# CLASSIFICATION OF ARRHYTHMIA USING DEEP LEARNING WITH 2-D ECG SPECTRAL IMAGE PRESENTATION

**TEAM ID : PNT2022TMID31222 TEAM MEMBERS:** 1. PRASANTH A

1. BHARATH S
2. NAVEENSRIDHAR K
3. RITHAN S
4. PRASANTH R

# INTRODUCTION

* 1. Project Overview

In recent years, cardiovascular diseases (CVDs) are the major cause of death. Over

17.7 million people died from CVDs in the year 2017 all over the world. 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. In this project, we build an effective electrocardiogram (ECG) arrhythmia classification method using a convolutional neural network (CNN), in which we classify ECG into six categories, one being normal and the other five being different types of arrhythmia using deep two-dimensional CNN with grayscale ECG images. We are creating a web application where the user selects the image which is to be classified. The image is fed into the model that is trained and the cited class will be displayed on the webpage.

1.2 Purpose

Deep learning techniques surpassed traditional techniques especially in pattern recognition.It is mostly applied to analyze visual images. So these techniques can be used to predict the types of arrhythmia using ECG images.We can create an application to upload and display the results. It makes it easier for everyone.

# LITERATURE SURVEY

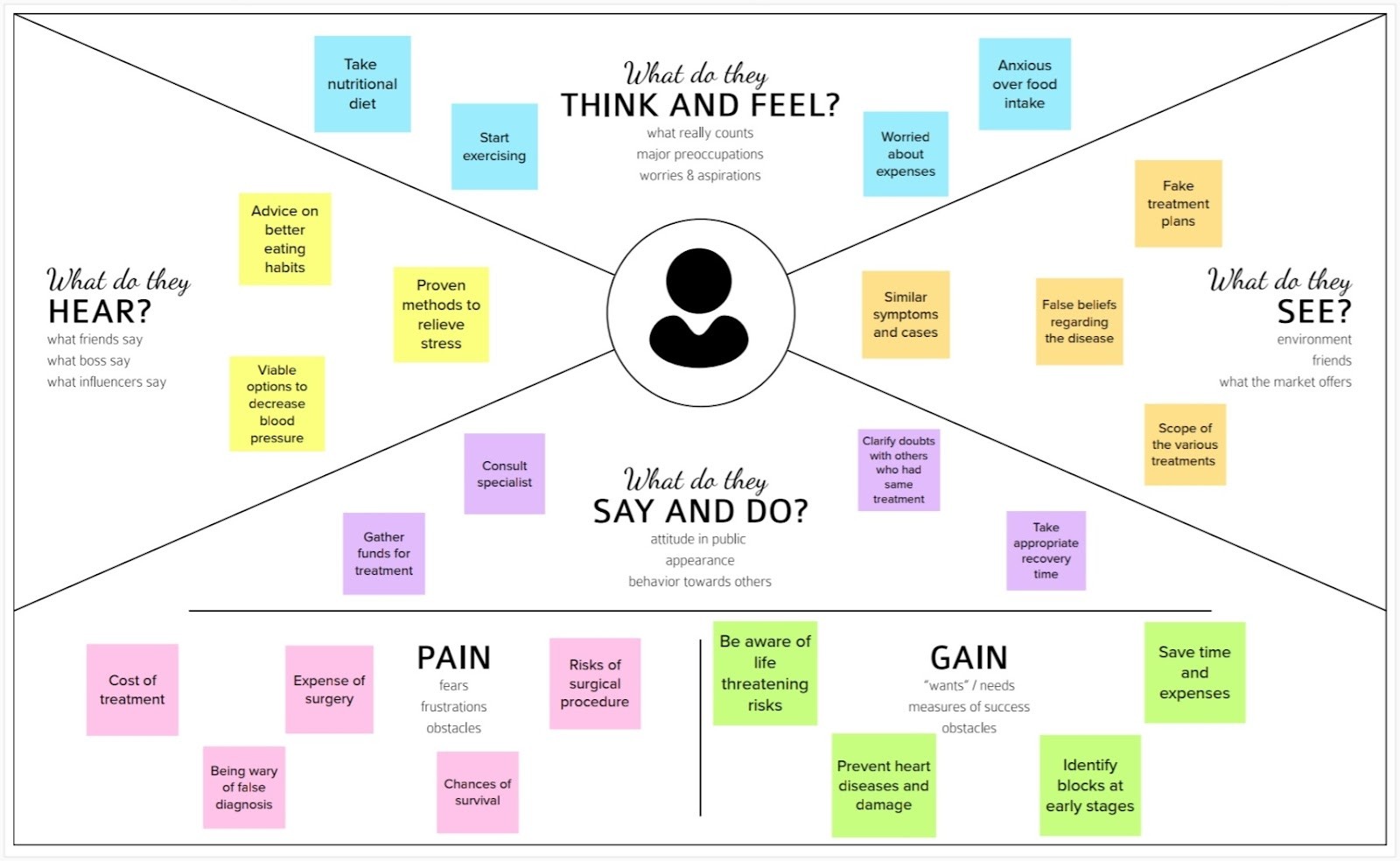
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S. NO** | **PAPER** | **AUTHOR** | **YEAR** | **DESCRIPTION** |
| 1 | Classification of | Wusat Ullah,Imran | 2021 | Aims to apply deep learning |
|  | Arrhythmia in | Siddique , Rana |  | techniques on the publicly |
|  | Heartbeat Detection | Muhammad |  | available dataset to classify |
|  | Using Deep | Zulqarnain |  | arrhythmia. The system |
|  | Learning | ,Mohamm ad |  | combines three different types |
|  |  | Mahtab Alam , |  | of information: RR |
|  |  | Irfan Ahmad, and |  | intervals,signal morphology, |
|  |  | Usman Ahmad |  | and higher-level statistical |
|  |  | Raza. |  | data. It is concluded that fuzzy |
|  |  |  |  | based technology is successful |
|  |  |  |  | in the analysis of computerized |
|  |  |  |  | ECG but needs more research |
| 2 | Arrhythmia | Ali Haider Khan | 2021 | The automated screening of |
|  | Classification | ,Muzammil |  | arrhythmia classification using |
|  | Techniques Using | Hussain ,and |  | ECG beats is developed for |
|  | Deep Neural | Muhammad |  | ages. The deep learning based |
|  | Network | Kamran Malik |  | automated Arrhythmia |
|  |  |  |  | Classification techniques are |
|  |  |  |  | developed with high accuracy. |
|  |  |  |  | The primary concerns that |
|  |  |  |  | affect the success of the |
|  |  |  |  | Developed arrhythmia |
|  |  |  |  | detection systems are (i) |
|  |  |  |  | manual features selection, (ii) |
|  |  |  |  | techniques used for features |
|  |  |  |  | extraction, and (iii) algorithm |
|  |  |  |  | used for classification and the |
|  |  |  |  | most important is the use of |
|  |  |  |  | imbalanced data for |
|  |  |  |  | classification. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 3 | Classification of | Amin Ullah, Syed | 2020 | Proposal of two-dimensional |
|  | Arrhythmia by | Anwar, |  | (2-D)convolutional neural |
|  | Using Deep | Muhammad Bilal, |  | network (CNN)model for the |
|  | Learning with 2- D | Raja Majid |  | classification of ECGsignals |
|  | ECG Spectral Image | Mehmood |  | into eight classes; |
|  | Representation |  |  | namely,normal beat, premature |
|  |  |  |  | ventricular contraction beat, |
|  |  |  |  | paced beat, right bundle branch |
|  |  |  |  | block beat, left bundle branch |
|  |  |  |  | block beat, atrial premature |
|  |  |  |  | contraction beat, ventricular |
|  |  |  |  | flutter wave beat, and |
|  |  |  |  | ventricular escape beat. The |
|  |  |  |  | one-dimensional ECG time |
|  |  |  |  | series signals are transformed |
|  |  |  |  | into 2-Dspectrograms through |
|  |  |  |  | a short-time Fouriertransform. |
|  |  |  |  | The 2-D CNN model |
|  |  |  |  | consisting of four |
|  |  |  |  | convolutional layers and four |
|  |  |  |  | pooling layers is designed for |
|  |  |  |  | extracting robust features from |
|  |  |  |  | the input spectrograms. |

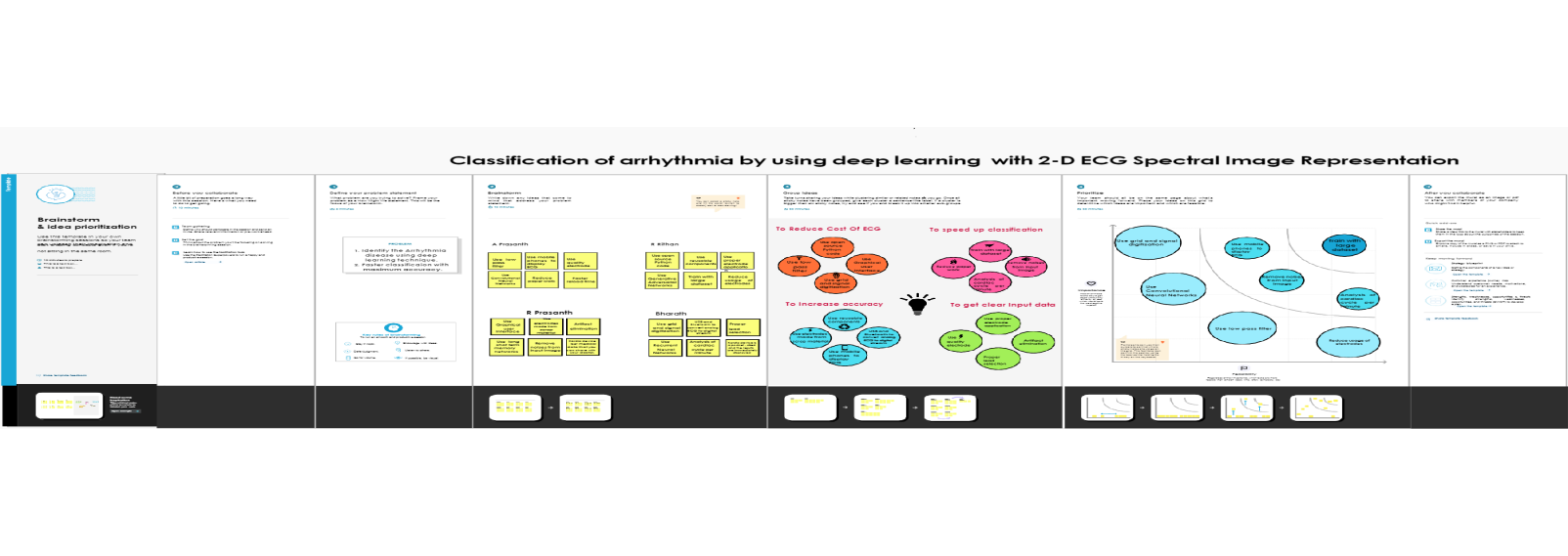
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 4 | A deep | Rajendra Acharya, | 2017 | The basis of arrhythmia |
|  | convolutional neural | Shu Lih Oh, Yuki |  | diagnosis is the identification |
|  | network model to | Hagiwara, Jen |  | of normal versus abnormal |
|  | classify heartbeats | Hong Tan, |  | individual heart beats, and |
|  |  | Muhammad Adam |  | their correct classification into |
|  |  |  |  | different diagnoses,based on |
|  |  |  |  | ECG morphology. Heartbeat |
|  |  |  |  | scan be subdivided into five |
|  |  |  |  | categories namely non-ectopic, |
|  |  |  |  | supraventricular ventricular |
|  |  |  |  | ectopic, ventricular ectopic, |
|  |  |  |  | fusion, andun-known beats. It |
|  |  |  |  | is challenging and time- |
|  |  |  |  | consuming to distinguish these |
|  |  |  |  | heartbeats on ECG as these |
|  |  |  |  | signals are typically corrupted |
|  |  |  |  | by noise. We developed a 9- |
|  |  |  |  | layer deep convolutional |
|  |  |  |  | neural network (CNN) to |
|  |  |  |  | automatically identify 5 |
|  |  |  |  | different categories of |
|  |  |  |  | heartbeats in ECG signals. Our |
|  |  |  |  | experiment was conducted in |
|  |  |  |  | original and noise attenuated |
|  |  |  |  | sets of ECG signals derived |
|  |  |  |  | from a publicly available |
|  |  |  |  | database. |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 5 | Cardiac arrhythmia | Ali Isina, | Selen | 2017 | An electrocardiogram is an |
|  | detection using deep | Ozdalili |  |  | important diagnostic tool for |
|  | Learning |  |  |  | the assessment of cardiac |
|  |  |  |  |  | arrhythmias in clinical routine. |
|  |  |  |  |  | A deep learning framework |
|  |  |  |  |  | previously trained on a general |
|  |  |  |  |  | image data set is transferred to |
|  |  |  |  |  | carry out automatic |
|  |  |  |  |  | ECGarrhythmia diagnostics by |
|  |  |  |  |  | classifying patient ECG’s into |
|  |  |  |  |  | corresponding cardiac |
|  |  |  |  |  | conditions. Transferred deep |
|  |  |  |  |  | convolutional neural networks |
|  |  |  |  |  | are used as feature extractor |
|  |  |  |  |  | and the extracted features are |
|  |  |  |  |  | fed into a simple |
|  |  |  |  |  | backpropagation neural |
|  |  |  |  |  | network to carry out the final |
|  |  |  |  |  | classification. |

1. **IDEATION & PROPOSED SOLUTION**
   1. Empathy Map Canvas



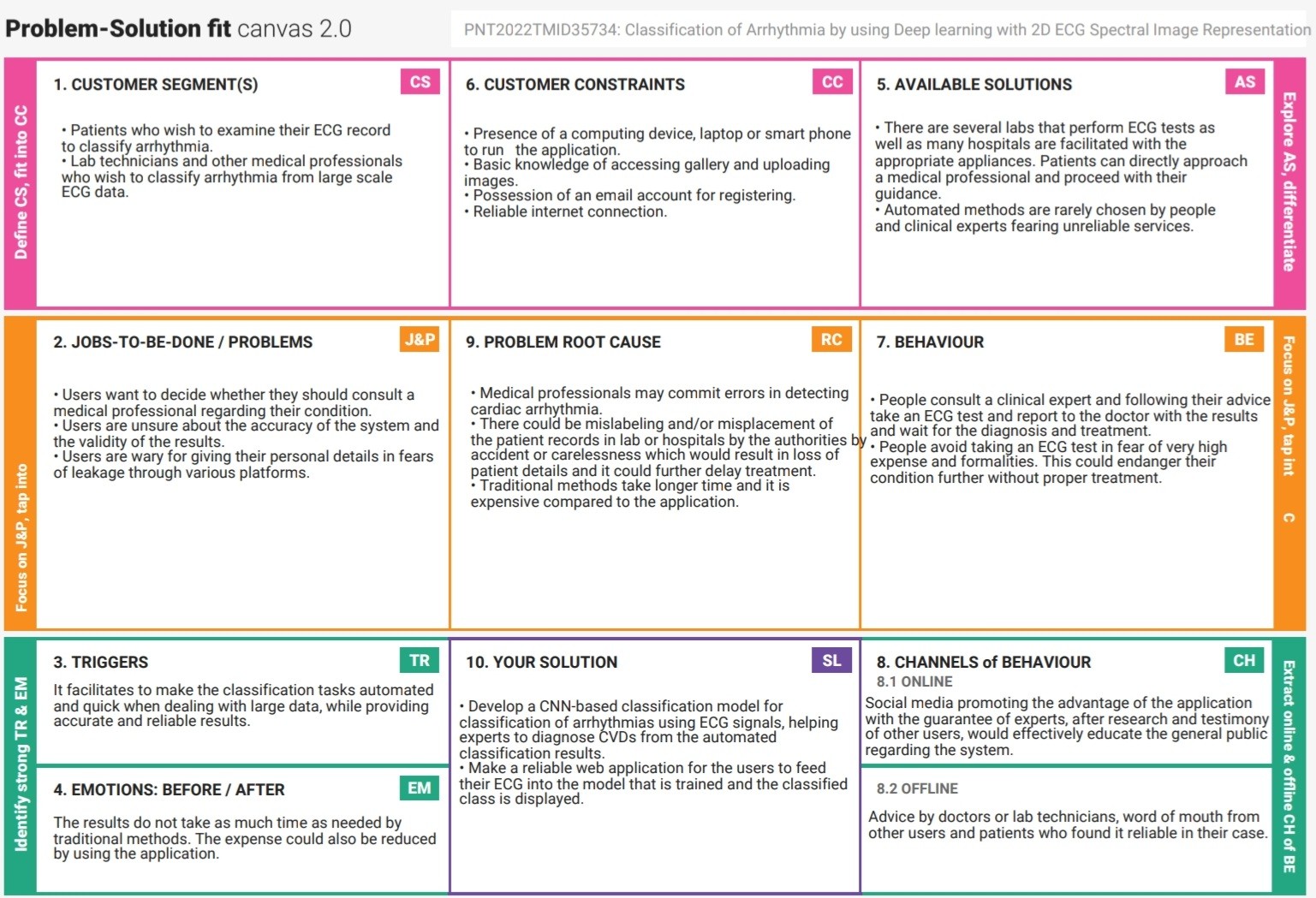
* 1. Ideation & Brainstorming



* 1. Proposed Solution

The automatic classification of arrhythmia using the ECG signal in a supervised way is proposed using a CNN-based model. 2. The type of arrhythmia present is identified by appropriate labeling on the ECG data utilized in the study.1 3. Expert cardiologists assigned these designations, which are then used for supervised training.The arrhythmia class label was applied to the associated spectrogram picture representation for each heartbeat segment.Comparative analysis of various CNN models like ResNet, Xception, VGG19 and a custom model will be performed before deploying the model with the best performance in the web application. It is extremely difficult to predict abnormal heart rates interactively. As a result, an automated system capable of identifying discrete abnormal heartbeats from a large amount of ECG data will promote safe and independent living among the public which will make them more self-reliable.

* 1. Problem Solution Fit



# REQUIREMENT ANALYSIS

* 1. Functional Requirements

|  |  |  |
| --- | --- | --- |
| **SNO** | **REQUIREMENTS** | **SUB REQUIREMENTS** |
| 1 | User Input | Upload ECG image as JPEG/PNG |
| 2 | Process Image | The trained CNN model processes the input image to classify the Arrhythmia. |
| 3 | Generate Result | Display the classification result on the screen. |

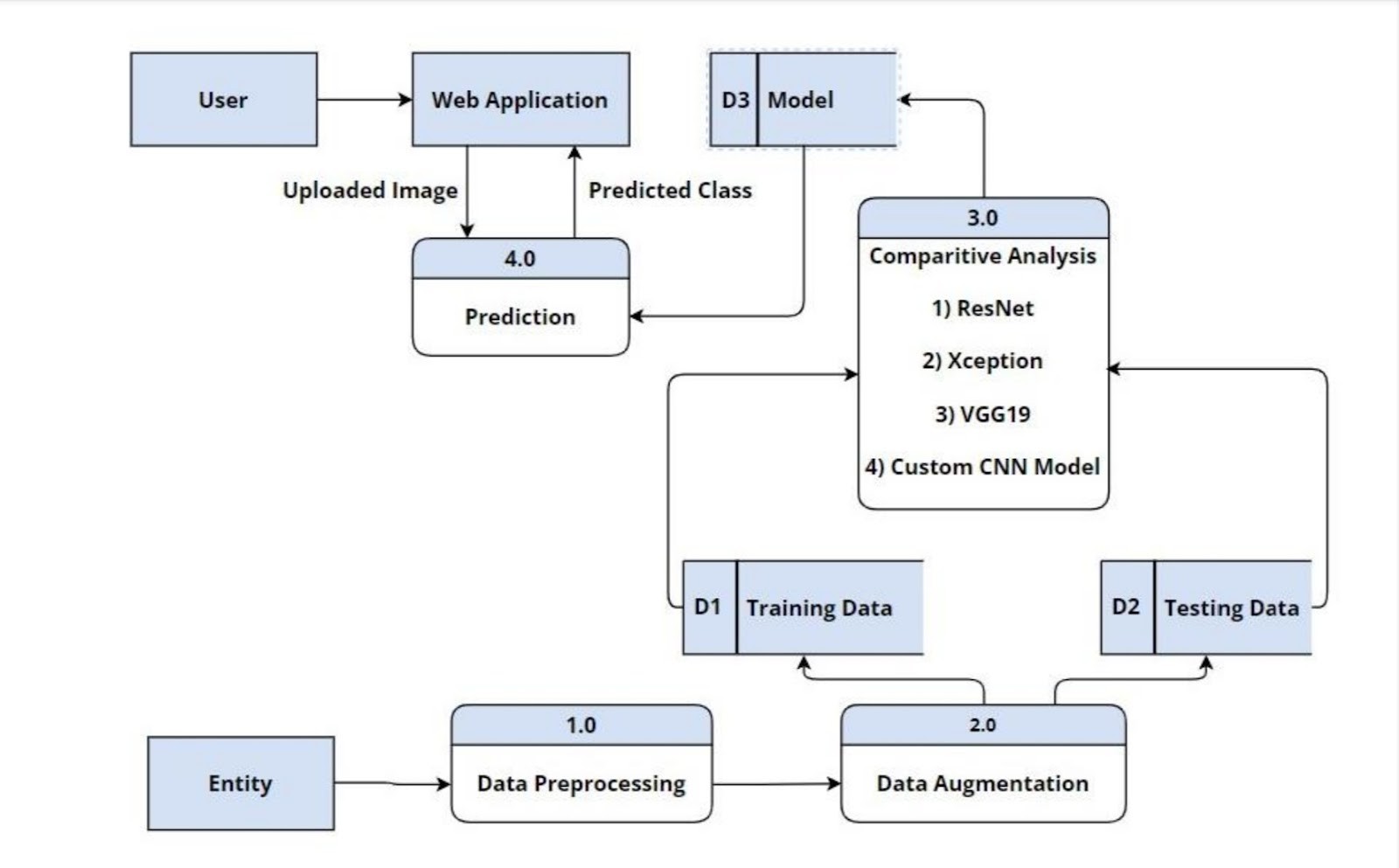
* 1. Non-Functional Requirements

|  |  |  |
| --- | --- | --- |
| **SNO** | **NON-FUNCTIONAL REQUIREMENTS** | **DESCRIPTION** |
| 1 | Usability | It is a user-friendly application which allows users to upload ECG images to classify Arrhythmia. |
| 2 | Security | Data is not used for any other purposes other than processing. Only users can view the results of the uploaded image. |
| 3 | Reliability | The application is defect free, deployed with a high accuracy CNN model which provides the correct prediction for the given input. |
| 4 | Performance | High accuracy models are used for classification thereby increasing the performance of the |

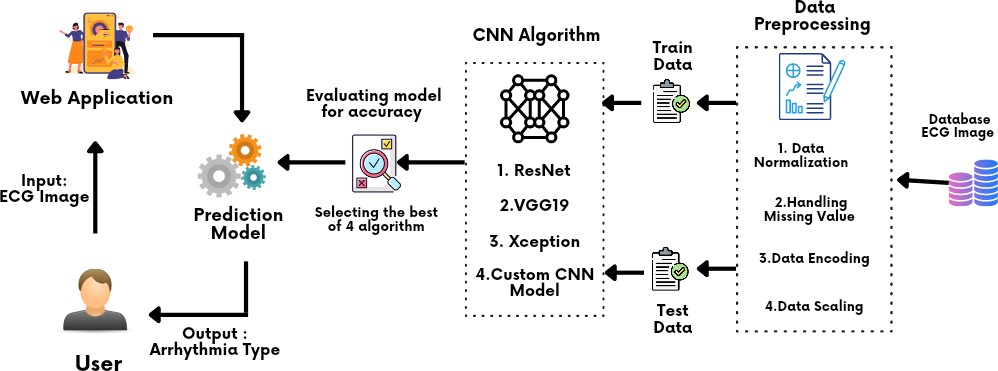
|  |  |  |
| --- | --- | --- |
|  |  | application |
| 5 | Availability | The application can be accessed anytime from anywhere with an internet connection. |
| 6 | Scalability | The system must be scalable to process multiple images. Multiple users must be able to access the system simultaneously without traffic. |

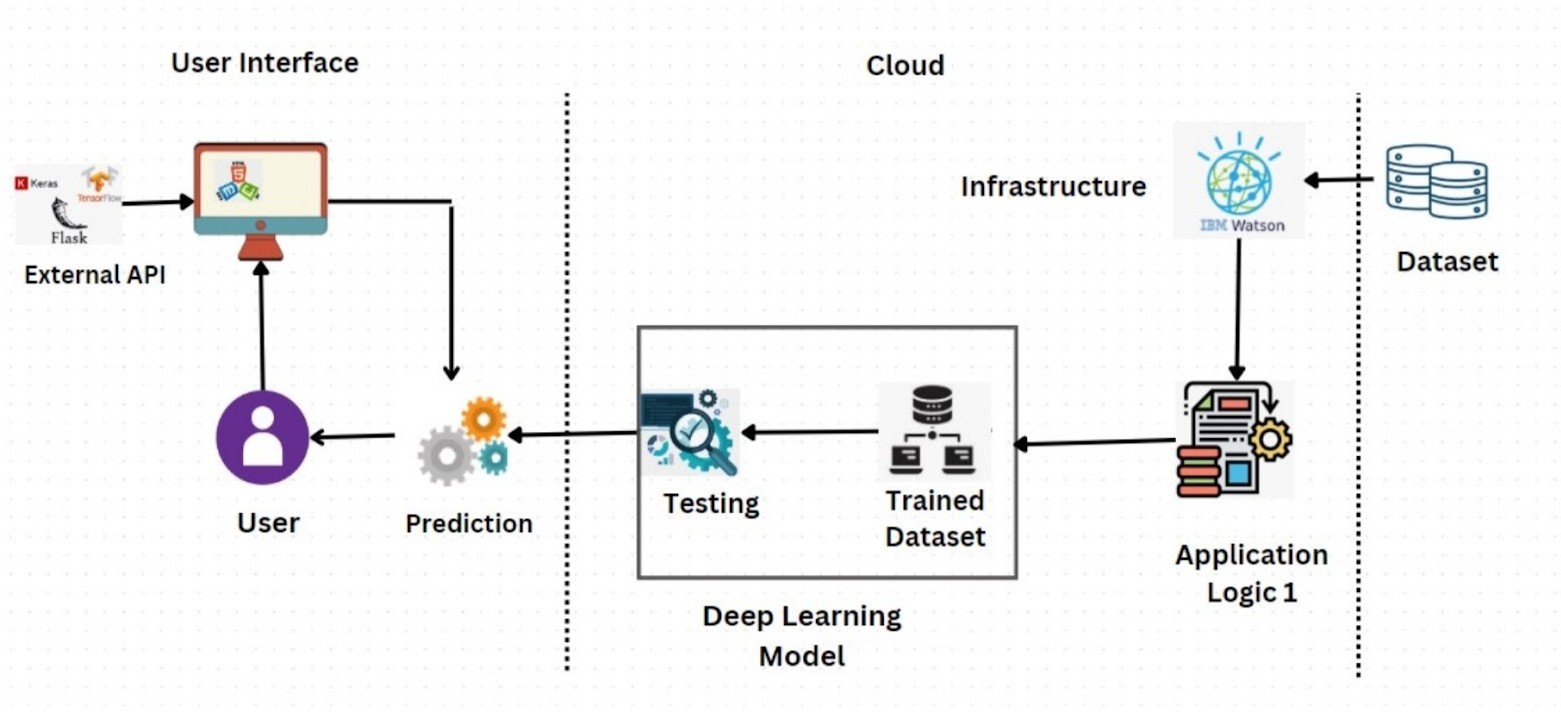
# PROJECT DESIGN

* 1. Data Flow Diagrams



* 1. Solution & Technical Architecture





* 1. User Stories

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User Type** | **Functional Requirement** | **User Story No** | **User Story** | **Acceptance Criteria** | **Priori ty** | **Relea se** |
| Mobile / Web User | Image Upload | USN-1 | As a user, I can upload an ECG image in the application. | I can upload an ECG  image and  click on ‘Upload’ to get the  result. | High | Sprint 2 |
|  | Prediction Result | USN-2 | As a user, I can view the predicted class in the application. | I can view the result of my uploaded image | High | Sprint 2 |
| Customer care executive | Support | USN-1 | As a customer care executive, I can provide support and  solve the  issues that  users face  with the application. | I can  provide support and solve issues  w.r.t the application | Medi um | Sprint 3 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Administr ator | Application Maintenance | USN-1 | As an  administrator, I can upgrade or update the application whenever a  bug is  discovered, or a newer  version is developed. | I can update or upgrade the application | High | Sprint 1 |
|  | Application Security | USN-2 | As an  administrator, I can  implement security measures and make necessary changes. | I can make the application more secure. | High | Sprint 1 |

# PROJECT PLANNING & SCHEDULING

* 1. Sprint Planning & Estimation

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Functional Requireme nt** | **User Story NO** | **User Story** | **Story Poin ts** | **Priori ty** | **Team Members** |
| Sprin t1 | Dashboard | US N1 | As a user, based on my requirement I can navigate through the | 2 | Low | Prasanth A |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  | dashboard. |  |  |  |
| Sprin t1 | Pre-process the dataset | US N2 | The image dataset is pre-processed. | 4 | Medi um | Prasanth A  Prasanth R |
| Sprin t2 | Upload images and display output page | US N1 | As a user, I should be able to upload the image of ECG and get the output | 6 | High | Rithan R |
| Sprin t2 | Train the pre- trained model | US N2 | The pre-trained models Inception, ResNet and AlexNet are trained on the preprocessed dataset | 6 | High | Rithan R  Naveensridhar K |
| Sprin t2 | Build Python Code | US N3 | Build the flask file ‘app.py’ which is a web framework written in python for server side scripting | 8 | High | Prasanth A |
| Sprin t3 | Train custom CNN model | US N1 | Train the model with the image dataset. fit\_generator functions are used to train a deep learning neural network | 10 | High | Prasanth A  Rithan  Bharath  Naveensridhar  Prasanth R |

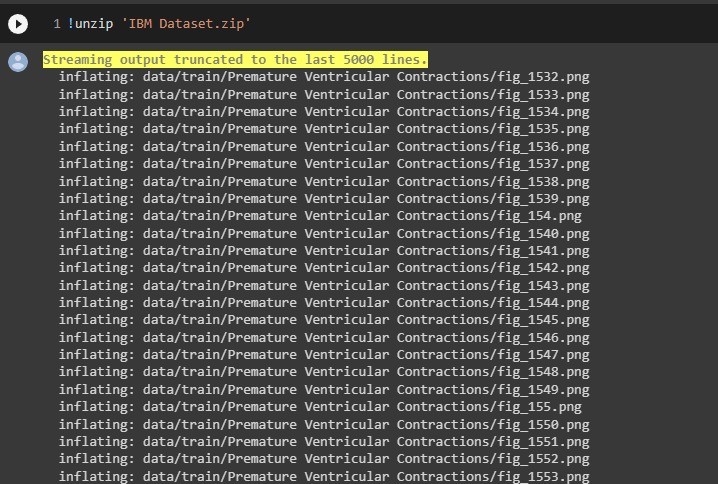
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Sprin t3 | Test the Models | US N2 | Test the model through Loaded necessary libraries, the model is evaluated for accurate results. | 10 | Medi um | Prasanth A  Rithan  Bharath  Naveensridhar  Prasanth R |
| Sprin t4 | Register in IBM cloud | US N1 | Register in IBM Cloud | 10 | Medi um | Prasanth A  Rithan  Bharath  Naveensridhar  Prasanth R |
| Sprin t4 | Train the model on IBM | US N2 | Train the model on IBM | 10 | High | Prasanth A  Rithan  Bharath  Naveensridhar  Prasanth R |

* 1. Sprint Delivery Schedule

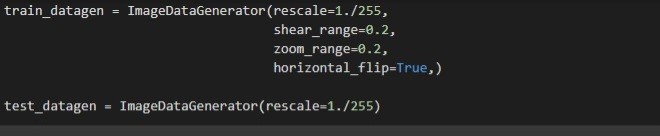
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sprint** | **Total Story Points** | **Durati on** | **Sprint start date** | **Sprint End date** | **Story Points comple ted** | **Sprint Release Date** |
| Sprint 1 | 20 | 6 Days | 24 Oct 2022 | 29 Oct 2022 | 20 | 29 Oct 2022 |
| Sprint 2 | 20 | 6 Days | 31 Oct 2022 | 05 Nov 2022 | 20 | 05 Nov 2022 |
| Sprint 3 | 20 | 6 Days | 07 Nov 2022 | 12 Nov 2022 | 20 | 12 Nov 2022 |
| Sprint 4 | 20 | 6 Days | 14 Nov 2022 | 19 Nov 2022 | 20 | 19 Nov 2022 |

# CODING & SOLUTIONING

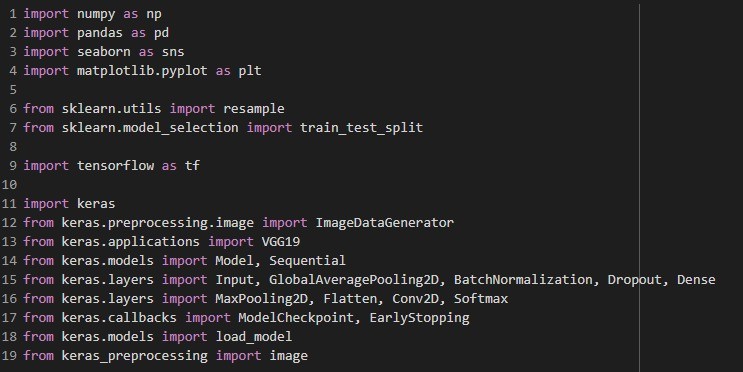
* 1. Download dataset



* 1. Image Preprocessing



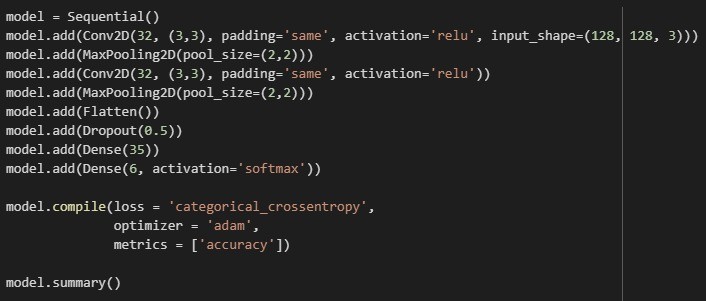
* 1. Import Libraries

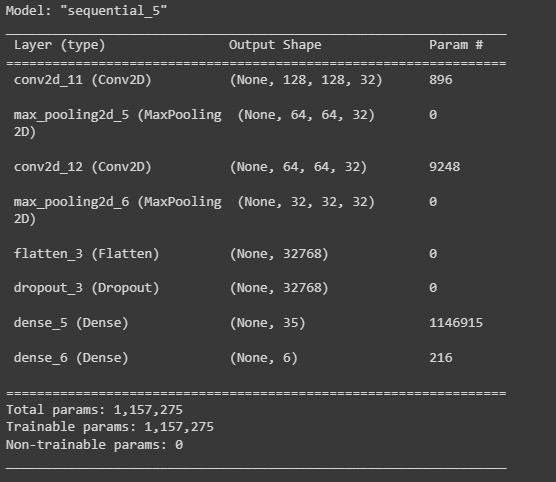


* 1. Configure learning process

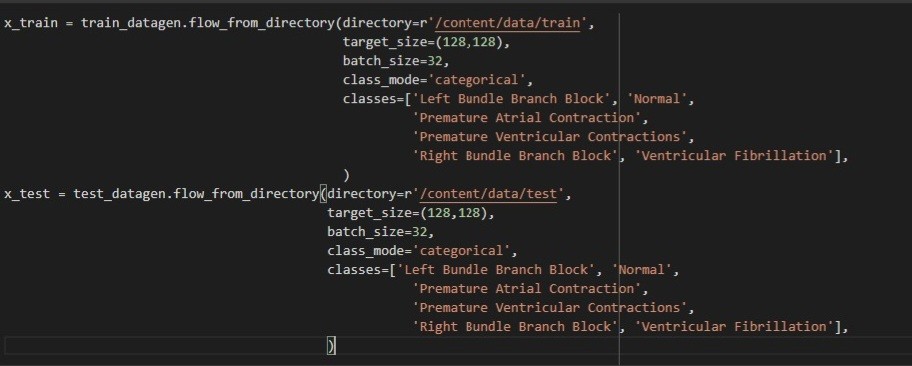


* 1. Adding layers

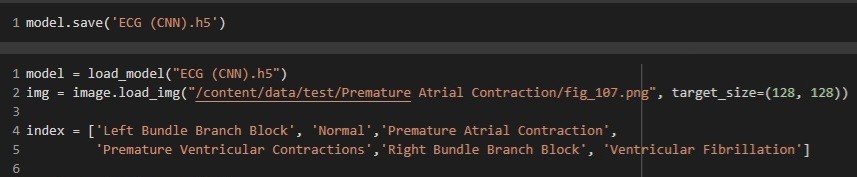




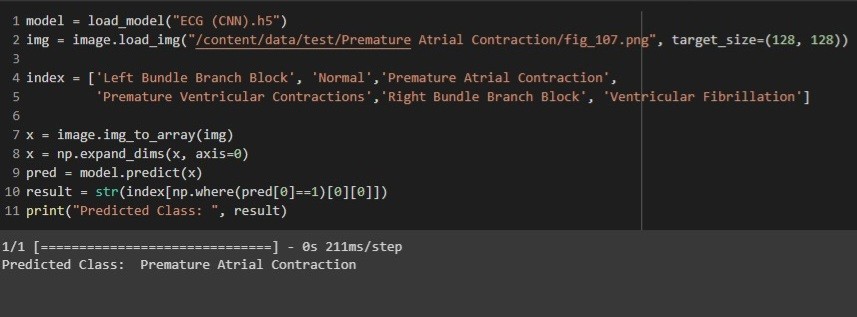
* 1. Train Model



* 1. Save Model



* 1. Test Model



* 1. flask python code

import os

import numpy as np

from flask import Flask,request,render\_template from keras.models import load\_model

import tensorflow as tf from PIL import Image

app=Flask( name ) model=load\_model('ECG.h5')

@app.route("/") def about():

return render\_template("home.html")

@app.route("/home") def home():

return render\_template("home.html")

@app.route("/info") def info():

return render\_template("info.html")

@app.route("/guide") def guide():

return render\_template("guide.html")

@app.route("/predict") def test():

return render\_template("predict.html")

@app.route("/predict",methods=["GET","POST"]) def upload():

if request.method=='POST': f=request.files['file'] basepath=os.path.dirname(' file ')

filepath=os.path.join(basepath,"uploads",f.filename) f.save(filepath)

img=tf.keras.utils.load\_img(filepath,target\_size=(128,128)) x=tf.keras.utils.img\_to\_array(img) x=np.expand\_dims(x,axis=0)

pred=model.predict(x) y\_pred = np.argmax(pred) print("prediction",y\_pred)

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

if name ==" main ": app.run(debug=False)

* 1. User Interface
     1. Home.html

<!DOCTYPE html>

<html>

<head>

<title>Home</title>

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="../static/css/index.css">

<link href="{{ url\_for('static', filename='css/index.css') }}" rel="stylesheet">

</head>

<body>

<div class="navbar">

<a href="/predict" >PREDICT</a>

<a href="/info">INFO</a>

</div>

<div>

<center><h2 class="header">ARRHYTHMIA PREDICTION</h2></center>

<br>

<center>

<b class="pd">

<font color = "#ffec78" size="13" font-family = "Helvetica"> ECG arrhythmia classification using CNN

</font>

</b>

</center>

</div> <br>

<center>

<p>

<font color = "white">

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.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.

</font>

</p>

</center>

</body>

</html>

* + 1. Info.html

<!DOCTYPE html>

<html>

<head>

<style> img{ width:20%; height:10%;

padding:20px; margin-top:5px;

}

</style>

<title>

Info

</title>

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="../static/css/index.css">

<link href="{{ url\_for('static', filename='css/index.css') }}" rel="stylesheet">

</head>

<body>

<div class="navbar">

<a href="/predict" >PREDICT</a>

<a href="/guide">ECG</a>

<a href="/home">HOME</a><br>

</div>

<div class="container" >

<p>

<h2>

Types of Arrhythmia

</h2>

<b>

<font color = "#bb8206"> Supraventricular arrhythmias :

</font>

</b>

Arrhythmias that begin in the atria (the heart's upper chambers). "Supra” means

above.

“Ventricular” refers to the lower chambers of the heart or ventricles. <br>

<b>

<font color = "#bb8206"> Ventricular arrhythmias :

</font>

</b>

Arrhythmias that begin in the ventricles (the heart's lower chambers).<br>

<b>

<font color = "#bb8206"> Bradyarrhythmias :

</font>

</b>

Slow heart rhythms that may be caused by disease in the heart's conduction

system,

such as the sinoatrial (SA) node, atrioventricular (AV) node or HIS-Purkinje network.<br>

</p>

<h2>

Symptoms of Arrhythmia

</h2>

<ol style="display:inline-table;">

<li>

A feeling of skipped heartbeat or that your heart is “running away,” fluttering or doing "flip-flops.

</li>

<li>

Pounding in your chest.

</li>

<li>

Dizziness or feeling lightheaded.

</li>

<li>

Shortness of breath.

</li>

<li>

Chest discomfort.

</li>

<li>

Weakness or fatigue (feeling very tired).

</li>

<li>

Weakening of the heart muscle or low ejection fraction.

</li>

</ol>

<h2>

Causes of Arrhythmia

</h2>

<ol style="display:inline-table;">

<li>

Coronary artery disease.

</li>

<li>

Irritable tissue in the heart (due to genetic or acquired causes).

</li>

<li>

High blood pressure.

</li>

<li>

Changes in the heart muscle (cardiomyopathy).

</li>

<li>

Valve disorders.

</li>

<li>

Electrolyte imbalances in your blood, such as sodium or potassium imbalances.

</li>

<li>

Injury from a heart attack.

</li>

<li>

The healing process after heart surgery.

</li>

<li>

Other medical conditions.

</li>

</ol>

</div>

</body>

</html>

* + 1. Predict\_base.html

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Predict</title>

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="../static/css/index.css">

<link href="{{ url\_for('static', filename='css/index.css') }}" rel="stylesheet">

</head>

<body>

<div class="navbar">

<a href="/guide">ECG</a>

<a href="/info">INFO</a>

<a href="/home">HOME</a><br>

</div>

<div class="container">

<center>

<div id="content" style="margin-top:2em">

{% block content %}{% endblock %}

</div>

</center>

</div>

</body>

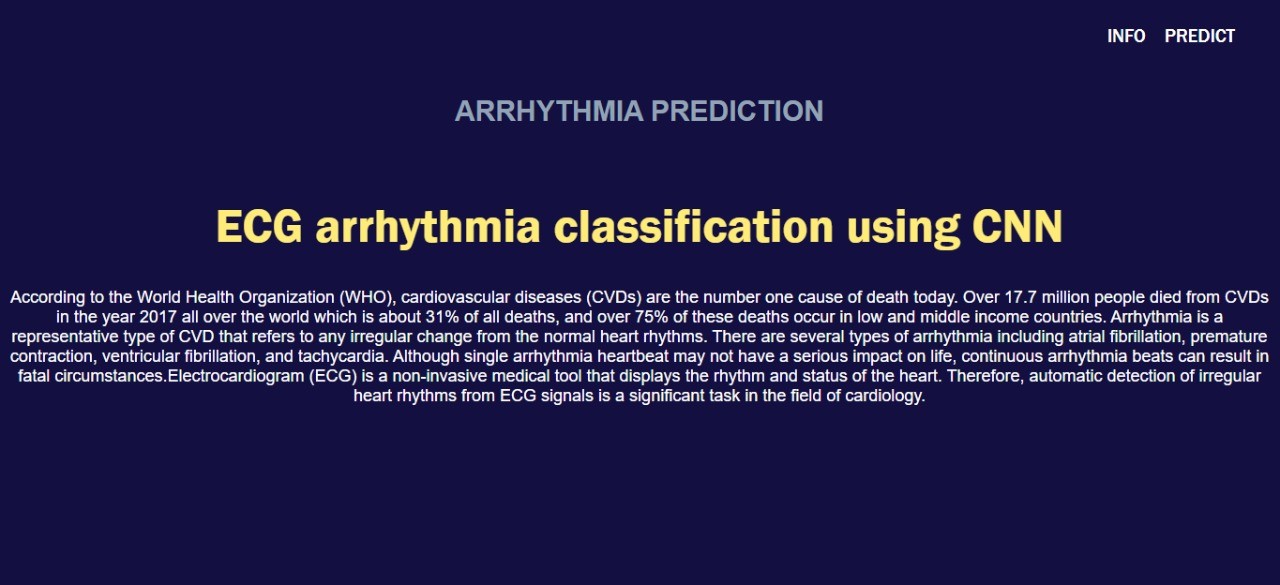
<footer>

<script src="{{ url\_for('static', filename='js/main.js') }}" type="text/javascript"></script>

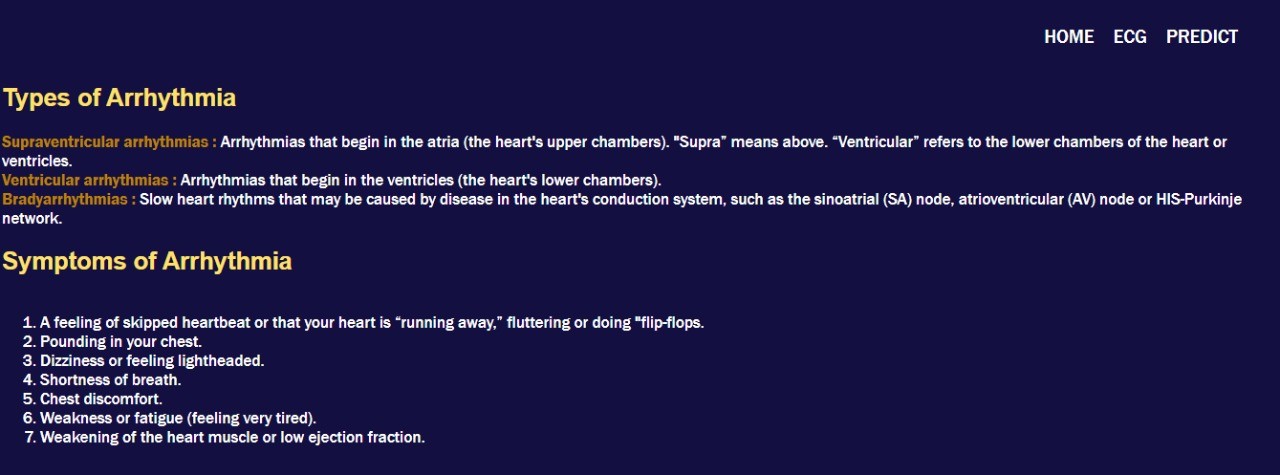
</footer>

</html>

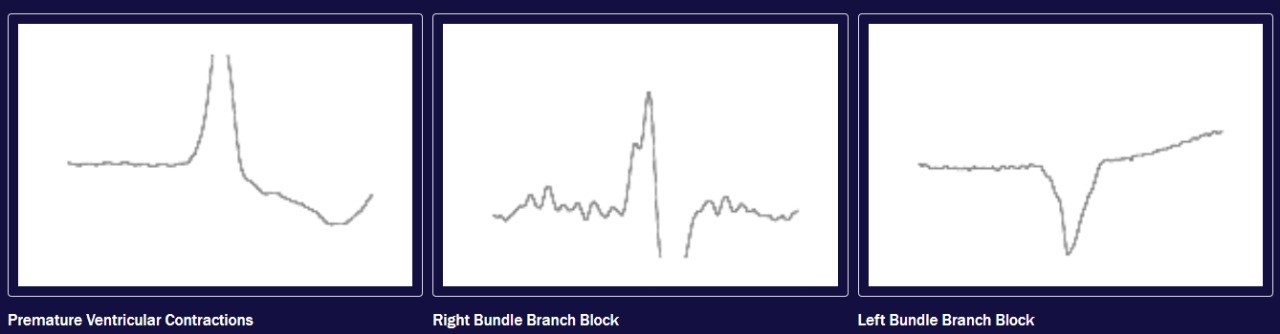
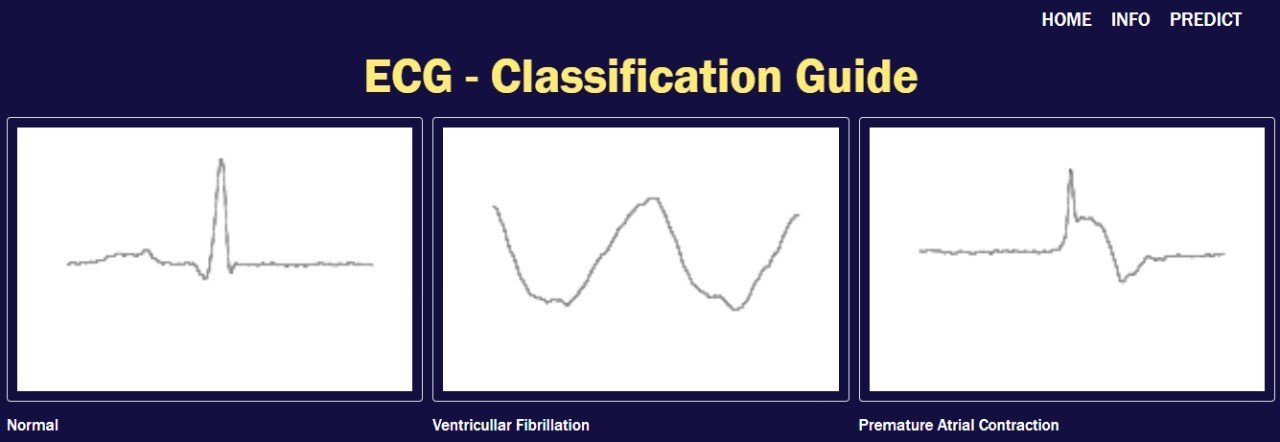
* 1. Run Application Home.html



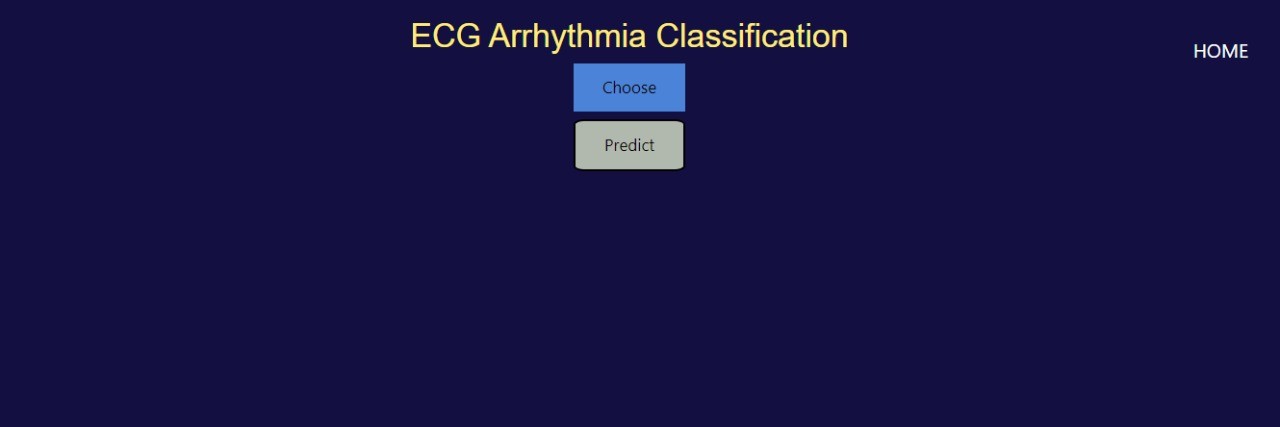
Info.html



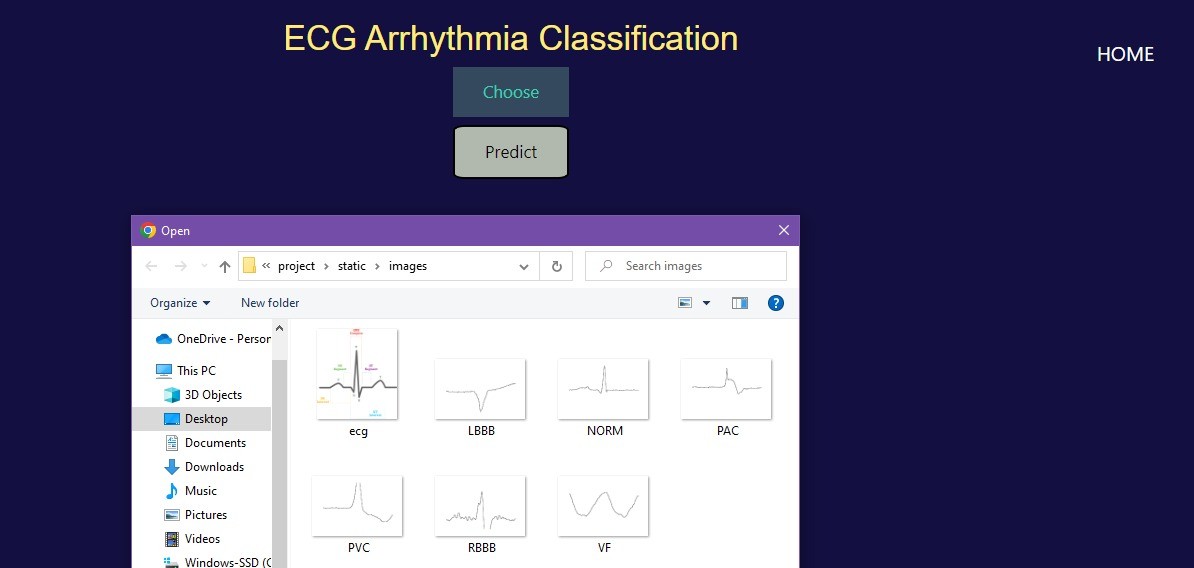
Guide.html



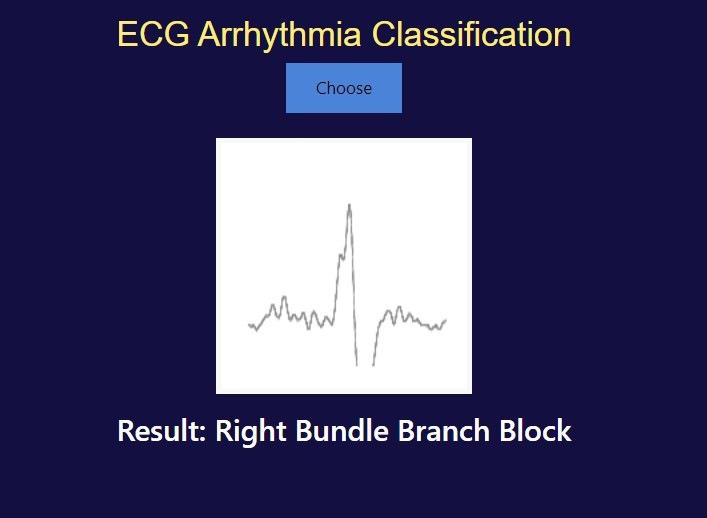
Predict.html



Upload file

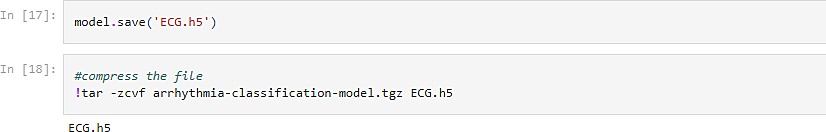


Predicting output and displaying

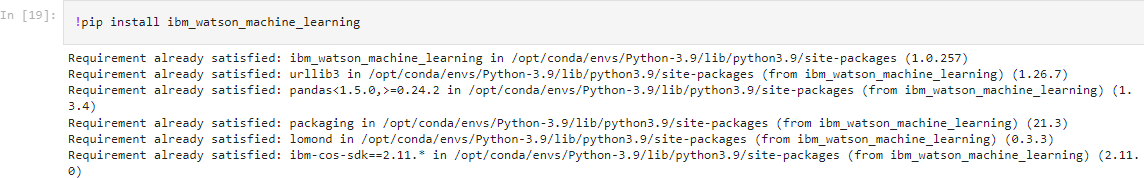


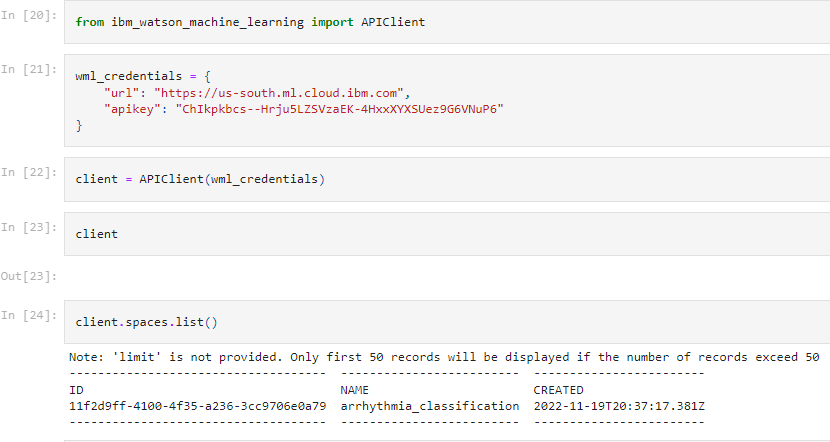
* 1. Train Model on IBM Watson

Compress the file

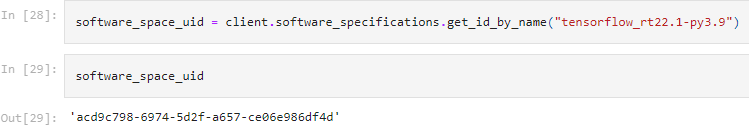


Deploying model in IBM Watson









# TESTING

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test case ID** | **Feature Type** | **Test Scenario** | **Steps to execute** | **Expected Result** | **Actual Result** | **Stat us** |
| TC\_ | Function | Verify user | 1.Enter URL and | Predicted | Worki | Pass |
| 001 | al | is able to | click go | Result | ng as |  |
|  |  | access the | 2.Click upload | Should | expect |  |
|  |  | landing | button | Display | ed |  |
|  |  | page | 3.Choose a image |  |  |  |
|  |  |  | from local directory |  |  |  |
|  |  |  | or paste or drop |  |  |  |
|  |  |  | 4.Click predict to |  |  |  |
|  |  |  | view result |  |  |  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| TC\_ | UI | Verify the | 1.Sliding banner | Application | Worki | Pass |
| 002 |  | UI | 2.buttons | should show | ng as |  |
|  |  | elements |  | upload and | expect |  |
|  |  |  |  | predict | ed |  |
|  |  |  |  | button |  |  |
| TC\_ | Function | Verify | 1.Enter URL and | User should | Worki | Pass |
| 003 | al | whether | click go | observe | ng as |  |
|  |  | the link is | 2.Type or copy paste | whether the | expect |  |
|  |  | legitimate | URL | website is | ed |  |
|  |  | or not | 3.Check the website | legitimate |  |  |
|  |  |  | is legitimate or not | or not |  |  |
|  |  |  | 4.Observe the results |  |  |  |
| TC\_ | Function | Verify user | 1.Enter URL and | Application | Worki | Pass |
| 004 | al | is able to | click go | should show | ng as |  |
|  |  | access the | 2.Type or copy paste | that safe | expect |  |
|  |  | legitimate | URL | webpage or | ed |  |
|  |  | website or | 3.Check the website | unsafe |  |  |
|  |  | not | is legitimate or not |  |  |  |
|  |  |  | 4.Continue if the |  |  |  |
|  |  |  | website is legitimate |  |  |  |
|  |  |  | or be cautious if it is |  |  |  |
|  |  |  | not legitimate |  |  |  |
| TC\_ | Function | Testing | 1.Enter URL and | User can | Worki | Pass |
| 005 | al | website | click go | able to | ng as |  |
|  |  | with | 2.Type or copy paste | identify the | expect |  |
|  |  | multiple | URL | websites | ed |  |
|  |  | url | 3.Check the website | whether it is |  |  |
|  |  |  | is legitimate or not | secure or |  |  |
|  |  |  | 4.Continue if the | not |  |  |
|  |  |  | website is Secure or |  |  |  |
|  |  |  | be cautious if it is not |  |  |  |
|  |  |  | Secure |  |  |  |

1. **RESULTS**
   1. Performance Metrics

|  |  |  |
| --- | --- | --- |
| **SNO** | **MODEL** | **VALIDATION ACCURACY** |
| 1. | Xception | 82.62% |
| 2. | ResNet50 | 66.6% |
| 3. | VGG19 | 87.5% |
| 4. | CNN | 87.9% |

# ADVANTAGES & DISADVANTAGES

* 1. Advantages
     + The proposed model predicts Arrhythmia in images with a high accuracy rate of nearly 96%.
     + The early detection of Arrhythmia gives better understanding of disease causes,

initiates therapeutic interventions and enables developing appropriate treatments.

* 1. Disadvantages
     + Not useful for identifying the different stages of Arrhythmia disease.
     + Not useful in monitoring motor symptoms

# CONCLUSION

Cardiovascular disease is a major health problem in today's world. The early diagnosis of cardiac arrhythmia highly relies on the ECG. Unfortunately, the expert level of medical resources is rare, visually identifying the ECG signal is challenging and time- consuming. The advantages of the proposed CNN network have been put to evidence. It is endowed with an ability to effectively process the non-filtered dataset with its potential anti-noise features. Besides that, ten-fold cross-validation is implemented in this work to further demonstrate the robustness of the network.

# FUTURE SCOPE

For future work, it would be interesting to explore the use of optimization techniques to find a feasible design and solution. The limitation of our study is that we have yet to apply any optimization techniques to optimize the model parameters and we believe that with the implementation of the optimization, it will be able to further elevate the performance of the proposed solution to the next level.

# APPENDIX

Source code

->CNN.ipynb file

**import** numpy **as** np **import** pandas **as** pd **import** seaborn **as** sns

**import** matplotlib.pyplot **as** plt

**from** sklearn.utils **import** resample

**from** sklearn.model\_selection **import** train\_test\_split

**import** tensorflow **as** tf

**import** keras

**from** keras.preprocessing.image **import** ImageDataGenerator

**from** keras.applications **import** VGG19

**from** keras.models **import** Model, Sequential

**from** keras.layers **import** Input, GlobalAveragePooling2D, BatchNormalization,

Dropout, Dense

**from** keras.layers **import** MaxPooling2D, Flatten, Conv2D, Softmax

**from** keras.callbacks **import** ModelCheckpoint, EarlyStopping

**from** keras.models **import** load\_model

**from** keras\_preprocessing **import** image

!unzip 'IBM Dataset.zip'

train\_datagen **=** ImageDataGenerator(rescale**=**1.**/**255,

shear\_range**=**0.2, zoom\_range**=**0.2, horizontal\_flip**=True**,)

test\_datagen **=** ImageDataGenerator(rescale**=**1.**/**255)

x\_train **=** train\_datagen**.**flow\_from\_directory(directory**=**r'/content/data/train', target\_size**=**(128,128),

batch\_size**=**32, class\_mode**=**'categorical',

classes**=**['Left Bundle Branch Block', 'Normal', 'Premature Atrial Contraction', 'Premature Ventricular Contractions',

'Right Bundle Branch Block', 'Ventricular Fibrillation'],

)

x\_test **=** test\_datagen**.**flow\_from\_directory(directory**=**r'/content/data/test', target\_size**=**(128,128),

batch\_size**=**32, class\_mode**=**'categorical',

classes**=**['Left Bundle Branch Block', 'Normal', 'Premature Atrial Contraction', 'Premature Ventricular Contractions',

'Right Bundle Branch Block', 'Ventricular Fibrillation'],

tf**.**config**.**optimizer**.**set\_jit(**True**) model **=** Sequential()

model**.**add(Conv2D(32, (3,3), padding**=**'same', activation**=**'relu', input\_shape**=**(128, 128,

3)))

model**.**add(MaxPooling2D(pool\_size**=**(2,2))) model**.**add(Conv2D(32, (3,3), padding**=**'same', activation**=**'relu')) model**.**add(MaxPooling2D(pool\_size**=**(2,2))) model**.**add(Flatten())

model**.**add(Dropout(0.5)) model**.**add(Dense(35)) model**.**add(Dense(6, activation**=**'softmax'))

model**.**compile(loss **=** 'categorical\_crossentropy', optimizer **=** 'adam',

metrics **=** ['accuracy'])

model**.**summary()

model\_history **=** model**.**fit(x\_train,

steps\_per\_epoch**=**len(x\_train), epochs**=**25, validation\_data**=**x\_test, validation\_steps**=**len(x\_test),)

model**.**save('ECG (CNN).h5')

model **=** load\_model("ECG (CNN).h5")

img **=** image**.**load\_img("/content/data/test/Premature Atrial Contraction/fig\_107.png", target\_size**=**(128, 128))

index **=** ['Left Bundle Branch Block', 'Normal','Premature Atrial Contraction', 'Premature Ventricular Contractions','Right Bundle Branch Block', 'Ventricular

Fibrillation']

x **=** image**.**img\_to\_array(img) x **=** np**.**expand\_dims(x, axis**=**0) pred **=** model**.**predict(x)

result **=** str(index[np**.**where(pred[0]**==**1)[0][0]])

print("Predicted Class: ", result)

-> flask python code

import os

import numpy as np

from flask import Flask,request,render\_template from keras.models import load\_model

import tensorflow as tf from PIL import Image

app=Flask( name ) model=load\_model('ECG.h5')

@app.route("/") def about():

return render\_template("home.html")

@app.route("/home") def home():

return render\_template("home.html")

@app.route("/info") def info():

return render\_template("info.html")

@app.route("/guide") def guide():

return render\_template("guide.html")

@app.route("/predict") def test():

return render\_template("predict.html")

@app.route("/predict",methods=["GET","POST"]) def upload():

if request.method=='POST': f=request.files['file'] basepath=os.path.dirname(' file ')

filepath=os.path.join(basepath,"uploads",f.filename) f.save(filepath)

img=tf.keras.utils.load\_img(filepath,target\_size=(128,128)) x=tf.keras.utils.img\_to\_array(img) x=np.expand\_dims(x,axis=0)

pred=model.predict(x) y\_pred = np.argmax(pred) print("prediction",y\_pred)

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

if name ==" main ": app.run(debug=False)

->User Interface:

Home.html

<!DOCTYPE html>

<html>

<head>

<title>Home</title>

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="../static/css/index.css">

<link href="{{ url\_for('static', filename='css/index.css') }}" rel="stylesheet">

</head>

<body>

<div class="navbar">

<a href="/predict" >PREDICT</a>

<a href="/info">INFO</a>

</div>

<div>

<center><h2 class="header">ARRHYTHMIA PREDICTION</h2></center>

<br>

<center>

<b class="pd">

<font color = "#ffec78" size="13" font-family = "Helvetica"> ECG arrhythmia classification using CNN

</font>

</b>

</center>

</div> <br>

<center>

<p>

<font color = "white">

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.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.

</font>

</p>

</center>

</body>

</html>

Info.html

<!DOCTYPE html>

<html>

<head>

<style> img{

width:20%; height:10%; padding:20px; margin-top:5px;

}

</style>

<title>

Info

</title>

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="../static/css/index.css">

<link href="{{ url\_for('static', filename='css/index.css') }}" rel="stylesheet">

</head>

<body>

<div class="navbar">

<a href="/predict" >PREDICT</a>

<a href="/guide">ECG</a>

<a href="/home">HOME</a><br>

</div>

<div class="container" >

<p>

<h2>

Types of Arrhythmia

</h2>

<b>

<font color = "#bb8206"> Supraventricular arrhythmias :

</font>

</b>

Arrhythmias that begin in the atria (the heart's upper chambers). "Supra” means

above.

“Ventricular” refers to the lower chambers of the heart or ventricles. <br>

<b>

<font color = "#bb8206"> Ventricular arrhythmias :

</font>

</b>

Arrhythmias that begin in the ventricles (the heart's lower chambers).<br>

<b>

<font color = "#bb8206"> Bradyarrhythmias :

</font>

</b>

Slow heart rhythms that may be caused by disease in the heart's conduction system,

such as the sinoatrial (SA) node, atrioventricular (AV) node or HIS-Purkinje network.<br>

</p>

<h2>

Symptoms of Arrhythmia

</h2>

<ol style="display:inline-table;">

<li>

A feeling of skipped heartbeat or that your heart is “running away,” fluttering or doing "flip-flops.

</li>

<li>

Pounding in your chest.

</li>

<li>

Dizziness or feeling lightheaded.

</li>

<li>

Shortness of breath.

</li>

<li>

Chest discomfort.

</li>

<li>

Weakness or fatigue (feeling very tired).

</li>

<li>

Weakening of the heart muscle or low ejection fraction.

</li>

</ol>

<h2>

Causes of Arrhythmia

</h2>

<ol style="display:inline-table;">

<li>

Coronary artery disease.

</li>

<li>

Irritable tissue in the heart (due to genetic or acquired causes).

</li>

<li>

High blood pressure.

</li>

<li>

Changes in the heart muscle (cardiomyopathy).

</li>

<li>

Valve disorders.

</li>

<li>

Electrolyte imbalances in your blood, such as sodium or potassium imbalances.

</li>

<li>

Injury from a heart attack.

</li>

<li>

The healing process after heart surgery.

</li>

<li>

Other medical conditions.

</li>

</ol>

</div>

</body>

</html>

Predict\_base.html

<!DOCTYPE html>

<html>

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Predict</title>

<meta name="viewport" content="width=device-width, initial-scale=1">

<link rel="stylesheet" href="../static/css/index.css">

<link href="{{ url\_for('static', filename='css/index.css') }}" rel="stylesheet">

</head>

<body>

<div class="navbar">

<a href="/guide">ECG</a>

<a href="/info">INFO</a>

<a href="/home">HOME</a><br>

</div>

<div class="container">

<center>

<div id="content" style="margin-top:2em">

{% block content %}{% endblock %}

</div>

</center>

</div>

</body>

<footer>

<script src="{{ url\_for('static', filename='js/main.js') }}" type="text/javascript"></script>

</footer>

</html>

Index.css

.header { padding: 30px;

text-align: center;

font-family: Arial, Helvetica, sans-serif; color: #93a3b6;

font-size: 30px;

}

body{

color: white; background: #130f40;

}

h2{

font-family: Arial, Helvetica, sans-serif; color: #ffe169;

}

p{

color:white;

font-family:Arial, Helvetica, sans-serif; font-size: 18px;

}

.navbar{ margin : 0px;

padding : 20px; padding-bottom: 20px; opacity : 5;

}

a:hover{

background-color: #ffe169; border-radius:10px; padding-left:20px; color:black;

}

a{

color:white; font-size: 20px;

text-align: center; text-decoration: none; float:right;

padding-right:20px;

}

.img-preview { width: 256px; height: 256px; position: relative;

border: 5px solid #F8F8F8;

box-shadow: 0px 2px 4px 0px rgba(0, 0, 0, 0.1); margin-top: 1em;

margin-bottom: 1em;

}

.img-preview>div { width: 100%;

height: 100%;

background-size: 256px 256px; background-repeat: no-repeat; background-position: center;

}

input[type="file"] { display: none;

}

.upload-label{

display: inline-block; padding: 12px 30px; background: #39D2B4; color: #fff;

font-size: 1em; transition: all .4s; cursor: pointer;

}

.upload-label:hover{ background: #34495E; color: #39D2B4;

}

.loader {

border: 8px solid #f3f3f3; border-top: 8px solid #ffec78; border-radius: 70%;

width: 25px; height: 25px;

animation: spin 1s linear infinite;

}

@keyframes spin {

0% { transform: rotate(0deg); } 100% { transform: rotate(360deg); }

}

GitHub & Project Demo Link

**Github Link:** <https://github.com/IBM-EPBL/IBM-Project-4249-1658725786>

**Project Demo Link: https://drive.google.com/file/d/12aqxRVgUeILN8uxQLqpttncQ-VMGL\_zi/view?usp=share\_link**