Application Building

Team ID - PNT2022TMID29524

Team Leader – Dhanushwaran.S

Team Member – Arun.B

Team Member – Naveen Raj.S

Team Member – Akash.A

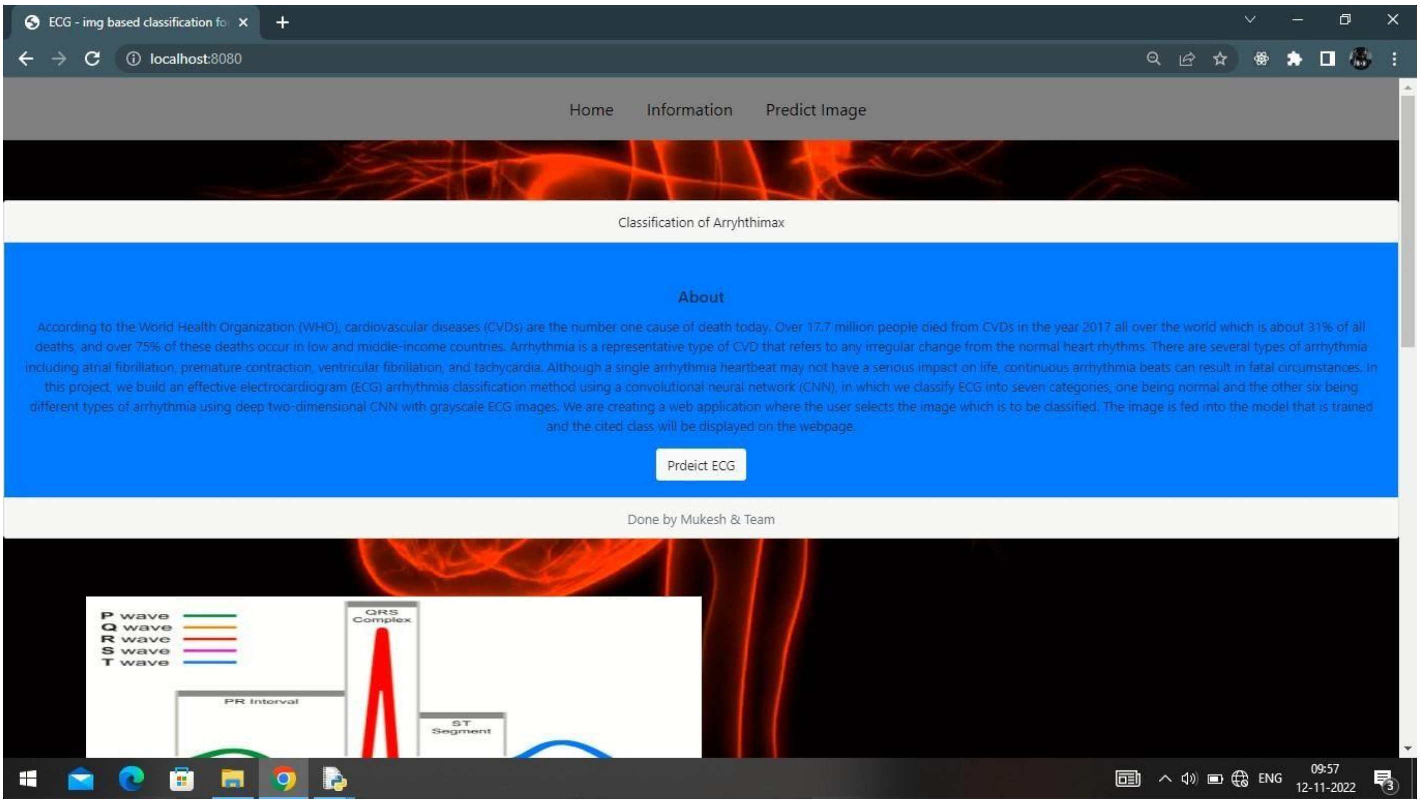
1.Create HTML Files

We use HTML to create the front end part of the web page.

Here, we created 4 html pages- upload.html, predict.html, home.html, info.html. home.html displays the home page.

Info.html displays all important details to be known about ECG.

upload.html and predict.html accept input from the user and predicts the values. **Home.html looks Like this**



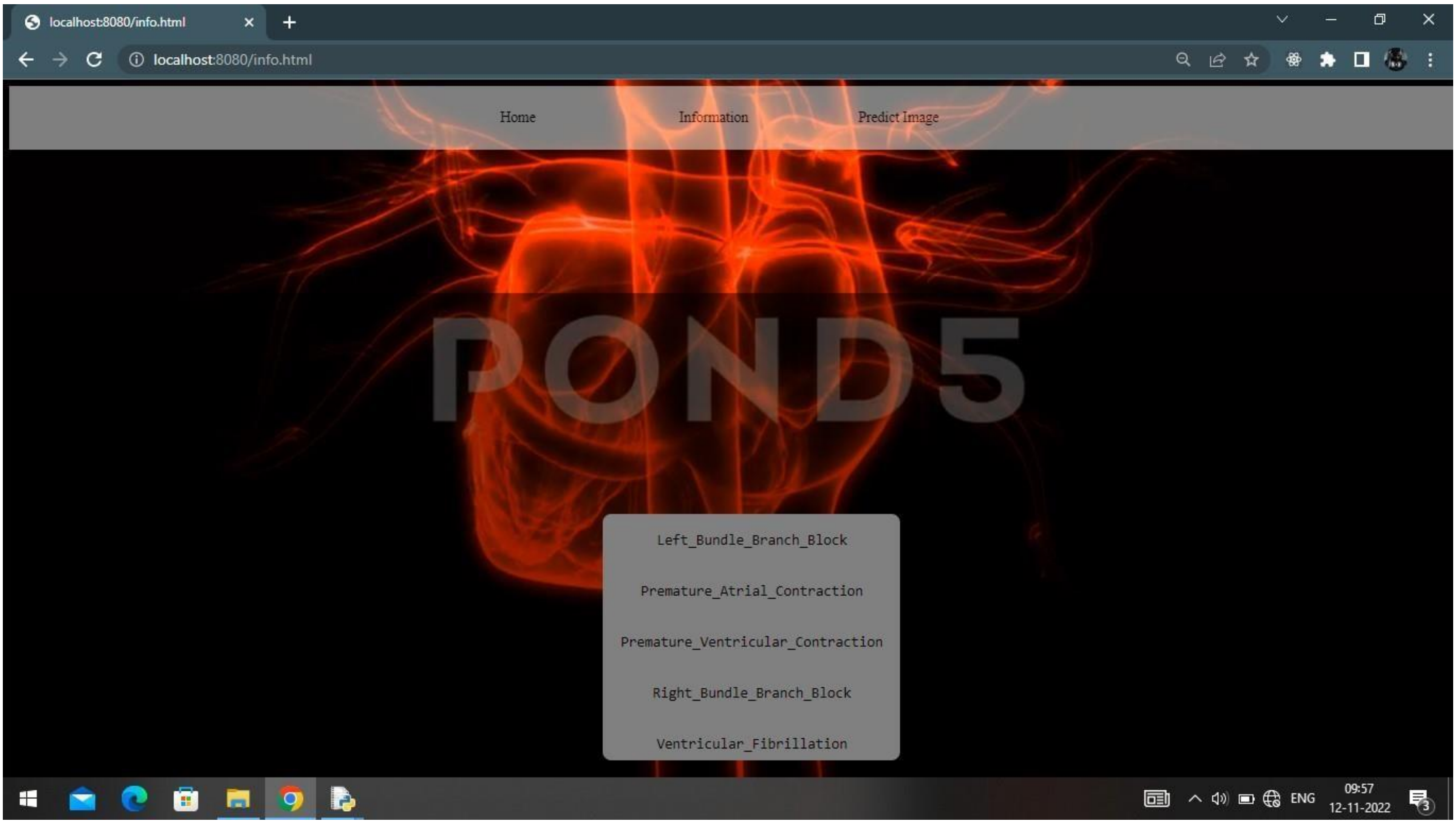
Classification of Arryhthimax

About

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 a single arrhythmia heartbeat may not have a serious impact on life, continuous arrhythmia beats can result in fatal circumstances. In this project, we build an effective electrocardiogram (ECG) arrhythmia classification method using a convolutional neural network (CNN), in which 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. 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.

Done by Mukesh & Team

Info.html looks like this



Left_Bundle_Branch_Block

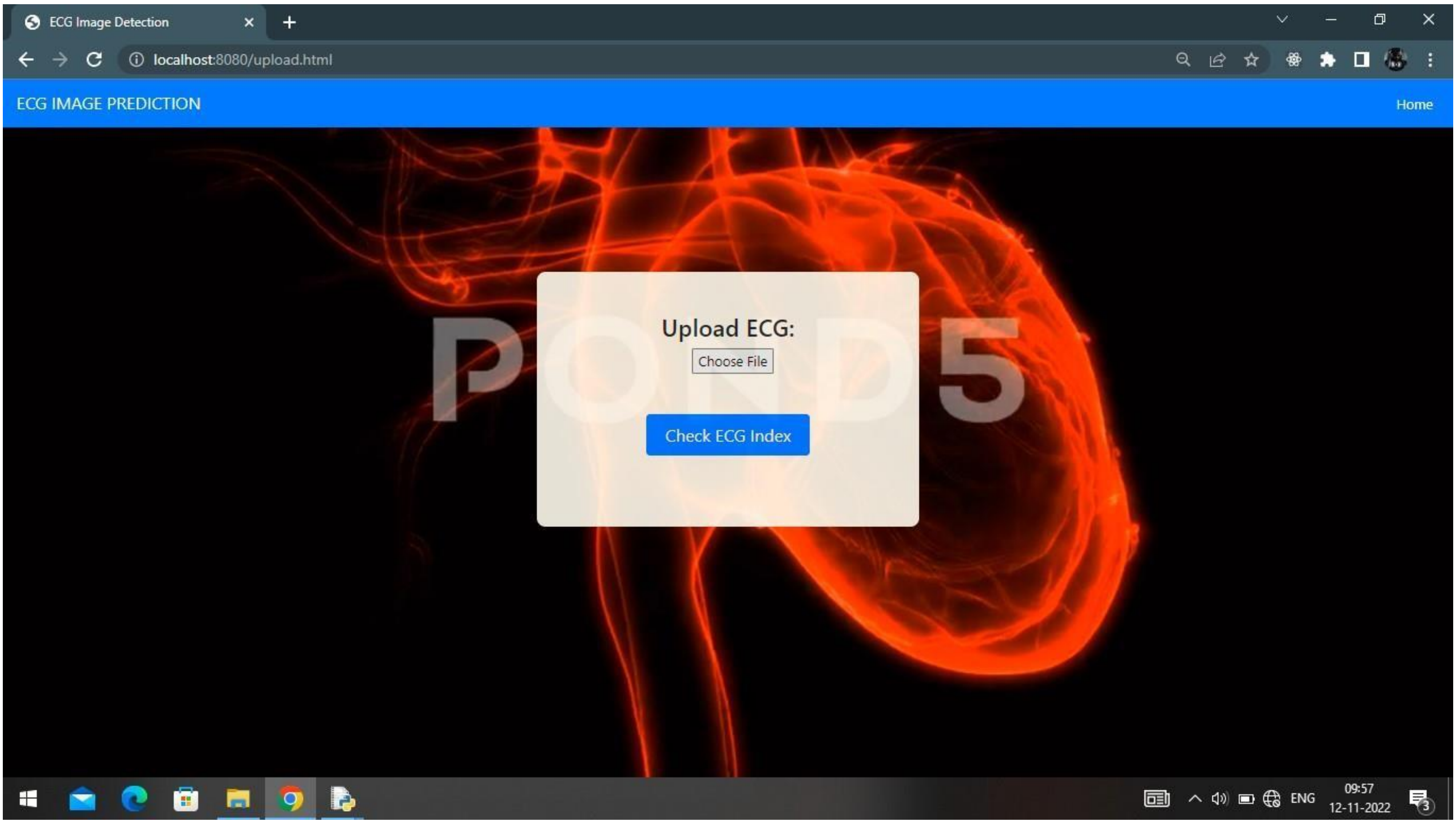
Premature_Atrial_Contraction

Premature_Ventricular_Contraction

Right_Bundle_Branch_Block

Ventricular_Fibrillation

upload.html look like this



2.Build Python Code

Let us build the flask file ‘app.py’ which is a web framework written in python for server-side scripting.

Let’s see step by step procedure for building the backend application.

1 . The app starts running when the "__name__" constructor is called in main.

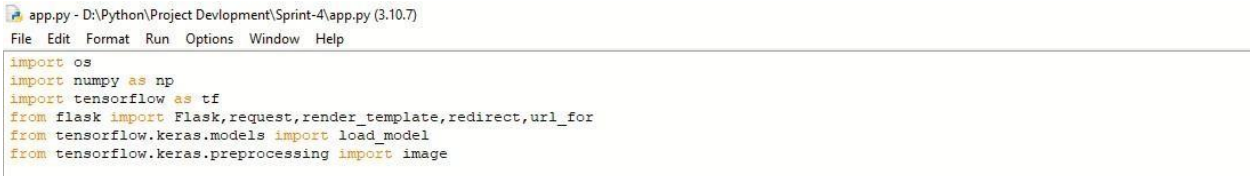
2.render_template is used to return HTML file.

3."GET" method is used to take input from the user.

4."POST" method is used to display the output to the user.

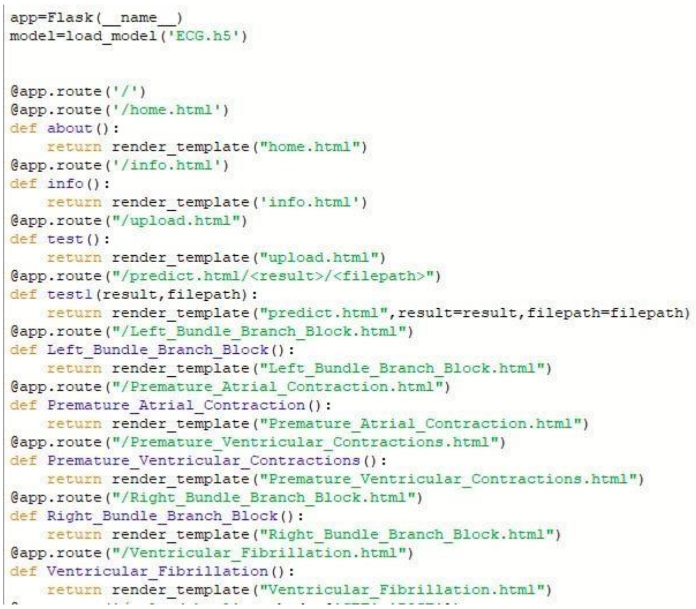
App.py

Import the libraries


app.py - D:\Python\Project Devlopment\Sprint-4\app.py (3.10.7)
File Edit Format Run Options Window Help

```
import os
import numpy as np
import tensorflow as tf
from flask import Flask,request,render_template,redirect,url_for
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
```

Routing to the HTML Page



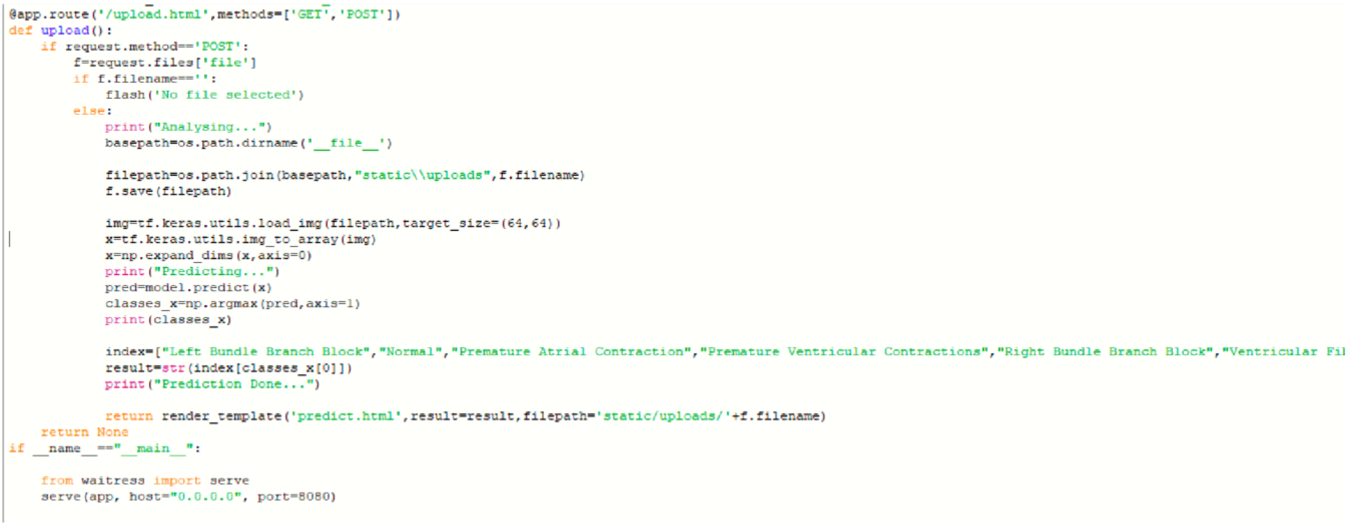
```
app=Flask(__name__)
model=load_model('ECG.h5')


@app.route('/')
@app.route('/home.html')
def about():
    return render_template("home.html")
@app.route('/info.html')
def info():
    return render_template('info.html')
@app.route("/upload.html")
def test():
    return render_template("upload.html")
@app.route("/predict.html/<result>/<filepath>")
def test1(result,filepath):
    return render_template("predict.html",result=result,filepath=filepath)
@app.route("/Left_Bundle_Branch_Block.html")
def Left_Bundle_Branch_Block():
    return render_template("Left_Bundle_Branch_Block.html")
@app.route("/Premature_Atrial_Contraction.html")
def Premature_Atrial_Contraction():
    return render_template("Premature_Atrial_Contraction.html")
@app.route("/Premature_Ventricular_Contractions.html")
def Premature_Ventricular_Contractions():
    return render_template("Premature_Ventricular_Contractions.html")
@app.route("/Right_Bundle_Branch_Block.html")
def Right_Bundle_Branch_Block():
    return render_template("Right_Bundle_Branch_Block.html")
@app.route("/Ventricular_Fibrillation.html")
def Ventricular_Fibrillation():
    return render_template("Ventricular_Fibrillation.html")
```

Showcasing prediction on UI

When the image is uploaded, it predicts the category of uploaded the image is either

'Left Bundle Branch Block', 'Normal', 'Premature Atrial Contraction', 'Premature Ventricular Contractions',

'Right Bundle Branch Block', 'Ventricular Fibrillation'. If the image predicts value as 0, then it is



```
@app.route('/upload.html',methods=['GET','POST'])
def upload():
    if request.method=='POST':
        f=request.files['file']
        if f.filename=='':
            flash('No file selected')
        else:
            print("Analysing...")
            basepath=os.path.dirname('__file__')

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

            img=tf.keras.utils.load_img(filepath,target_size=(64,64))
            x=tf.keras.utils.img_to_array(img)
            x=np.expand_dims(x,axis=0)
            print("Predicting...")
            pred=model.predict(x)
            classes_x=np.argmax(pred,axis=1)
            print(classes_x)

            index=["Left Bundle Branch Block","Normal","Premature Atrial Contraction","Premature Ventricular Contractions","Right Bundle Branch Block","Ventricular Fil
            result=str(index[classes_x[0]])
            print("Prediction Done...")

            return render_template('predict.html',result=result,filepath='static/uploads/'+f.filename)
    return None
if __name__=="__main__":

    from waitress import serve
    serve(app, host="0.0.0.0", port=8080)
```

displayed as “Left Bundle Branch”. Similarly, if the predicted value is 1, it displays “Normal” as output and so on.

Run The APP

Open Python IDLE from the start menu

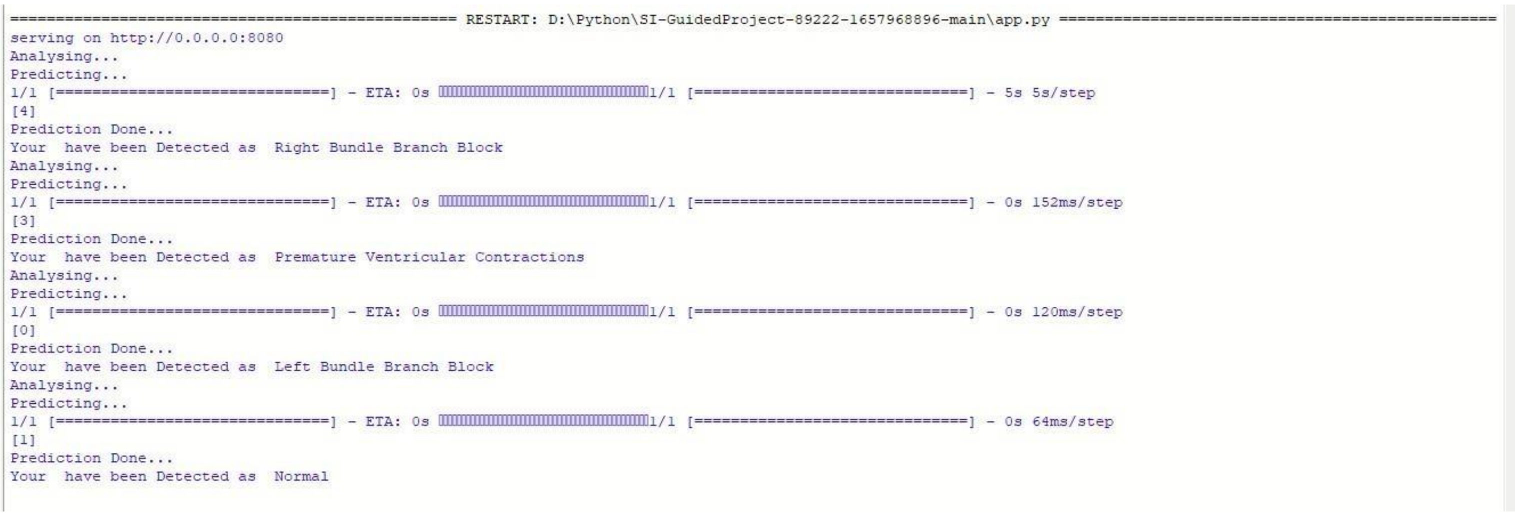
Navigate to the folder where your python script is.

Now type app.py” command

Navigate to the localhost where you can view your web page

Then it will run on localhost:8080

Navigate to the localhost:8080 where you can view your web page.



```
================================================= RESTART: D:\Python\SI-GuidedProject-89222-1657968896-main\app.py =================================================
serving on http://0.0.0.0:8080
Analysing...
Predicting...
1/1 [==============================] - ETA: 0s ▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯1/1 [==============================] - 5s 5s/step
[4]
Prediction Done...
Your  have been Detected as  Right Bundle Branch Block
Analysing...
Predicting...
1/1 [==============================] - ETA: 0s ▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯1/1 [==============================] - 0s 152ms/step
[3]
Prediction Done...
Your  have been Detected as  Premature Ventricular Contractions
Analysing...
Predicting...
1/1 [==============================] - ETA: 0s ▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯1/1 [==============================] - 0s 120ms/step
[0]
Prediction Done...
Your  have been Detected as  Left Bundle Branch Block
Analysing...
Predicting...
1/1 [==============================] - ETA: 0s ▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯▯1/1 [==============================] - 0s 64ms/step
[1]
Prediction Done...
Your  have been Detected as  Normal
```