLUTION AND TECHNICAL ARCHITECTURE

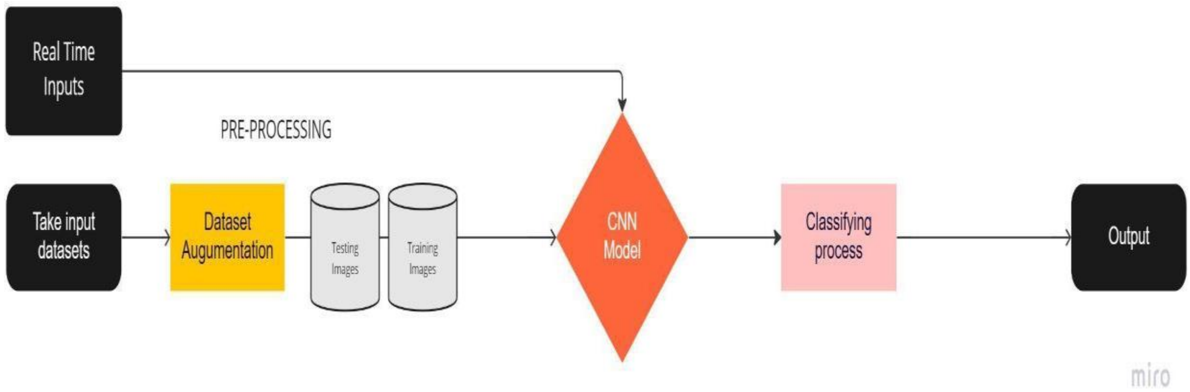


Table-1: Components and Technologies:

S.No	Components	Description	Technology
1.	Application Logic 1	Logic for a process in the application	Python, Flask
2.	Application Logic 2	Logic for a function in the application	IBM Watson service
3.	Application Logic 3	Logic for a process in the application	IBM Watson service
4.	Database	Datatype and other configurations	Numpy (Convolution)
5.	Cloud Database	Database service on Cloud	IBM Cloudant, etc
6.	External API	External API used in application	IBM API
7.	Machine Learning Model	Purpose of Machine Learning	Recognition model and such
8.	Infrastructure	Application Development	Local, cloud

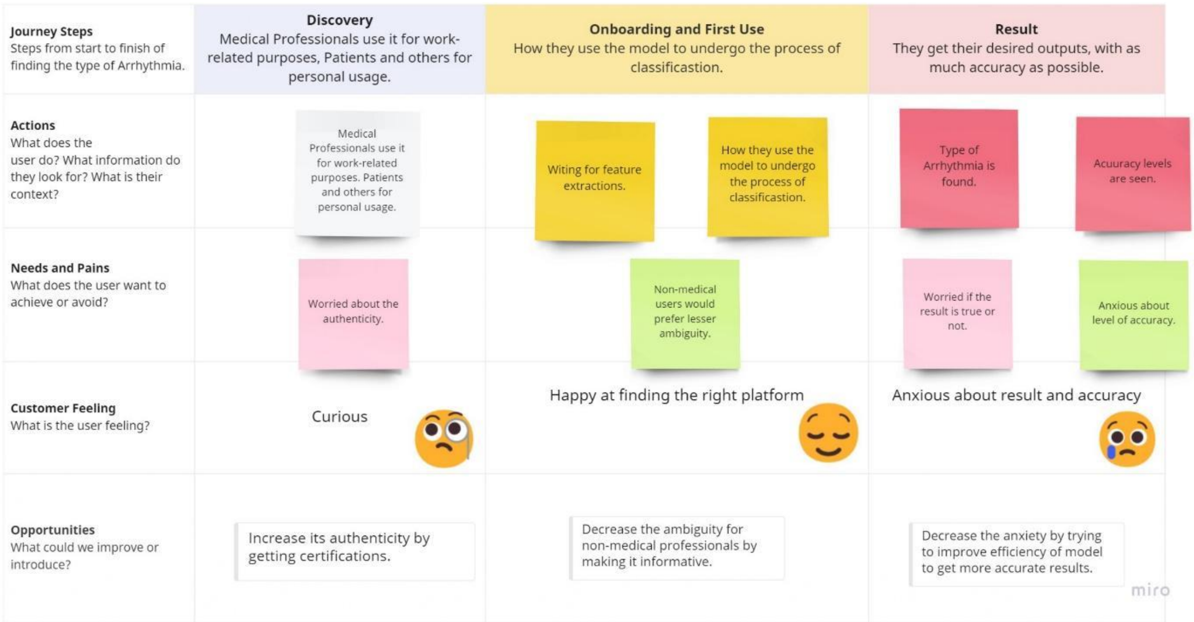
Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	Deep learning framework trains on provided datasets and gives out predictive results with accuracy and precision.	TensorFlow
2.	Security Implementations	System should contain data regarding health conditions and must be able to take images uploaded to process them.	Flask
3.	Scalable Architecture	The system must be able to handle different types of images and must figure out all the conditions accurately.	Data Augmentation-Keras
4.	Availability	There should be open information about the Arrhythmia types for anyone who wants to access it.	Flask
5.	Performance	Should reduce human errors.	CNN

5.3 USER STORIES

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer/Patient (Web)	Website Introduction	USN-1	As a user, I would be to enter the application and read about the information of cardiovascular diseases and the effects of undetected arrythmia	I can access dashboard	Low	Sprint-1
Customer/Patient (Web)	Website Introduction	USN-2	As a user, I would be able to identify different types of arrythmia in the application.	I can access dashboard	Low	Sprint-1
Customer	Image selection	USN-3	As a user, I can choose image files from my system and give as input into the application.	I can access website and choose files from my system	Medium	Sprint-2
Patient	Prediction	USN-4	As a user, I can find out about the condition of my heart beat as Left Bundle Branch Block	I can access the model and predict my arrythmia type.	High	Sprint-3
Patient	Prediction	USN-5	As a user, I can find out about the condition of my heart beat as Normal	I can access the model and predict my arrythmia type.	High	Sprint-3
Patient	Prediction	USN-6	As a user, I can find out about the condition of my heart beat as Premature Atrial Contraction	I can access the model and predict my arrythmia type.	High	Sprint-3
Patient	Prediction	USN-7	As a user, I can find out about the condition of my heart beat as Premature Ventricular Contraction	I can access the model and predict my arrythmia type.	High	Sprint-3
Patient	Prediction	USN-8	As a user, I can find out about the condition of my heart beat as Right Bundle Branch Block	I can access the model and predict my arrythmia type.	High	Sprint-3
Patient	Prediction	USN-9	As a user, I can find out about the condition of my heart beat as Ventricular Fibrillation	I can access the model and predict my arrythmia type.	High	Sprint-3
Doctor/Patient	Outcome	USN-10	As a user, I can find the results and preventive methods of my condition	I can access dashboard	Medium	Sprint-4

USER JOURNEY MAP



CHAPTER 6: PROJECT PLANNING AND SCHEDULING

6.1 SPRINT PLANNING AND ESTIMATION

Sprint	Functional Requirement (Epic)	User Story Number	User Story / Task	Story Points	Priority	Team Members
Sprint-1	Website Introduction	USN-1	As a user, I would be to enter the application and read about the information of cardiovascular diseases and the effects of undetected arrythmia	1	Low	Vedha, Sarvesh
Sprint-1	Website Introduction	USN-2	As a user, I would be able to identify different types of arrythmia in the application.	1	Low	Sowmya
Sprint-2	Image selection	USN-3	As a user, I can choose image files from my system and give as input into the application.	2	Low	Sarvesh
Sprint-3	Prediction	USN-4	As a user, I can find out about the condition of my heart beat as Left Bundle Branch Block	3	High	Swedha
Sprint-3	Prediction	USN-5	As a user, I can find out about the condition of my heart beat as Normal	5	High	Swedha
Sprint-3	Prediction	USN-6	As a user, I can find out about the condition of my heart beat as Premature Atrial Contraction	5	High	Sarvesh
Sprint-3	Prediction	USN-7	As a user, I can find out about the condition of my heart beat as Premature Ventricular Contraction	5	High	Sowmya
Sprint-3	Prediction	USN-8	As a user, I can find out about the condition of my heart beat as Right Bundle Branch Block	5	High	Swedha
Sprint-3	Prediction	USN-9	As a user, I can find out about the condition of my heart beat as Ventricular Fibrillation	5	Medium	Swedha
Sprint-4	Outcome	USN-10	As a user, I can find the results and preventive methods of my condition	3	Medium	Sowmya

6.2 SPRINT DELIVERY SCHEDULE

Sprint	Total Story Points	Duration	Sprint Start Date	Sprint End Date (Planned)	Story Points Completed (as on Planned End Date)	Sprint Release Date (Actual)
Sprint-1	20	7 Days	24 Oct 2022	31 Oct 2022	2	29 Oct 2022
Sprint-2	20	3 Days	31 Oct 2022	03 Nov 2022	2	29 Oct 2022
Sprint-3	20	8 Days	03 Nov 2022	12 Nov 2022	30	29 Oct 2022
Sprint-4	20	6 Days	14 Nov 2022	19 Nov 2022	3	29 Oct 2022

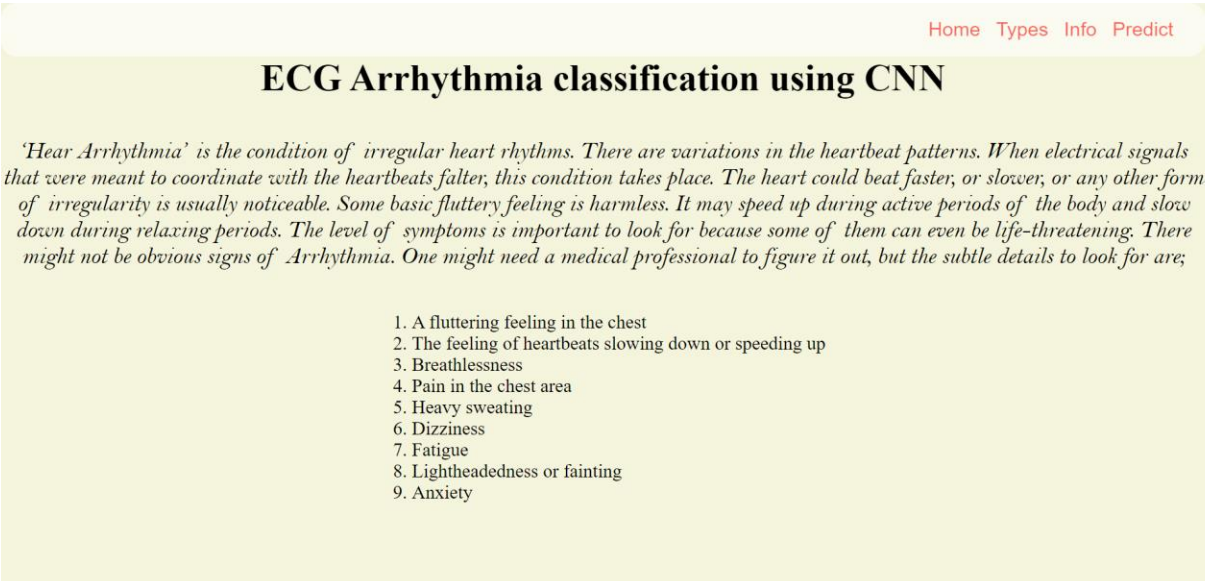
CHAPTER 7: CODING AND SOLUTIONING

Now, an API framework is constructed using flask application in VSC IDE with a python interpreter 3.9 version.

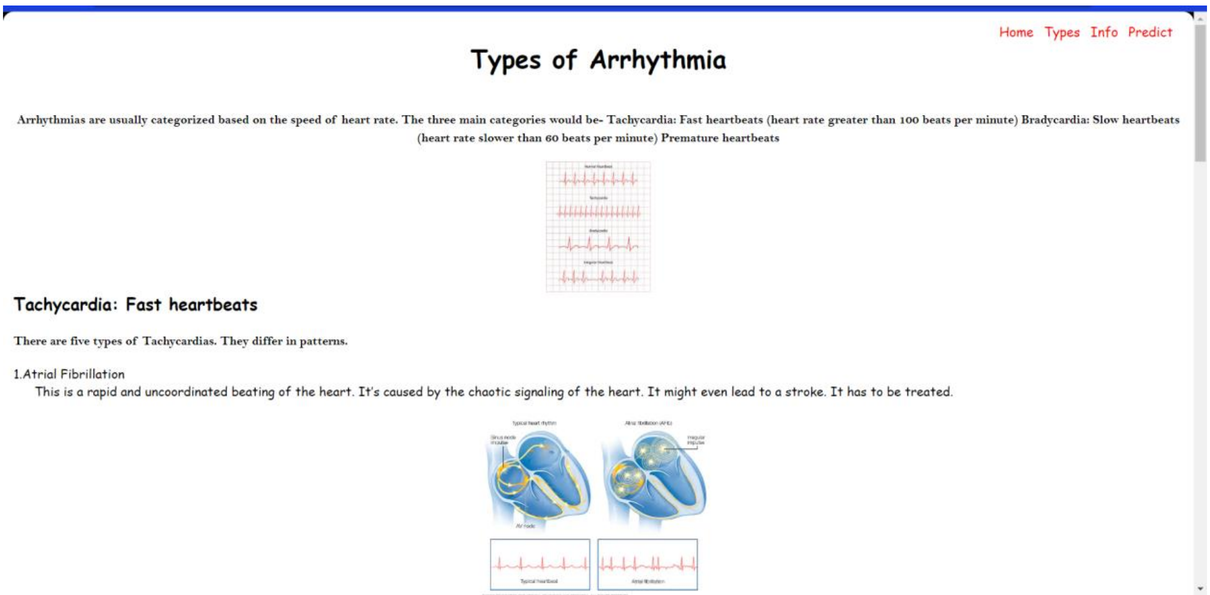
HTML Files are created to build the website to run the model in real time application.

HTML FILES

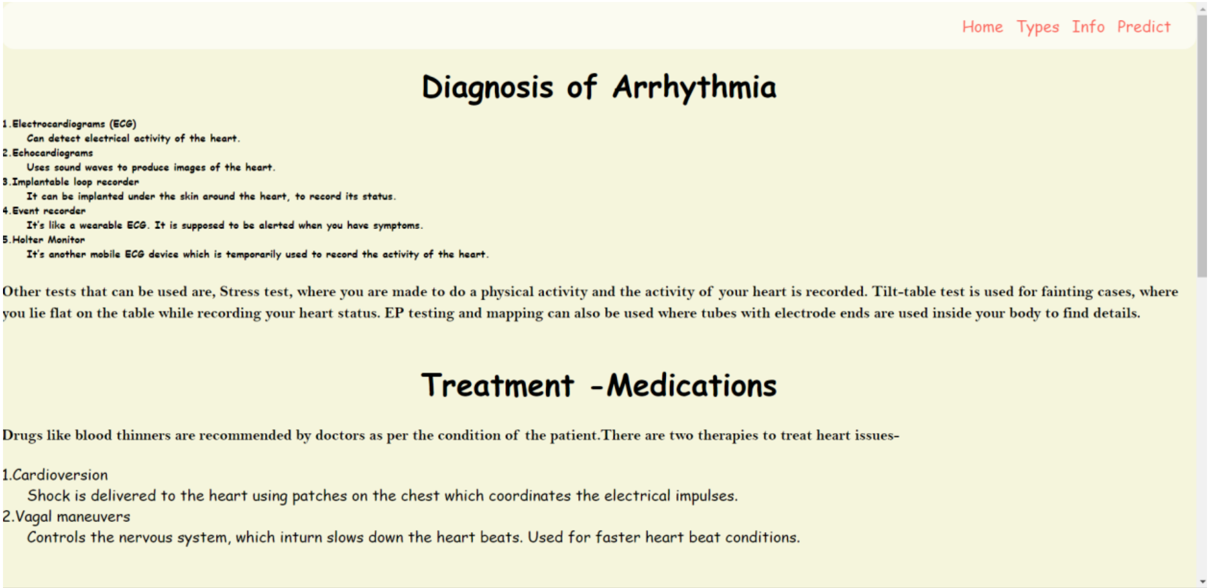
1) The Home Page:



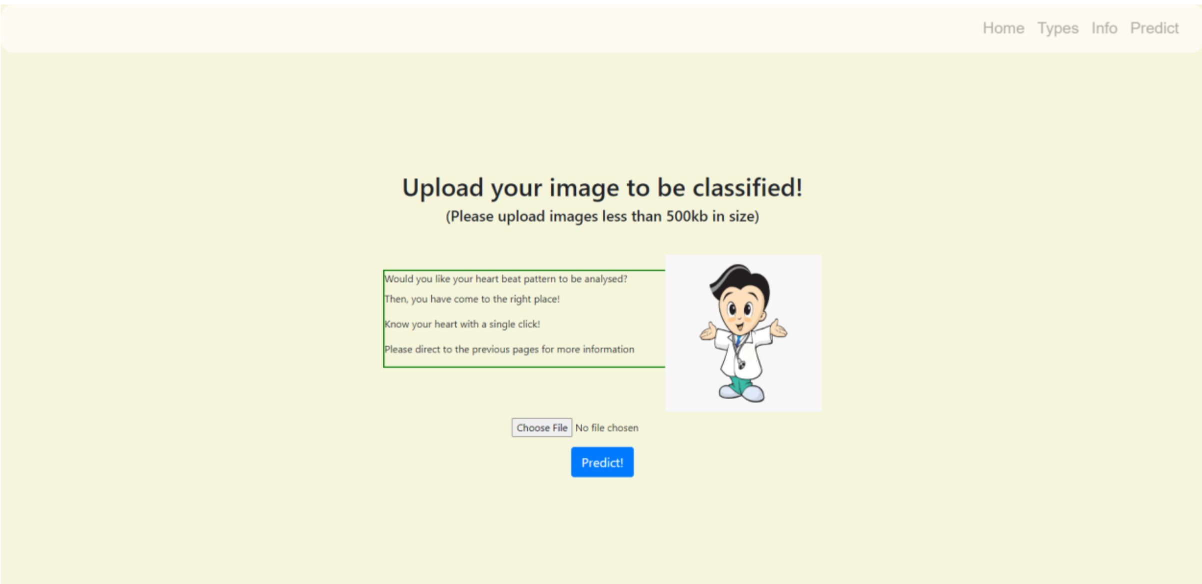
2) The Types Page- Shows the user the types of arrhythmia that exist



3) The Info Page -It shows the different remedies and treatment a patient can follow to cure themselves



4) The Predict Page- This page is where the CNN Model is running and will predict your heart rhythm to cpossibly classify the type of arrhythmia.



API FRAMEWORK

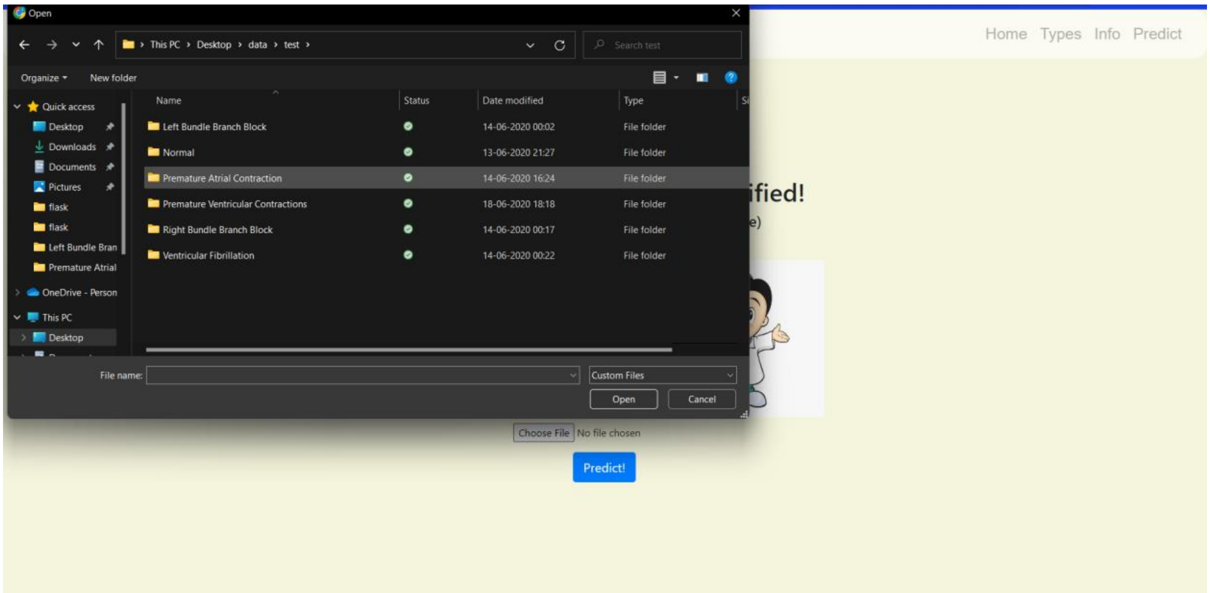
This will allow the website and the saved h5 model to integrate and work as a real time running website.

```python
import os
import numpy as np
from flask import Flask,request,render_template

from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image

app=Flask(__name__)
model=load_model('ECG_Classification.h5')

@app.route("/")
def about():
    return render_template("home.html")
@app.route("/about")
def home():
    return render_template("home.html")

@app.route("/types")
def types():
    return render_template("types.html")

@app.route("/info")
def information():
    return render_template("info.html")

@app.route("/upload")
def test():
    return render_template("predict.html")

```

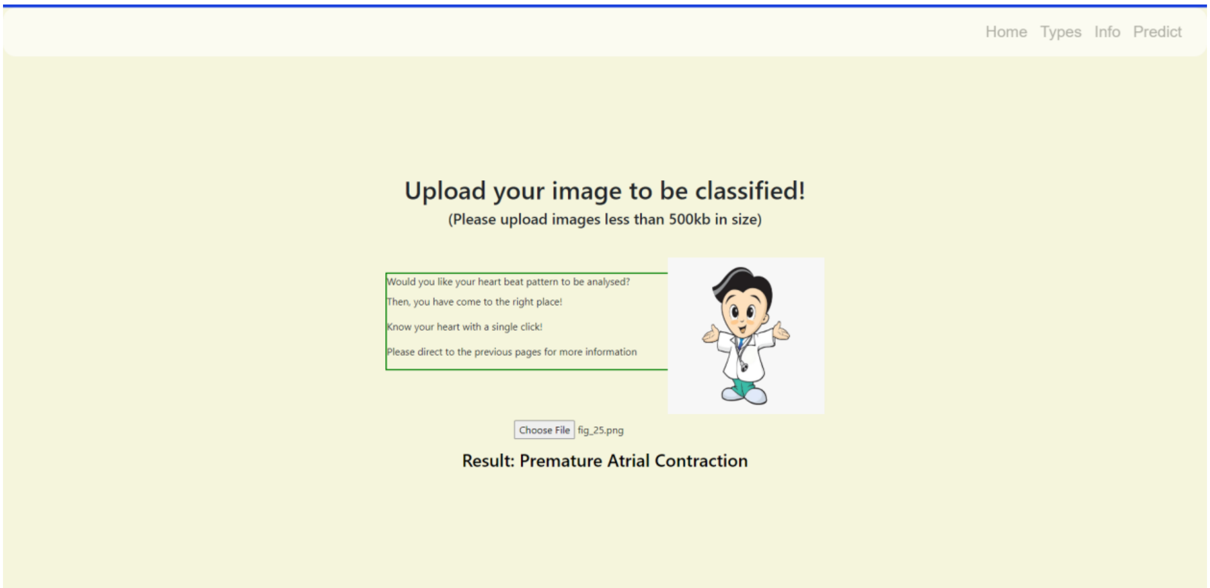
```python
@app.route("/predict",methods=["GET","POST"])
def upload():
    if request.method == 'POST':
        f=request.files['file'] #requesting the file
        basepath=os.path.dirname('__file__')#storing the file directory
        filepath=os.path.join(basepath,"uploads",f.filename)#storing the file in uploads folder
        f.save(filepath)#saving the file

        img=image.load_img(filepath,target_size=(64,64)) #load and reshaping the image
        x=image.img_to_array(img)#converting image to array
        x=np.expand_dims(x,axis=0)#changing the dimensions of the image

        pred=model.predict(x)#predicting classes
        y_pred = np.argmax(pred)
        print("prediction",y_pred)#printing the prediction

        index=['Left Bundle Branch Block','Normal','Premature Atrial Contraction',
       'Premature Ventricular Contractions', 'Right Bundle Branch Block','Ventricular Fibrillation']
        result=str(index[y_pred])

        return result
    return None

#port = int(os.getenv("PORT"))
if __name__=="__main__":
    app.run(debug=False)
    #app.run(host='0.0.0.0', port=8000)
```

THE WORKING OF PREDICTION PAGE

Lets upload an image to see if the model works with the website



An image that has Premature Atrial Contraction is chosen



The model has predicted it is indeed Premature Atrial Contraction.

CHAPTER 8: TESTING

Test Case ID	Feature Type	Component	Test Scenario	Expected Result	Actual Result	Status
HP_TC_001	UI	Home Page	Verify UI elements in the home page	The home page must be displayed properly	Working as expected	PASS
HP_TC_002	UI	Home Page	Check if the UI elements are displayed properly in different screen sizes.	The Home Pages must be displayed properly in all sizes	Working as expected	PASS
PP_TC_001	Functional	Predict Page	Check if user can upload their file	The input image should be uploaded to the application successfully	Working as expected	PASS
PP_TC_002	UI	Predict Page	Check if the input image is displayed properly	The input image should be displayed properly	The size of the input image exceeds the display container	FAIL
PP_TC_003	UI	Predict Page	Check if the result is displayed properly	The result should be displayed properly	Working as expected	PASS
M_TC_001	Functional	Model	Check if the model can predict the ECG	The model should predict the type of arrythmia	Working as expected	PASS
M_TC_002	Functional	Model	Check if the model can handle various image sizes	The model should rescale the image and predict the results	Working as expected	PASS
BE_TC_001	Functional	Backend	Check if all the routes are working properly	All the routes should properly work	Working as expected	PASS

CHAPTER 9: RESULTS

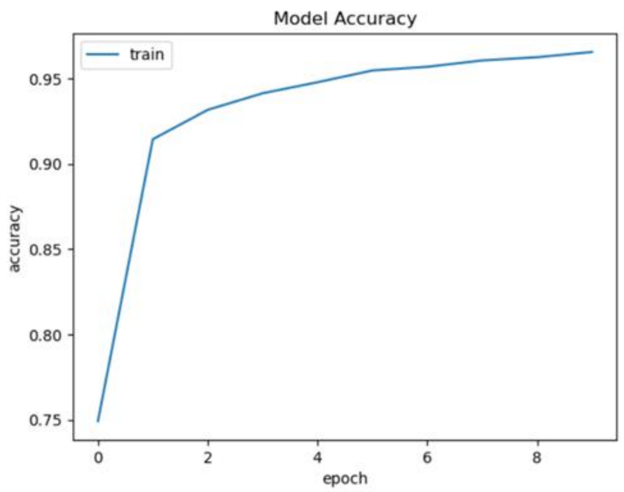
9.1 PERFORMANCE METRICS

MODEL SUMMARY

```
In [24]: #Summary of the model
         model.summary()

         Model: "sequential_1"
         _________________________________________________________________
          Layer (type)                Output Shape              Param #
         =================================================================
          conv2d (Conv2D)             (None, 62, 62, 32)        896

          max_pooling2d (MaxPooling2D  (None, 31, 31, 32)       0
          )

          conv2d_1 (Conv2D)           (None, 29, 29, 32)        9248

          max_pooling2d_1 (MaxPooling  (None, 14, 14, 32)       0
          2D)

          flatten (Flatten)           (None, 6272)              0

          dense (Dense)               (None, 32)                200736

          dense_1 (Dense)             (None, 32)                1056

          dense_2 (Dense)             (None, 6)                 198

         =================================================================
         Total params: 212,134
         Trainable params: 212,134
         Non-trainable params: 0
         _________________________________________________________________
```

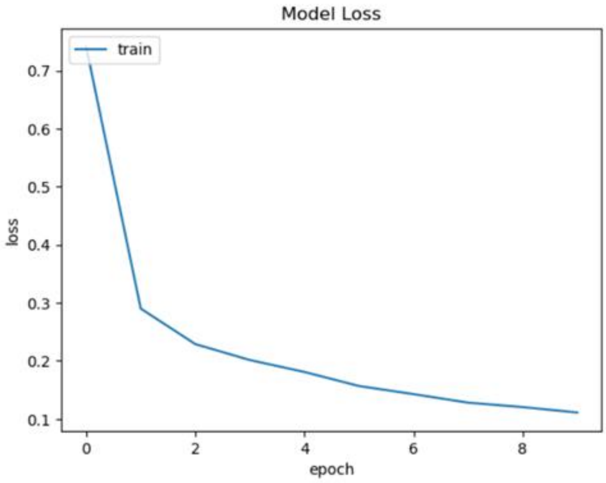
MODEL ACCURACY

```
In [46]: import matplotlib.pyplot as plt

         plt.plot(ECG.history['accuracy'])
         plt.title('Model Accuracy')
         plt.ylabel('accuracy')
         plt.xlabel('epoch')
         plt.legend(['train'],loc='upper left')
         plt.show()
```

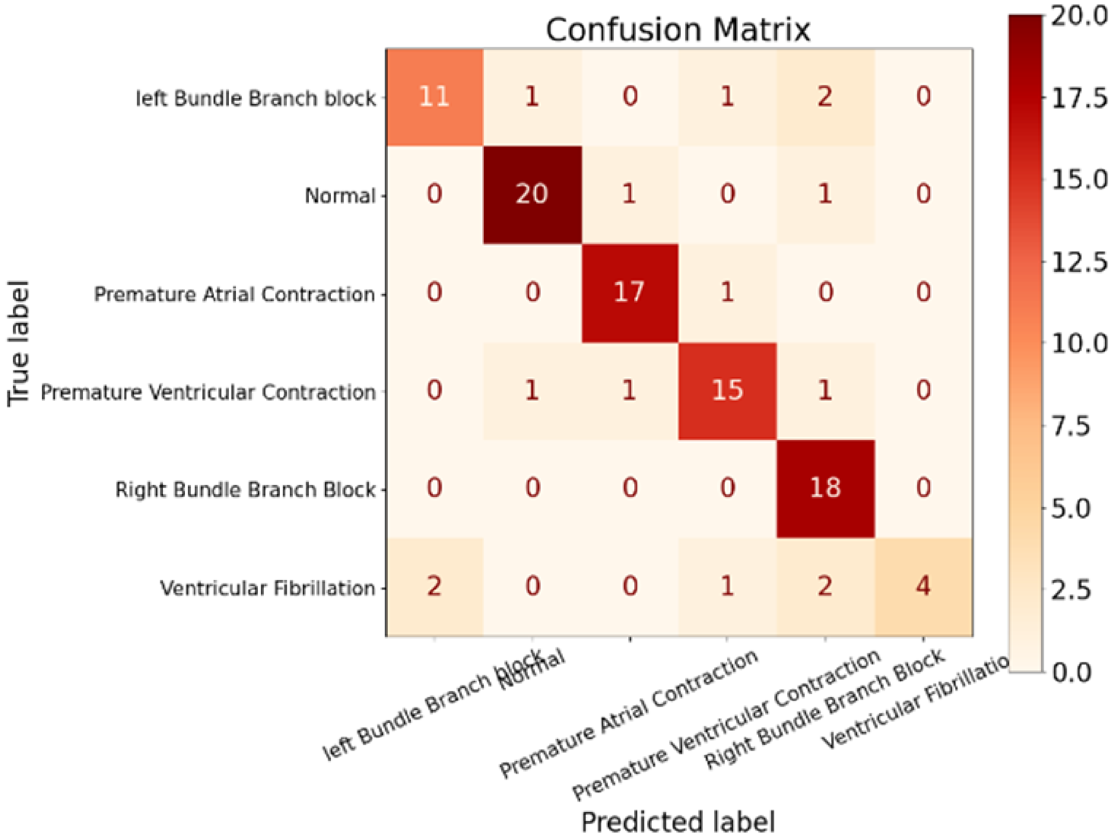


MODEL LOSS

```
In [47]: plt.plot(ECG.history['loss'])
         plt.title('Model Loss')
         plt.ylabel('loss')
         plt.xlabel('epoch')
         plt.legend(['train'],loc='upper left')
         plt.show()
```



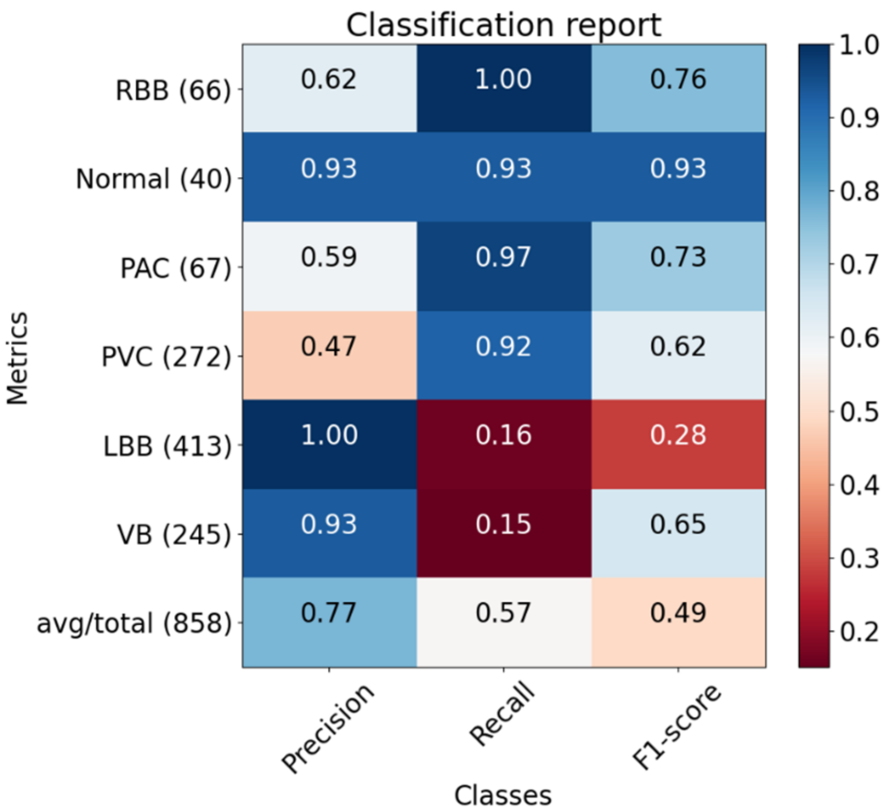
CONFUSION MATRIX



MODEL TRAINING

```
In [34]: ECG=model.fit_generator(generator=x_train,steps_per_epoch = len(x_train), epochs=10, validation_data=x_test,validation_steps = le

C:\Users\Adithyan\AppData\Local\Temp\ipykernel_1880\1751879658.py:1: UserWarning: `Model.fit_generator` is deprecated and will
be removed in a future version. Please use `Model.fit`, which supports generators.
  ECG=model.fit_generator(generator=x_train,steps_per_epoch = len(x_train), epochs=10, validation_data=x_test,validation_steps
= len(x_test))

Epoch 1/10
480/480 [==============================] - 53s 109ms/step - loss: 0.7406 - accuracy: 0.7491 - val_loss: 0.5269 - val_accuracy:
0.8328
Epoch 2/10
480/480 [==============================] - 55s 114ms/step - loss: 0.2901 - accuracy: 0.9143 - val_loss: 0.4048 - val_accuracy:
0.8793
Epoch 3/10
480/480 [==============================] - 55s 115ms/step - loss: 0.2289 - accuracy: 0.9315 - val_loss: 0.3774 - val_accuracy:
0.8730
Epoch 4/10
480/480 [==============================] - 53s 111ms/step - loss: 0.2015 - accuracy: 0.9412 - val_loss: 0.4461 - val_accuracy:
0.8693
Epoch 5/10
480/480 [==============================] - 52s 108ms/step - loss: 0.1809 - accuracy: 0.9477 - val_loss: 0.3213 - val_accuracy:
0.8875
Epoch 6/10
480/480 [==============================] - 52s 109ms/step - loss: 0.1568 - accuracy: 0.9546 - val_loss: 0.3302 - val_accuracy:
0.9007
Epoch 7/10
480/480 [==============================] - 52s 108ms/step - loss: 0.1428 - accuracy: 0.9567 - val_loss: 0.3251 - val_accuracy:
0.8993
Epoch 8/10
480/480 [==============================] - 52s 109ms/step - loss: 0.1281 - accuracy: 0.9604 - val_loss: 0.3794 - val_accuracy:
0.8813
Epoch 9/10
480/480 [==============================] - 55s 114ms/step - loss: 0.1204 - accuracy: 0.9623 - val_loss: 0.3525 - val_accuracy:
0.9067
Epoch 10/10
480/480 [==============================] - 54s 113ms/step - loss: 0.1111 - accuracy: 0.9654 - val_loss: 0.3401 - val_accuracy:
0.8989
```

CLASSIFICATION REPORT



CHAPTER 10: ADVANTAGES AND DISADVANTAGES

ADVANTAGES

1. Reduces human errors.
2. Increased accuracy rates.
3. Vast dataset handling.
4. Saves time.
5. User friendly and open availability.

DISADVANTAGES

1. Requires high data rates and performance servers.
2. Complexity increases.
3. Scope for error has to be very minimum.

CHAPTER 11:CONCLUSION

This project, "Classification of Arrhythmias using 2-D ECG signals with Deep Learning" involves the two-dimensional convolution of electrocardiogram signals (ECG) for pattern dissemination and the implementation of a Convolutional Neural Network (CNN) deep-learning model for prediction. The proposed model predicts Arrhythmia in images with a high accuracy rate of nearly 96%. The early detection of Arrhythmia gives a better understanding of disease causes, initiates therapeutic interventions, and enables the development of appropriate treatments. It is imperative to use computational techniques over the simple and hefty process of ECG record analysis. One such computation is dealt with in this project. Its main goals refer to reducing human errors, increasing efficiency, and thus guaranteeing better success rates. This project thus opens up an arena towards possible solutions amidst the challenges of the growing field of digital electronics and healthcare.

CHAPTER 12:FUTURE SCOPE

The project is far from completion. There is always room for improvement and advancement of technological edge in it. There could be furthering like-

1. Increased accuracy of model.
2. Determination of stage of Arrhythmias.
3. Bettering model to recognize motor symptoms.
4. Adding language efficiency for widening the user base.

This indicates how the scope can be extended for this project. The efficiency levels can be employed to greater extents. This would benefit the healthcare industry immensely. It would make medical professionals and other users rely more on technological involvement in medicine.

APPENDIX

SOURCE CODE:

The model was built using IBM Watson Studio and deployed

A h5 file is generated and saved in the system

IMAGE PREPROCESSING

In [118…
```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.layers import Convolution2D
from tensorflow.keras.layers import MaxPooling2D
from tensorflow.keras.layers import Flatten
```

In [119…
```
#Loading dataset into the cloud
import os, types
import pandas as pd
from botocore.client import Config
import ibm_boto3

def __iter__(self): return 0

# @hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
# You might want to remove those credentials before you share the notebook.
cos_client = ibm_boto3.client(service_name='s3',
    ibm_api_key_id='67UVPINvQa8xHKQGFZ7G0S6A3_Yta4PBlnd8F11OZr30',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'ecgarrythmia-donotdelete-pr-xnjamjasezeqom'
object_key = 'Classification of Arrhythmia by Using Deep Learning with 2-D ECG Spectral Image Representation.zip'

streaming_body_1 = cos_client.get_object(Bucket=bucket, Key=object_key)['Body']

# Your data file was loaded into a botocore.response.StreamingBody object.
# Please read the documentation of ibm_boto3 and pandas to learn more about the possibilities to load the data.
# ibm_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/
# pandas documentation: http://pandas.pydata.org/
```

In [120…
```
#unzipping your data file
from io import BytesIO
import zipfile
unzip=zipfile.ZipFile(BytesIO(streaming_body_1.read()),'r')
file_paths=unzip.namelist()
for path in file_paths:
    unzip.extract(path)
```

In [121…
```
pwd
```

Out[121]:
```
'/home/wsuser/work'
```

In [122…
```
import os
filenames=os.listdir('/home/wsuser/work/data/train')
```

In [123…
```
from tensorflow.keras.preprocessing.image import ImageDataGenerator
```

In [124…
```
# setting parameters for data augmentation for training data
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
```

In [125…
```
# data augmentation to the testing data
test_datagen=ImageDataGenerator(rescale=1./255)
```

In [126…
```
#performing data augmentation to train data
x_train=train_datagen.flow_from_directory(directory=r'/home/wsuser/work/data/train',target_size=(64,64),batch_size=32,class_mode='categorical')
```

```
Found 15341 images belonging to 6 classes.
```

In [127…
```
x_test=test_datagen.flow_from_directory(directory=r'/home/wsuser/work/data/test',target_size=(64,64),batch_size=32,class_mode='categorical')
```

```
Found 6825 images belonging to 6 classes.
```

In [128…
```
#Lets see the classes the different types of arrythmia is stored in
x_train.class_indices
```

Out[128]:
```
{'Left Bundle Branch Block': 0,
 'Normal': 1,
 'Premature Atrial Contraction': 2,
 'Premature Ventricular Contractions': 3,
 'Right Bundle Branch Block': 4,
 'Ventricular Fibrillation': 5}
```

MODEL BUILDING

MODEL BUILDING- IMPORTING LIBRARIES

In [129…
```
import numpy as np
import tensorflow
from tensorflow.keras.models import Sequential
from tensorflow.keras import layers

from tensorflow.keras.layers import Dense,Flatten

from tensorflow.keras.layers import Conv2D,MaxPooling2D
import tensorflow.keras
```

In [130…
```
model=Sequential()
```

Adding CNN Layers

In [131…
```
model.add(Conv2D(32,(3,3),input_shape=(64,64,3),activation='relu')) #Activation Function
model.add(MaxPooling2D(pool_size=(2,2))) #Downsampling Purposes
model.add(Conv2D(32,(3,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten()) #Flatten the dimensions of the image
```

ADDING DENSE LAYERS

In [132…
```
model.add(Dense(32))# Deeply connected neural network layers
model.add(Dense(6,activation='softmax'))#Output layer with 6 neurons
```

In [133…
```
#Summary of the model
model.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d_2 (Conv2D)           (None, 62, 62, 32)        896

 max_pooling2d_2 (MaxPooling  (None, 31, 31, 32)       0
 2D)

 conv2d_3 (Conv2D)           (None, 29, 29, 32)        9248

 max_pooling2d_3 (MaxPooling  (None, 14, 14, 32)       0
 2D)

 flatten_1 (Flatten)         (None, 6272)              0
```

EVALUATING LOSS FUNCTION

```
In [134…  model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])
```

TRAINING THE MODEL

```
In [135…  model.fit_generator(generator=x_train,steps_per_epoch = len(x_train), epochs=10, validation_data=x_test,validation_steps = len(x_test))
```

```
/tmp/wsuser/ipykernel_165/53529210.py:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which supports generators.
  model.fit_generator(generator=x_train,steps_per_epoch = len(x_train), epochs=10, validation_data=x_test,validation_steps = len(x_test))
Epoch 1/10
480/480 [==============================] - 92s 191ms/step - loss: 0.8288 - accuracy: 0.7166 - val_loss: 0.4710 - val_accuracy: 0.8513
Epoch 2/10
480/480 [==============================] - 95s 197ms/step - loss: 0.2687 - accuracy: 0.9216 - val_loss: 0.3495 - val_accuracy: 0.8847
Epoch 3/10
480/480 [==============================] - 92s 192ms/step - loss: 0.2053 - accuracy: 0.9409 - val_loss: 0.3422 - val_accuracy: 0.8894
Epoch 4/10
480/480 [==============================] - 93s 194ms/step - loss: 0.1747 - accuracy: 0.9469 - val_loss: 0.2514 - val_accuracy: 0.9262
Epoch 5/10
480/480 [==============================] - 93s 194ms/step - loss: 0.1547 - accuracy: 0.9524 - val_loss: 0.2279 - val_accuracy: 0.9366
Epoch 6/10
480/480 [==============================] - 92s 191ms/step - loss: 0.1448 - accuracy: 0.9560 - val_loss: 0.2173 - val_accuracy: 0.9262
Epoch 7/10
480/480 [==============================] - 93s 193ms/step - loss: 0.1307 - accuracy: 0.9608 - val_loss: 0.2533 - val_accuracy: 0.9483
Epoch 8/10
480/480 [==============================] - 93s 193ms/step - loss: 0.1247 - accuracy: 0.9616 - val_loss: 0.2501 - val_accuracy: 0.9355
Epoch 9/10
480/480 [==============================] - 93s 194ms/step - loss: 0.1112 - accuracy: 0.9654 - val_loss: 0.2837 - val_accuracy: 0.9231
Epoch 10/10
480/480 [==============================] - 92s 192ms/step - loss: 0.1052 - accuracy: 0.9677 - val_loss: 0.2334 - val_accuracy: 0.9481
Out[135]: <keras.callbacks.History at 0x7f2c841c2eb0>
```

SAVING THE MODEL

```
In [136…  model.save('ECG_Classification.h5')
```

```
In [137…  !tar -zcvf ECG-Arrythmia-model_new.tgz ECG_Classification.h5
```

```
ECG_Classification.h5
```

```
In [138…  ls -1
```

```
data/
ECG-arrhythmia-classification-model_new.tgz
ECG-Arrythmia-model_new.tgz
ECG-classification
ECG_Classification.h5
my_model.tar1.gz
my_model.tar.gz
```

```
In [139…  #Installing amchine learning service
          !pip install watson-machine-learning-client --upgrade
```

```
Requirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas->watson-machi
```

```
In [179…  #Replace the credentials that you got from Watson Machine Learning Service
          from ibm_watson_machine_learning import APIClient
          wml_credentials={
              "url":"https://us-south.ml.cloud.ibm.com",
              "apikey":"ge1RCI4GpBshW0Zu98SpfrdAdy6U4WbXKbiRUIxluG10"
          }
          client=APIClient(wml_credentials)
```

```
In [141…  client=APIClient(wml_credentials)
```

```
In [142…  def guid_from_space_name(client,space_name):
              space=client.spaces.get_details()
              #print(space)
              return(next(item for item in space['resources'] if item['entity']["name"]==space_name)['metadata']['id'])
```

```
In [143…  space_uid=guid_from_space_name(client, 'ECG-Arrythmia')
          print("space UID="+ space_uid)
```

```
space UID=0680da27-17fb-450a-8b11-fc04d2de3272
```

```
In [144…  client.set.default_space(space_uid)
Out[144]: 'SUCCESS'
```

```
In [145…  client.software_specifications.list()
```

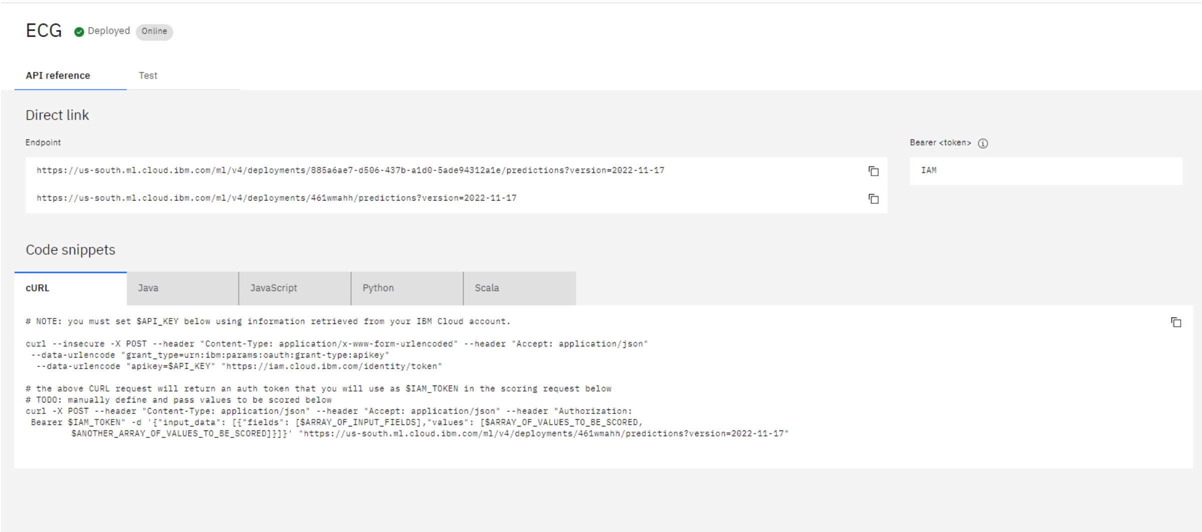
```
In [146…  software_spec_uid = client.software_specifications.get_uid_by_name("tensorflow_rt22.1-py3.9")
          software_spec_uid

Out[146]: 'acd9c798-6974-5d2f-a657-ce06e986df4d'

In [147…  model_details = client.repository.store_model(model='ECG-Arrythmia-model_new.tgz',meta_props={
              client.repository.ModelMetaNames.NAME:"CNN",
              client.repository.ModelMetaNames.TYPE:"tensorflow_2.7",
              client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_spec_uid})
          model_id=client.repository.get_model_uid(model_details)

          This method is deprecated, please use get_model_id()
          /opt/conda/envs/Python-3.9/lib/python3.9/site-packages/ibm_watson_machine_learning/repository.py:1453: UserWarning: This method is deprecated, please use get_model_id()
            warn("This method is deprecated, please use get_model_id()")

In [148…  model_id

Out[148]: '73ecebcb-85ef-4e32-aabd-577361841231'

In [150…  client.repository.download(model_id,'my_model.tar2.gz')

          Successfully saved model content to file: 'my_model.tar2.gz'
Out[150]: '/home/wsuser/work/my_model.tar2.gz'
```

TESTING THE MODEL

```
In [156…  from keras.models import load_model
          from keras.preprocessing import image

In [157…  import os, types
          import pandas as pd
          from botocore.client import Config
          import ibm_boto3

          def __iter__(self): return 0

          # @hidden_cell
          # The following code accesses a file in your IBM Cloud Object Storage. It includes your credentials.
          # You might want to remove those credentials before you share the notebook.
          cos_client = ibm_boto3.client(service_name='s3',
              ibm_api_key_id='67UVPINvQa8xHKQGFZ7G0S6A3_Yta4PBlnd8F11OZr30',
              ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
              config=Config(signature_version='oauth'),
              endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

          bucket = 'ecgarrythmia-donotdelete-pr-xnjamjasezeqom'
          object_key = 'pac.jpg'

          streaming_body_3 = cos_client.get_object(Bucket=bucket, Key=object_key)['Body']

          # Your data file was loaded into a botocore.response.StreamingBody object.
          # Please read the documentation of ibm_boto3 and pandas to learn more about the possibilities to load the data.
          # ibm_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/
          # pandas documentation: http://pandas.pydata.org/

In [175…  pwd

Out[175]: '/home/wsuser/work'

Out[175]: '/home/wsuser/work'

In [176…  model=load_model('/home/wsuser/work/ECG_Classification.h5')

In [177…  #Loading of image
          img=image.load_img(r'/home/wsuser/work/data/test/Premature Ventricular Contractions/fig_5660.png',target_size=(64,64))
          #Converting image to array
          x=image.img_to_array(img)
          #Changing the shape
          x=np.expand_dims(x,axis=0)
          #Predicting the classes
          pred=model.predict(x)
          y_pred=np.argmax(pred)
          y_pred

Out[177]: 3

In [178…  index=['left Bundle Branch block','Normal','Premature Atrial Contraction','Premature Ventricular Contraction','Right Bundle Branch Block','Ventricular Fibrillation']
          result = str(index[y_pred])
          result

Out[178]: 'Premature Ventricular Contraction'
```

So, the model is saved and tested . An image was given as input ans the model predicted Premature Ventricular Contraction marking success of the model.

Below image shows that the model deployment is successful.



GITHUB

https://github.com/IBM-EPBL/IBM-Project-42347-1660660408

DEMO LINK

https://drive.google.com/drive/folders/1p0H-xlYuyAOw0GwTj_j94I_fHPXr6Z1g?usp=sharing