# CLASSIFICATION OF ARRHYTHMIA BY USING DEEP LEARNING WITH 2D ECG SPECTRAL IMAGE REPRESENTATION

| | |
|---|---|
| **DATE** | **18 NOVEMBER 2022** |
| **TEAM MEMBERS** | **Kaavya Lakshmanan**<br>**Agallya A N**<br>**Diloshaa sri R**<br>**Hiruthik K**<br>**Jawahar R** |
| **TEAM ID** | **PNT2022TMID18693** |

# 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**

# 1.Introduction

## 1.1 Project Overview

This project is design and implementation of detection of arrythmia detection using deep learning models Electrocardiogram (ECG) is a simple non-invasive measure to identify heart-related issues such as irregular heartbeats known as arrhythmias Deep CNN based algorithm is implemented for train the model , artificial intelligence and machine learning is being utilized in a wide range of healthcare related applications and datasets, many arrhythmia classifiers using deep learning methods have been proposed in recent years. However, sizes of the available datasets from which to build and assess machine learning models is often very small and the lack of well-annotated public ECG datasets is evident. In this paper, we propose a deep transfer learning framework that is aimed to perform classification on a small size training dataset. The proposed method is to fine-tune a general-purpose image classifier ResNet-18 with MIT-BIH arrhythmia dataset in accordance with the AAMI EC57 standard. This paper further investigates many existing deep learning models that have failed to avoid data leakage against AAMI recommendations

## 1.2 Purpose

The purpose of the project is design and implementation of deep learning model deployed for detection of heart disease and prediction

# 2. Literature Survey

## 2.1 Existing Problem

Cardiologists use mostly the raw ECG to diagnose. The simplest and fastest method of feature extraction is then to extract sampled points from an ECG signal curve. However, one should be aware of the fact that the amount of the extracted features used to characterize the heartbeat can be a burden for the classification algorithm. For this reason, most of the works that use the raw signal perform a down sampling of the waveform or some feature selection in order to reduce the computation time. In order to circumvent this issue, a simple machine learning method is chosen to classify the arrhythmias.

## 2.2 References

**1.** Kachuee, Mohammad, Shayan Fazeli, and Majid Sarrafzadeh. "Ecg heartbeat classification: A deep transferable representation." 2018 IEEE international conference on healthcare informatics (ICHI). IEEE, 2018

2.S. Zhang, W. Wang, J. Ford, and F. Make don, "Learning from incomplete ratings using nonnegative matrix factorization," in Proc. 6th SIAM Int. Conf. Data Mining, 2006, pp. 549–553.

3.T. Hofmann and J. Puzicha, "Latent class models for collaborative filtering," in Proc. 6th Int. Joint Conf. Artif. Intell., 1999, pp. 688–693.

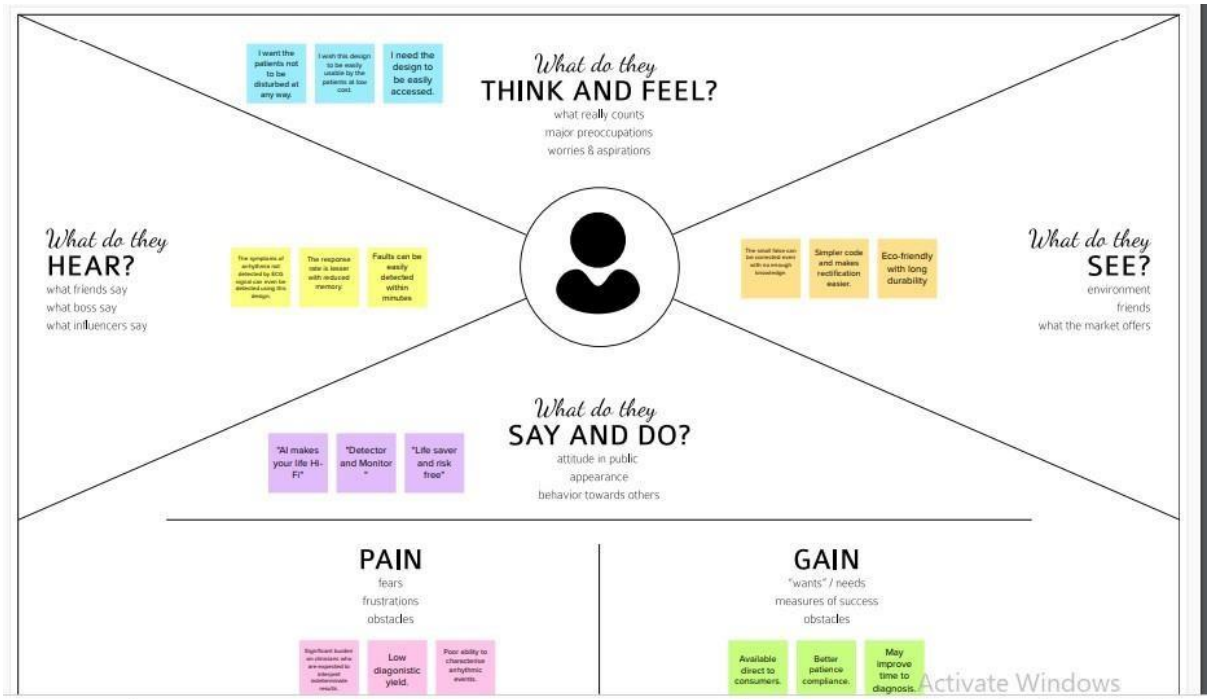
4.B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Reidl, "Item-based collaborative filtering recommendation algorithms," in Proc. 10th Int. World Wide Web Conf., 2001, pp. 285–295 5.T. George and S. Merugu, "A scalable collaborative filtering framework based on coclustering," in Proc. 5th IEEE Int. Conf. Data Mining, 2005, pp. 625–628

## 2.3 Problem Statement Definition

- ✓ Cardiologists by using various values which occurred during the ECG recording can decide whether the heart beat is normal or not. Since observation of these values are not always clear, existence of automatic ECG detection system is required
- ✓ Luz, Eduardo José da S., et al. "ECG-based heartbeat classification for arrhythmia detection: A survey." Computer methods and programs in biomedicine 127 (2016): 144-164
- ✓ Romdhane, Taissir Fekih, and Mohamed Atri Pr. "Electrocardiogram heartbeat classification based on a deep convolutional neural network and focal loss."
  Computers in Biology and Medicine 123 (2020): 103866

# 3.Ideation and Proposed Solution

## 3.1 Empathy Map Canvas




## 3.2 Ideation and Brain Storming


## 3.3 Proposed Solutions

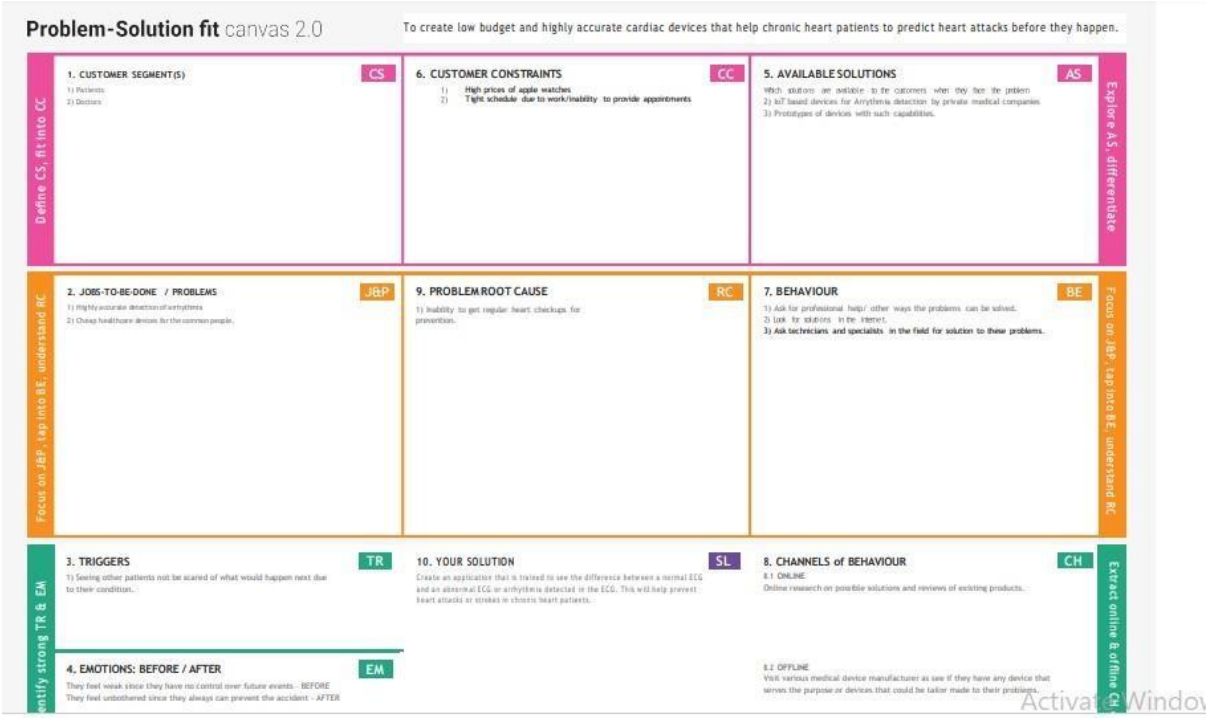
Deep learning based using for train the image The MIT-BIH database, an ECG database provided by the Massachusetts Institute of Technology and based on international standards and annotated information by multiple experts (Moody and Mark, 2001) is used in this study. The MIT-BIH database has been frequently used by the academic community in research for the detection and classification of

arrhythmic heartbeats. The MIT-BIH database contains 48 ECG recordings, each recording time is 30 min, the sampling frequency is 360 Hz, and each ECG record is composed of two leads. MIT-BIH database can make adjustments and corrections based on the information annotated by experts and optimization algorithms. Furthermore, it learns from existing solutions for self-optimization.

This paper proposes a novel deep learning approach to identify arrhythmias in ECG signals.

The proposed approach identifies arrhythmia classes using Convolutional Neural Network (CNN) trained by two-dimensional (2D) ECG beat images. Firstly, ECG signals, which consist of 5 different arrhythmias, are segmented into heartbeats which are transformed into 2D grayscale images. Afterward, the images are used as input for training a new CNN architecture to classify heartbeats.

## 3.4 Problem Solution Fit



# 4.Requirement Analysis

## 4.1 Functional Requirement

| FR No. | Functional Requirement (Epic) | Sub Requirement (Story / Sub-Task) |
|---|---|---|
| FR-1 | User Registration | Registration through Form , Registration through Gmail |
| FR-2 | User Confirmation | Confirmation via Email , Confirmation via OTP |
| FR-3 | Get User Input | Upload image as jpeg , Upload image as png |
| FR-4 | Save Image | Images are saved in the uploads folder |
| FR-5 | Chat with Doctor | Consult with Doctor |
| FR-6 | Report Generation | Get complete Report |

## 4.2 Non -Functional Requirement

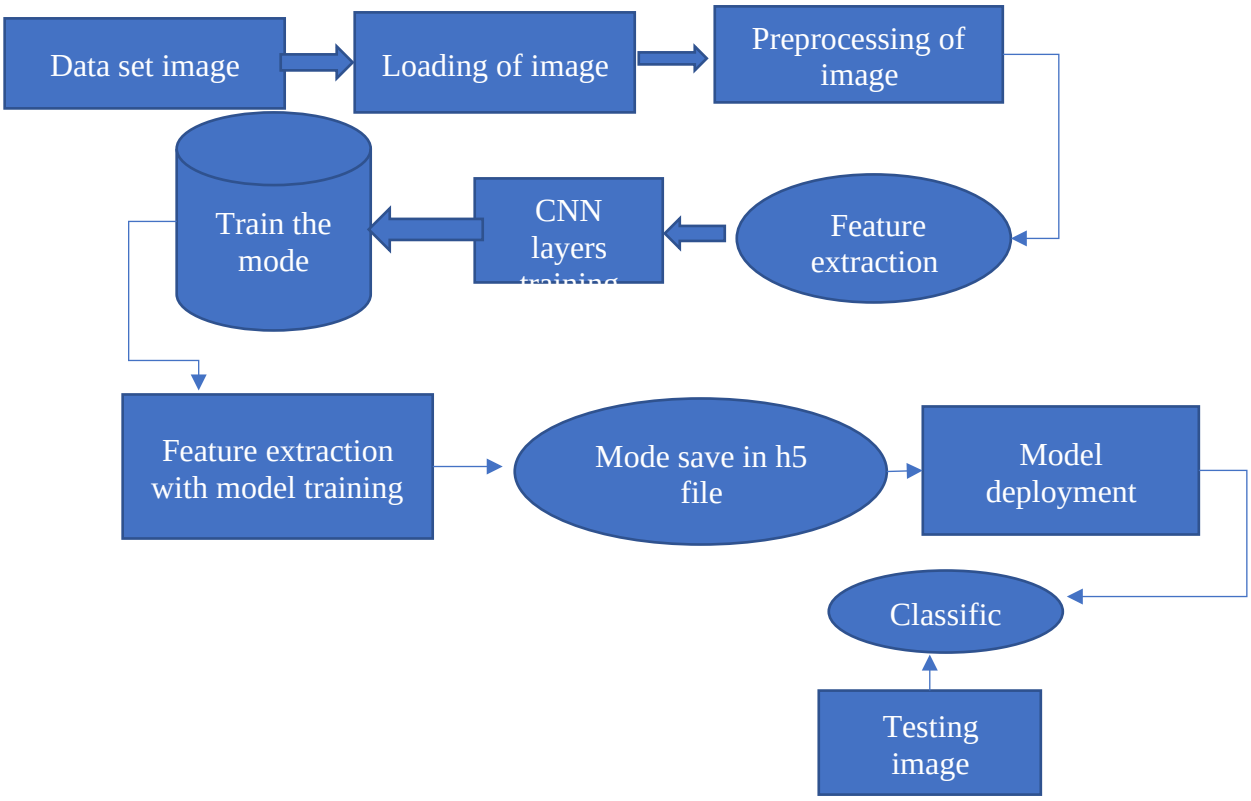
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where data is stored.

Nonfunctional Requirements (NFRs) define system attributes such as :

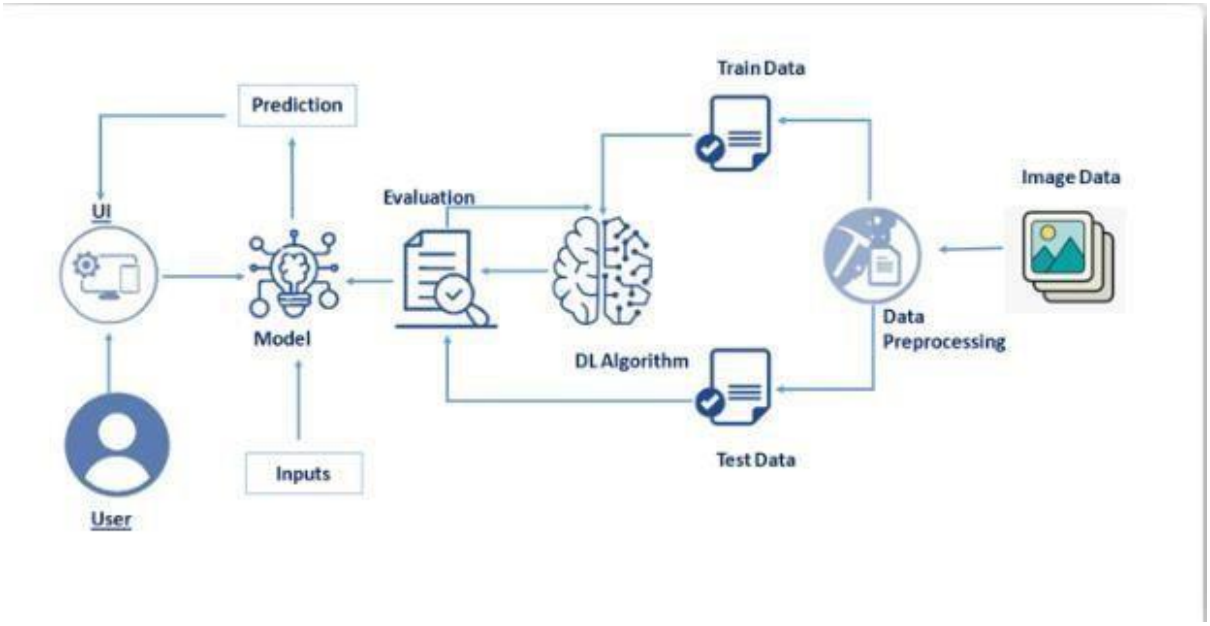
- ✓ Security,
- ✓ reliability,
- ✓ performance,
- ✓ maintainability,
- ✓ scalability and
- ✓ usability.

# 5.Project Design

## 5.1 Data Flow Diagrams




## 5.2 Solutions and Technical Architecture



## 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 |

# 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 | Dataset Collection | USN-1 | Collect the dataset from the sources available in IBM | 10 | High | Kaavya Lakshmanan<br>Agallya A N<br>Diloshaa Sri R<br>Hiruthik K<br>Jawahar R |
| Sprint-2 | Image Preprocessing | USN-2 | Remove noise present in the images collected and perform data pre-processing | 10 | High | Kaavya Lakshmanan<br>Agallya A N<br>Diloshaa Sri R<br>Hiruthik K<br>Jawahar R |
| Sprint-2 | Build the CNN Model | USN-3 | Identify the appropriate layers required for the model and determine the model parameters | 2 | High | Kaavya Lakshmanan<br>Agallya A N<br>Diloshaa Sri R<br>Hiruthik K |
| Sprint-2 | Configure the model | USN-4 | Perform model configuration by compiling it and implement techniques for loss reduction | 5 | Medium | Kaavya Lakshmanan<br>Agallya A N<br>Diloshaa Sri R<br>Hiruthik K<br>Jawahar R |
| Sprint-2 | Train, test and validate | USN-5 | Initiate model training phase, later based on model and validation loss values, start test phase | 13 | High | Kaavya Lakshmanan<br>Agallya A N<br>Diloshaa Sri R<br>Hiruthik K<br>Jawahar R |
| Sprint-3 | Register for IBM Cloud | USN-6 | Set up IBM Watson Assistant with Cloud Service | 2 | High | Kaavya Lakshmanan<br>Agallya A N<br>Diloshaa Sri R<br>Hiruthik K<br>Jawahar R |
| Sprint-4 | Develop the web interface using Flask | USN-7 | Design a UI for the web interface, with login, registration and input adding features | 5 | High | Kaavya Lakshmanan<br>Agallya A N<br>Diloshaa Sri R<br>Hiruthik K<br>Jawahar R |
| Sprint-4 | Perform server-side scripting | USN-8 | Develop an application using python for back-end functions | 13 | Medium | Kaavya Lakshmanan<br>Agallya A N<br>Diloshaa Sri R<br>Hiruthik K<br>Jawahar R |

## 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 | 6 Days | 24 Oct 2022 | 28 Oct 2022 | 20 | 28 Oct 2022 |
| Sprint-2 | 20 | 6 Days | 30 Oct 2022 | 05 Nov 2022 | 20 | 05 Nov 2022 |
| Sprint-3 | 20 | 6 Days | 06 Nov 2022 | 12 Nov 2022 | 20 | 12 Nov 2022 |
| Sprint-4 | 20 | 6 Days | 13 Nov 2022 | 19 Nov 2022 | 20 | 19 Nov 2022 |

# 7.Coding & Solutioning

# Software Designing

**Anaconda Navigator :**

Anaconda is a free and open-source distribution of the Python and R programming languages for scientific computing that aims to simplify package management and deployment. Package versions are managed by the package management system conda. The Anaconda distribution includes data-science packages suitable for Windows, Linux, and macOS. Anaconda distribution comes with 1,500 packages selected from PyPI as well as the conda package and virtual environment manager. It also includes a GUI, Anaconda Navigator, as a graphical alternative to the command-line interface (CLI).

**Jupyter Notebook :**

Anaconda distribution comes with 1,500 packages selected from PyPI as well as the conda package and virtual environment manager. It also includes a GUI, Anaconda Navigator, as a graphical alternative to the command line interface (CLI). A Jupyter Notebook document is a JSON document, following a versioned schema, and containing an ordered list of input/output cells which can contain code, text mathematics, plots and rich media, usually ending with the ". jpynb" extension.

**Tensor flow :**

TensorFlow is an end-to-end open-source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries, and community resources that lets researchers push the state-of-the-art in ML and developers can easily build and deploy ML powered applications.

**Keras :**

Keras is an open-source neural-network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognitive Toolkit, R, Theano, or Plaid ML. Designed to enable fast experimentation with deep neural networks, it focuses on being user-friendly, modular, and extensible. Keras contains numerous implementations of commonly used neural- network building blocks such as layers, objectives, activation functions, optimizers, and a host of tools to make working with image and text data easier to simplify the coding necessary for writing deep neural network code.

**Flask :**

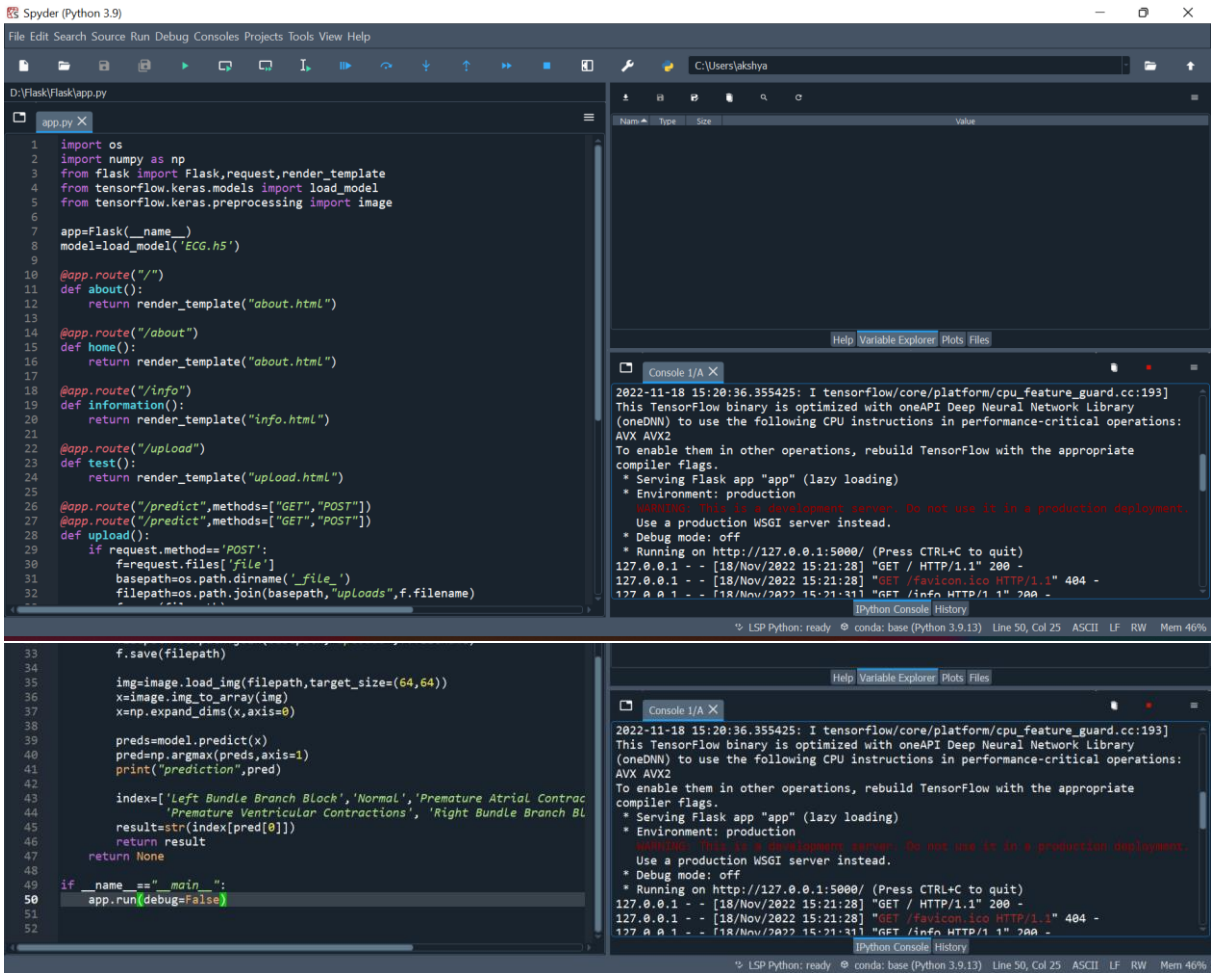
Flask is a microframework written in python. It is classified as a microframework because it does not require particular tools or libraries. It has no database abstraction layer, form validation, or any other components where

preexisting third-party libraries provide common functions. However, Flask supports extensions that can add application features as if they were implemented in Flask itself. Extensions exist for object- relational mappers, form validation, upload handling, various open authentication technologies and several common framework related tools.

**Source Code Flask**

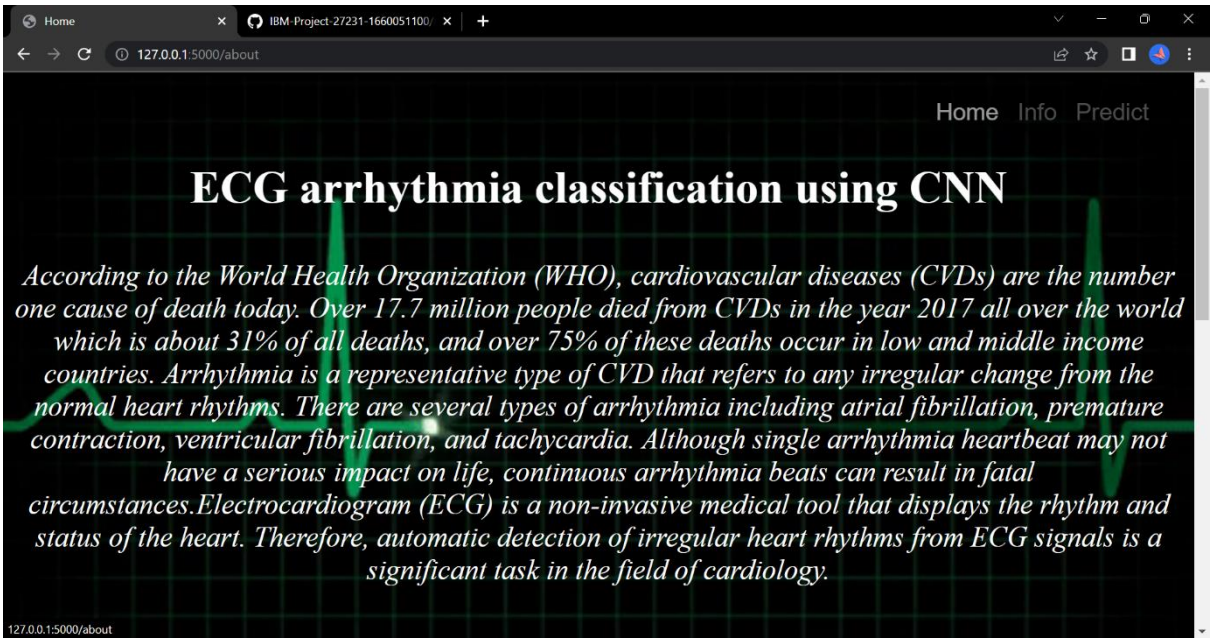
```
import os
import numpy as np
from flask import Flask,request,render_template
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image

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

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

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

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

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

@app.route("/predict",methods=["GET","POST"])
@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=image.load_img(filepath,target_size=(64,64))
        x=image.img_to_array(img)
        x=np.expand_dims(x,axis=0)

        preds=model.predict(x)
```

```
        pred=np.argmax(preds,axis=1)
        print("prediction",pred)

        index=['Left Bundle Branch Block','Normal','Premature Atrial Contraction',
               'Premature Ventricular Contractions', 'Right Bundle Branch Block','Ventricular
Fibrillation']
        result=str(index[pred[0]])
        return result
    return None

if __name__=="__main__":
    app.run(debug=False)
```

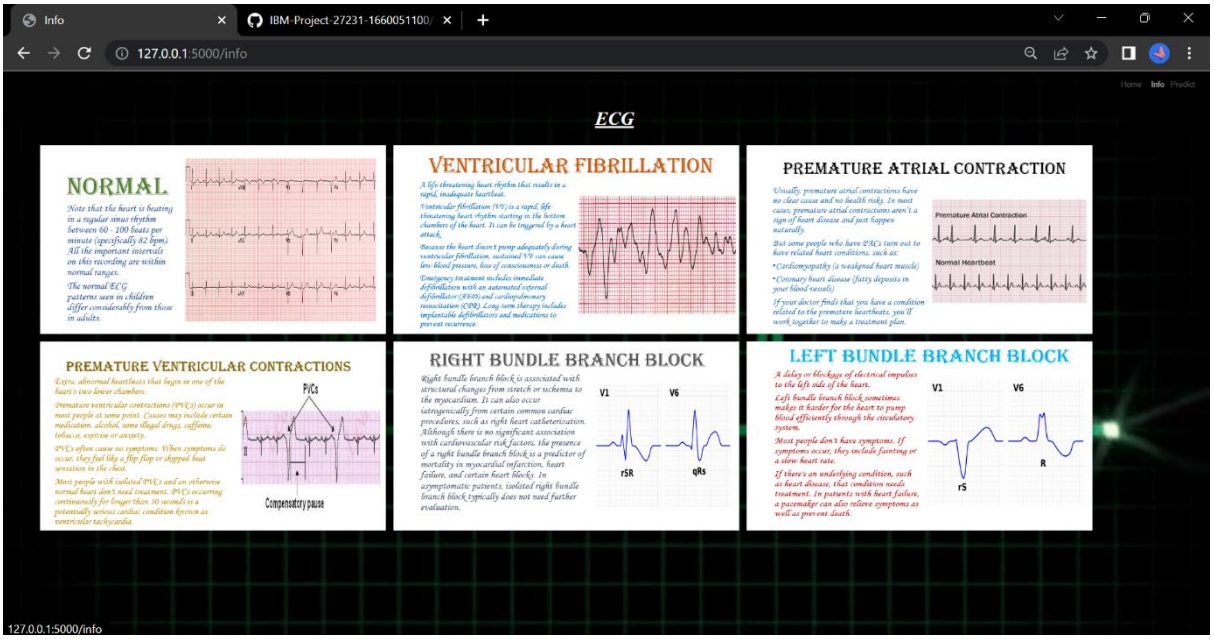


**OUTPUT:**

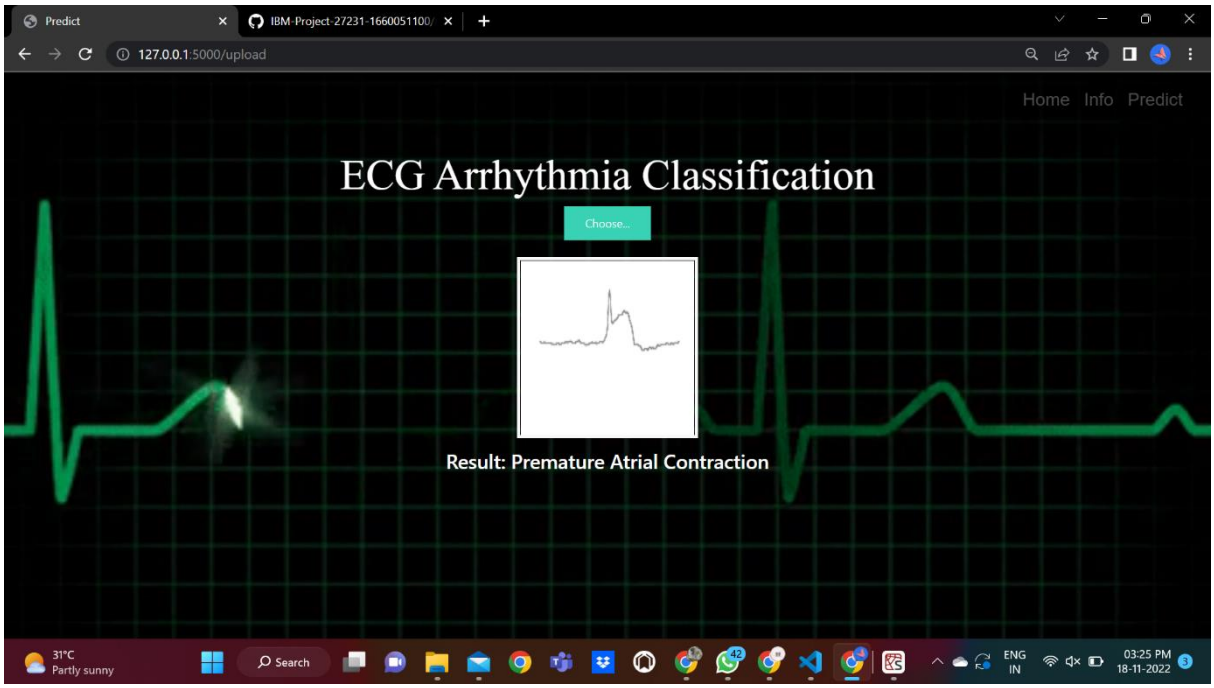
**HOME PAGE:**




**INFO PAGE :**



**PREDICTION PAGE:**




# 8.TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

**Test the model :**

The model is to be tested with different images to know if it is predicting correctly. .
In split the data we set the image as 80% Training Data and 20% Testing Data. Then build CNN model train deep neural network for epochs

## 8.1TYPES OF TESTS

**Unit testing :**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration.

This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**Integration testing :**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

**Functional test :**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.
Invalid Input : identified classes of invalid input must be rejected.
Functions : identified functions must be exercised.
Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**System Test :**

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

**White Box Testing :**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**Black Box Testing :**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

**8.2 User Acceptance Testing:**

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases**.**

**Test strategy and approach**

Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- ✓ All field entries must work properly.
- ✓ Pages must be activated from the identified link.
- ✓ The entry screen, messages and responses must not be delayed.

Features to be tested

- ✓ Verify that the entries are of the correct format

- ✓ No duplicate entries should be allowed
- ✓ All links should take the user to the correct page.

**Integration Testing**

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications,
e.g. components in a software system or – one step up – software applications at the company level – interact without error.

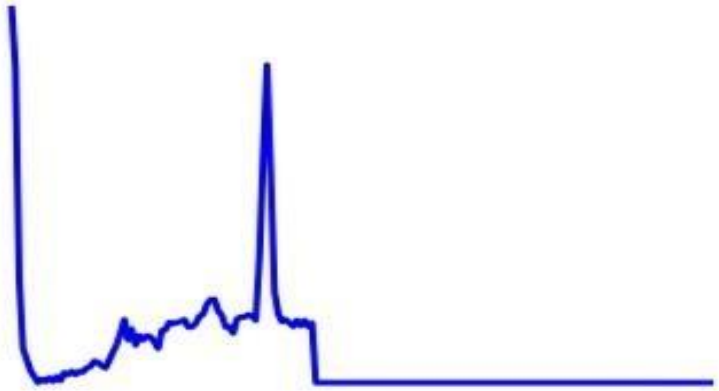
**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**Acceptance Testing**

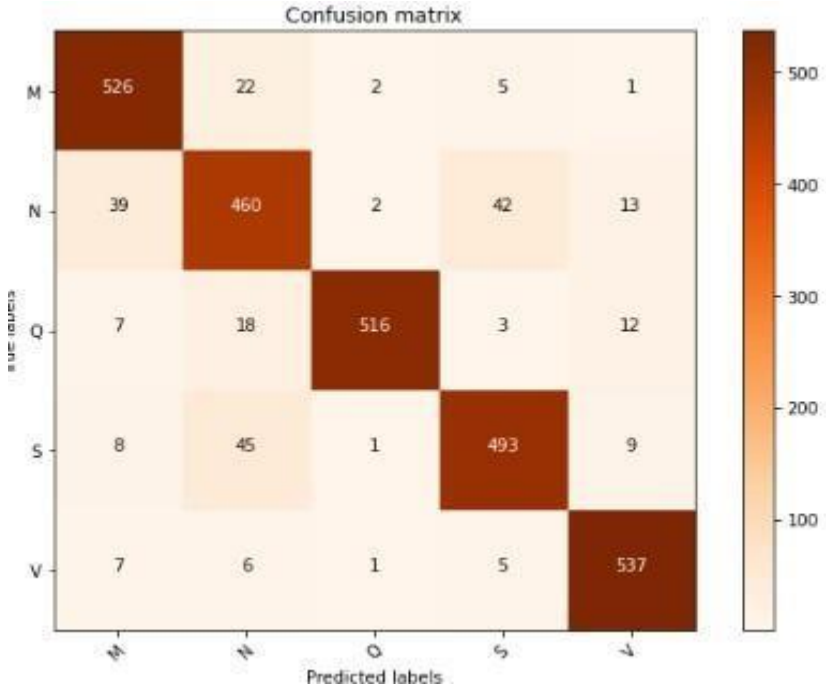
User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.
**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

# 9.RESULTS

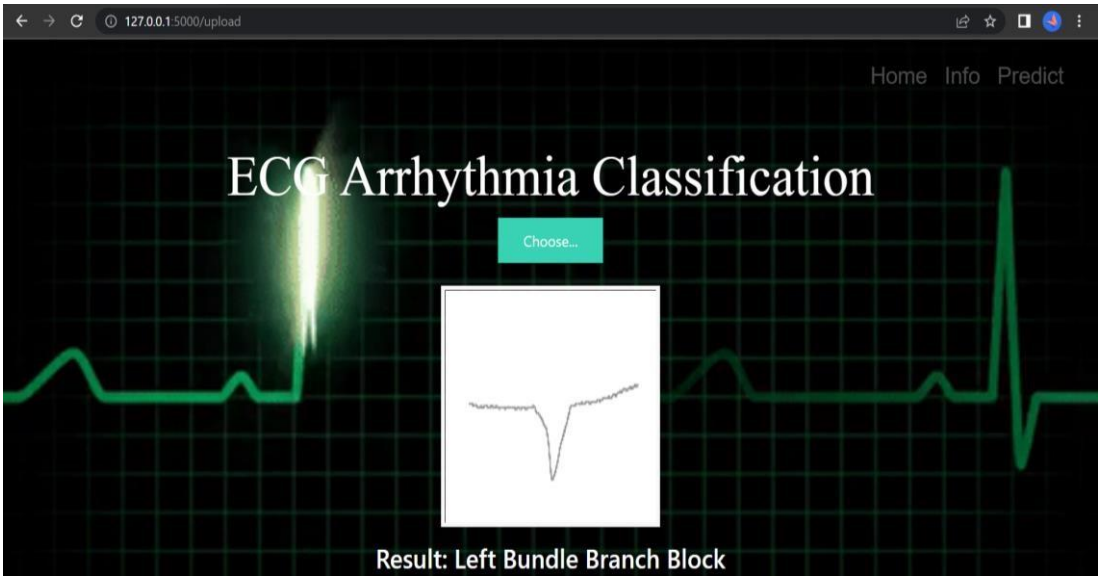
## Testing image

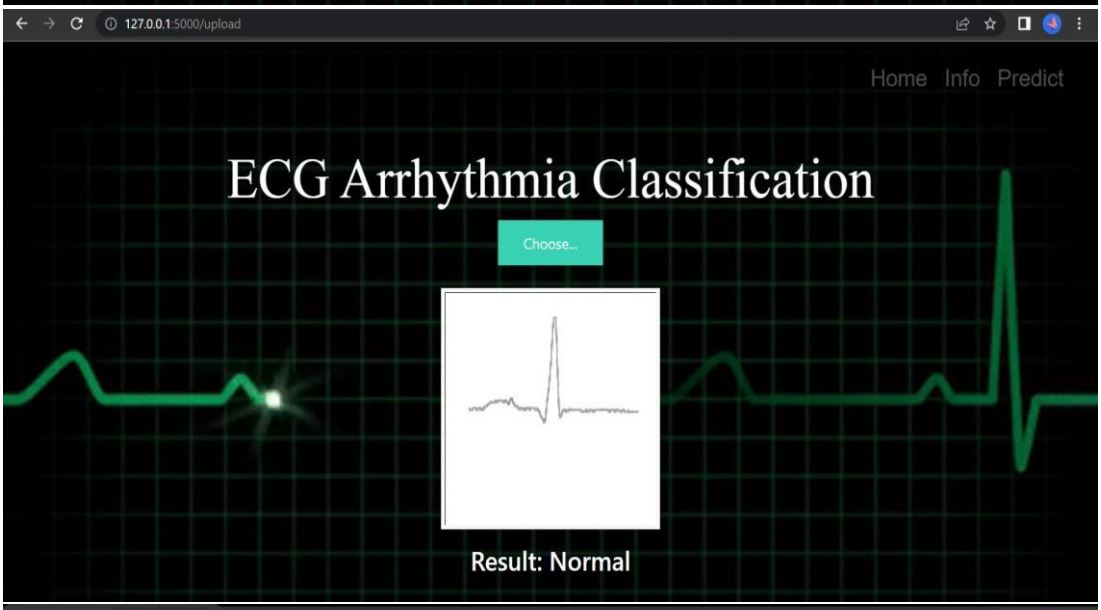


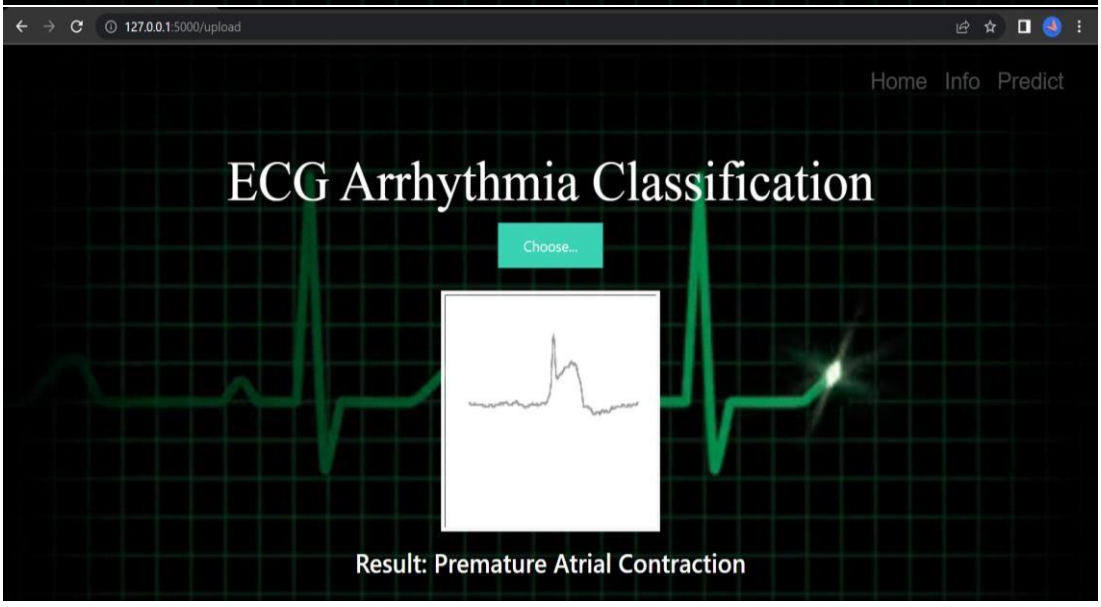
## Confusion matrix

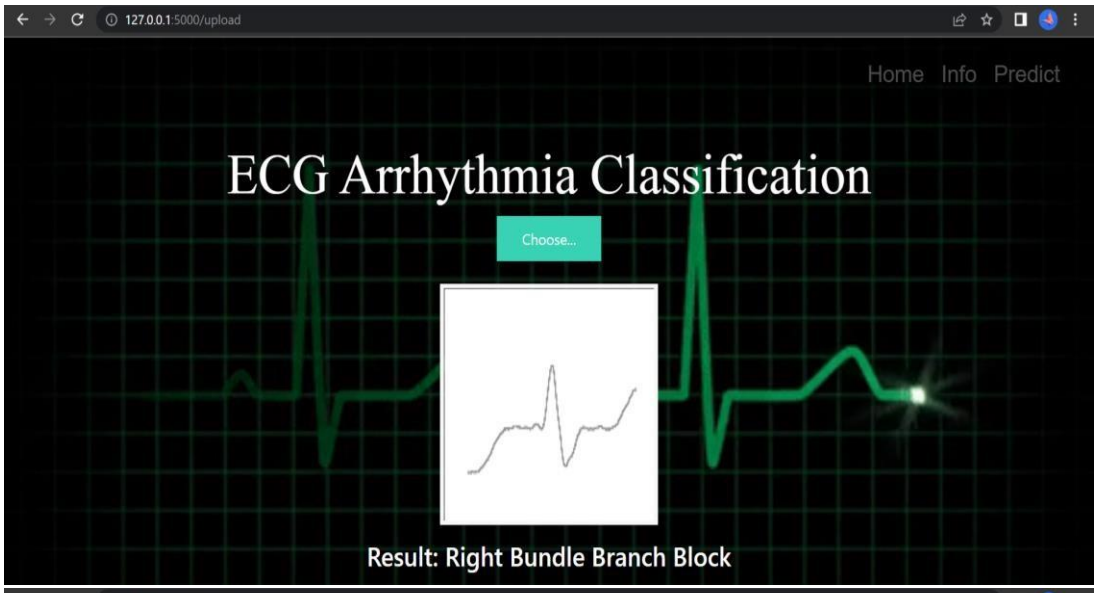


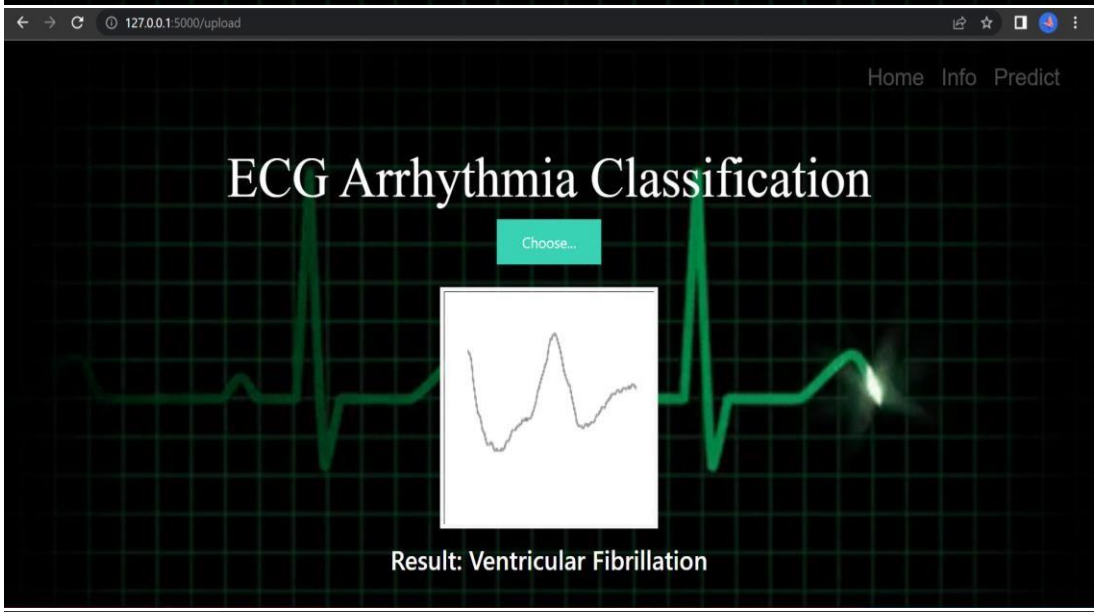
## 9.1 Performance Metrics

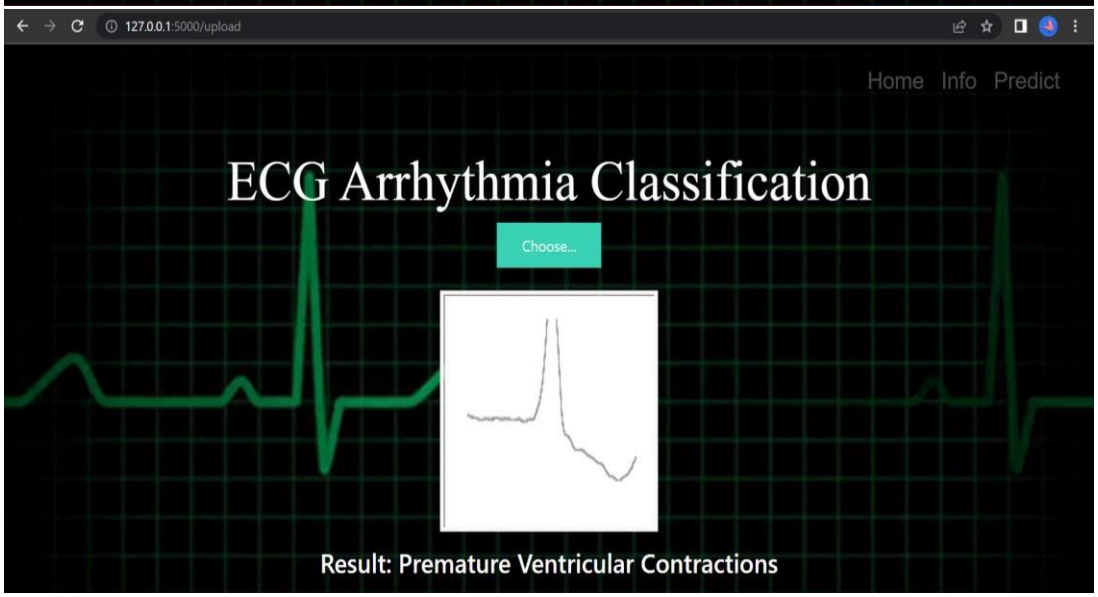


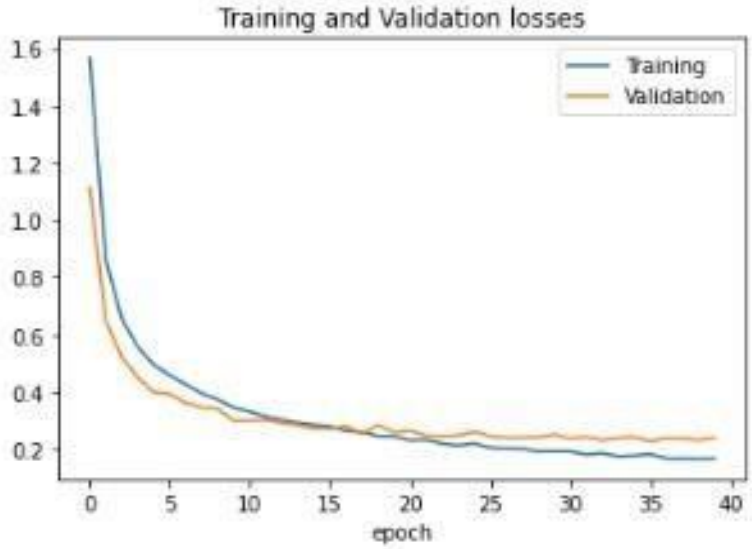




127.0.0.1:5000/upload
Home Info Predict
ECG Arrhythmia Classification
Choose...
Result: Right Bundle Branch Block

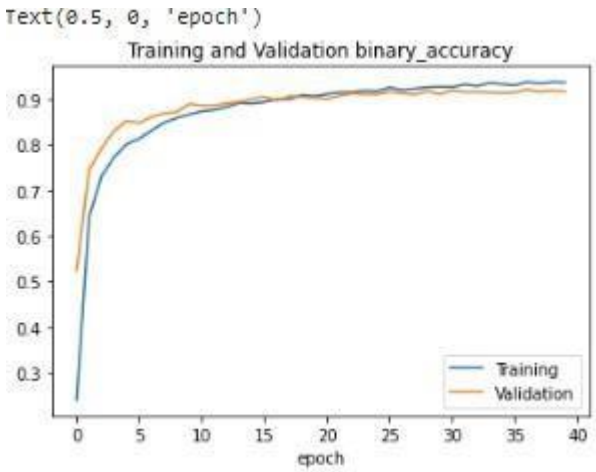

127.0.0.1:5000/upload
Home Info Predict
ECG Arrhythmia Classification
Choose...
Result: Ventricular Fibrillation


127.0.0.1:5000/upload
Home Info Predict
ECG Arrhythmia Classification
Choose...
Result: Premature Ventricular Contractions

## Training and validation losses




## Training and testing accuracy



## 10.Advantage and Disadvantages

1. High accuracy
2. High sensitivity
3. High reliability
4. Reduced loss

The advantage of this project is that the initiative for the early detection of diseases is a famous study and classification. The issues of biometric authentication and the application of emotional recognition can be resolved by various techniques, unlike heartbeat type detection.

The imbalance of the ECG dataset is a significant issue because certain classes have a lot of data relative to others, which might lead to false information about model performance.This is consider to be the disadvantage of the project .

## 11.Conclusion

This project is designed the In using the MIT-BIH arrhythmia database, we have proposed a system for the automatic processing of the ECG for the classification of arrhythmia images. The database of MIT-BIH is processed visually and a waveform detection method is proposed for detecting the QRS waveform. A CNN model was built to train and classify the ECG images. Experimental results show that according to the ANSI/AAMI EC57 evaluation criteria, The accuracy rate of ventricular ectopic beat can reach 95.9% and the sensitivity evaluation is 93.0%. For the supraventricular ectopic beat class, the accuracy rate is 93.2% and the sensitivity evaluation is 81.3%

The proposed model has the potential to be introduced into clinical settings as a helpful tool to aid the cardiologists in the reading of ECG heartbeat signals and to understand more about them.

# 12.Future work

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

### Importing Neccessary Libraries

In [1]:
```
import numpy as np#used for numerical analysis
import tensorflow #open source used for both ML and DL for computation
from tensorflow.keras.models import Sequential #it is a plain stack of layers
from tensorflow.keras import layers #A layer consists of a tensor-in tensor-out computation function
#Dense layer is the regular deeply connected neural network layer
from tensorflow.keras.layers import Dense,Flatten
#Faltten-used fot flattening the input or change the dimension
from tensorflow.keras.layers import Conv2D,MaxPooling2D #Convolutional layer
#MaxPooling2D-for downsampling the image
from keras.preprocessing.image import ImageDataGenerator
```

### Image Data Agumentation

In [2]:
```
#setting parameter for Image Data agumentation to the traing data
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
```

In [3]:
```
#Image Data agumentation to the testing data
test_datagen=ImageDataGenerator(rescale=1./255)
```

### Loading our data and performing data agumentation

In [4]:
```
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='NN3uEviLwb1ABILT6QsqOT1rXx4vlrrrw5JS9yxoPlyY',
    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')

streaming_body_1 = cos_client.get_object(Bucket='arrhythmiaecg-donotdelete-pr-auuumsehlgbz0a', Key='Dataset.zip')['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/
if not hasattr (streaming_body_1, "__iter__"): streaming_body_1.__iter__ = types.MethodType( __iter__, streaming_body_1)
```

In [5]:
```
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 [6]:
```
pwd
```

Out[6]:
```
'/home/wsuser/work'
```

In [7]:
```
import os
filenames=os.listdir('/home/wsuser/work/Dataset/train')
```

In [8]:
```
#performing data agumentation to train data
x_train=train_datagen.flow_from_directory(directory='/home/wsuser/work/Dataset/train'
                                          ,target_size=(64,64),batch_size=32,class_mode='categorical')
#performing data agumentation to test data
x_test=test_datagen.flow_from_directory(directory='/home/wsuser/work/Dataset/test'
                                        ,target_size=(64,64),batch_size=32,class_mode='categorical')
```

```
Found 15341 images belonging to 6 classes.
Found 6825 images belonging to 6 classes.
```

In [9]:
```
print(x_train.class_indices)#checking the number of classes
```

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

In [10]:
```
from collections import Counter as c
c(x_train.labels)
```

Out[10]:
```
Counter({0: 504, 1: 7346, 2: 2054, 3: 2759, 4: 2239, 5: 439})
```

### Creating the model

In [11]:
```
# create model
model=Sequential()
# adding model layer
model.add(Conv2D(32,(3,3),input_shape=(64,64,3),activation='relu'))#convolutional layer
model.add(MaxPooling2D(pool_size=(2,2))) #MaxPooling2D-for downsampling the input

model.add(Conv2D(32,(3,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Flatten())#flatten the dimension of the image
model.add(Dense(32))#deeply connected neural network layers.
model.add(Dense(6,activation='softmax'))#output layer with 6 neurons
```

In [12]:
```
model.summary()#summary of our model
```

```
Model: "sequential"
_________________________________________________________________
 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, 6)                 198

=================================================================
Total params: 211,078
Trainable params: 211,078
Non-trainable params: 0
_________________________________________________________________
```

### Compiling the model

In [13]:
```
# Compile model
model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])
```

### Fitting the model

In [14]:
```
# Fit the model
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_164/1433457599.py:2: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, which s
upports generators.
  model.fit_generator(generator=x_train,steps_per_epoch = len(x_train),
Epoch 1/10
480/480 [==============================] - 88s 181ms/step - loss: 0.5529 - accuracy: 0.8160 - val_loss: 0.4815 - val_accuracy: 0.8459
Epoch 2/10
480/480 [==============================] - 92s 191ms/step - loss: 0.2409 - accuracy: 0.9278 - val_loss: 0.4750 - val_accuracy: 0.8646
Epoch 3/10
480/480 [==============================] - 95s 198ms/step - loss: 0.1985 - accuracy: 0.9413 - val_loss: 0.3185 - val_accuracy: 0.9095
Epoch 4/10
480/480 [==============================] - 89s 185ms/step - loss: 0.1758 - accuracy: 0.9486 - val_loss: 0.2898 - val_accuracy: 0.9146
Epoch 5/10
480/480 [==============================] - 91s 189ms/step - loss: 0.1514 - accuracy: 0.9544 - val_loss: 0.3326 - val_accuracy: 0.8980
Epoch 6/10
480/480 [==============================] - 91s 190ms/step - loss: 0.1369 - accuracy: 0.9567 - val_loss: 0.2876 - val_accuracy: 0.9184
Epoch 7/10
480/480 [==============================] - 92s 192ms/step - loss: 0.1220 - accuracy: 0.9623 - val_loss: 0.3215 - val_accuracy: 0.9320
Epoch 8/10
480/480 [==============================] - 92s 192ms/step - loss: 0.1161 - accuracy: 0.9651 - val_loss: 0.3569 - val_accuracy: 0.9160
Epoch 9/10
480/480 [==============================] - 92s 192ms/step - loss: 0.1049 - accuracy: 0.9673 - val_loss: 0.3626 - val_accuracy: 0.9338
Epoch 10/10
480/480 [==============================] - 90s 188ms/step - loss: 0.0993 - accuracy: 0.9692 - val_loss: 0.4166 - val_accuracy: 0.8995
```

Out[14]:
```
<keras.callbacks.History at 0x7f36bc2b24c0>
```

In [15]:
```
#model.fit_generator(x_train,epochs=10,validation_data=x_test)
```

## Saving our model

In [16]:
```
# Save the model
model.save('ECG.h5')
```

In [17]:
```
!tar -zcvf image-classification-model_new.tgz ECG.h5
```

```
ECG.h5
```

In [18]:
```
ls -1
```

```
Dataset/
ECG.h5
image-classification-model_new.tgz
```

In [19]:
```
!pip install watson-machine-learning-client --upgrade
```

```
Collecting watson-machine-learning-client
  Downloading watson_machine_learning_client-1.0.391-py3-none-any.whl (538 kB)
     |████████████████████████████████| 538 kB 14.9 MB/s eta 0:00:01
Requirement already satisfied: lomond in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (0.3.3)
Requirement already satisfied: urllib3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.26.7)
Requirement already satisfied: pandas in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.3.4)
Requirement already satisfied: requests in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2.26.0)
Requirement already satisfied: certifi in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2022.9.24)
Requirement already satisfied: tabulate in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (0.8.9)
Requirement already satisfied: tqdm in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (4.62.3)
Requirement already satisfied: ibm-cos-sdk in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (2.11.0)
Requirement already satisfied: boto3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from watson-machine-learning-client) (1.18.21)
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning-client) (0.10.
0)
```

```
Requirement already satisfied: botocore<1.22.0,>=1.21.21 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning-client) (1.21.41)
Requirement already satisfied: s3transfer<0.6.0,>=0.5.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from boto3->watson-machine-learning-client) (0.5.0)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from botocore<1.22.0,>=1.21.21->boto3->watson-machine-learning-client) (2.8.2)
Requirement already satisfied: six>=1.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from python-dateutil<3.0.0,>=2.1->botocore<1.22.0,>=1.21.21->boto3->watson-machine-learning-client) (1.15.0)
Requirement already satisfied: ibm-cos-sdk-s3transfer==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk->watson-machine-learning-client) (2.11.0)
Requirement already satisfied: ibm-cos-sdk-core==2.11.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from ibm-cos-sdk->watson-machine-learning-client) (2.11.0)
Requirement already satisfied: idna<4,>=2.5 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->watson-machine-learning-client) (3.3)
Requirement already satisfied: charset-normalizer~=2.0.0 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from requests->watson-machine-learning-client) (2.0.4)
Requirement already satisfied: pytz>=2017.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas->watson-machine-learning-client) (2021.3)
Requirement already satisfied: numpy>=1.17.3 in /opt/conda/envs/Python-3.9/lib/python3.9/site-packages (from pandas->watson-machine-learning-client) (1.20.3)
Installing collected packages: watson-machine-learning-client
Successfully installed watson-machine-learning-client-1.0.391
```

In [21]:
```
from ibm_watson_machine_learning import APIClient
wml_credentials = {
                   "url" :"https://us-south.ml.cloud.ibm.com",
                   "apikey" :"BPSj6HFczuOBxpbl0EmnzzZQDVtTmZj_IXWMrA6h7woF"
                  }
client = APIClient(wml_credentials)
```

In [22]:
```
client = APIClient(wml_credentials)
```

In [23]:
```
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 [24]:
```
space_uid = guid_from_space_name(client, 'models')
print("Space UID = " + space_uid)
```

```
Space UID = 6495d2c8-d0f5-4a46-a7f2-5f3ab86642aa
```

In [25]:
```
client.set.default_space(space_uid)
```

Out[25]:
```
'SUCCESS'
```

In [26]:
```
client.software_specifications.list()
```

```
------------------------------  ------------------------------------  ----
NAME                            ASSET_ID                              TYPE
default_py3.6                   0062b8c9-8b7d-44a0-a9b9-46c416adcbd9  base
kernel-spark3.2-scala2.12       020d69ce-7ac1-5e68-ac1a-31189867356a  base
pytorch-onnx_1.3-py3.7-edt      069ea134-3346-5748-b513-49120e15d288  base
scikit-learn_0.20-py3.6         09c5a1d0-9c1e-4473-a344-eb7b665ff687  base
spark-mllib_3.0-scala_2.12      09f4cff0-90a7-5899-b9ed-1ef348aebdee  base
pytorch-onnx_rt22.1-py3.9       0b848dd4-e681-5599-be41-b5f6fccc6471  base
ai-function_0.1-py3.6           0cdb0f1e-5376-4f4d-92dd-da3b69aa9bda  base
shiny-r3.6                      0e6e79df-875e-4f24-8ae9-62dcc2148306  base
tensorflow_2.4-py3.7-horovod    1092590a-307d-563d-9b62-4eb7d64b3f22  base
pytorch_1.1-py3.6               10ac12d6-6b30-4ccd-8392-3e922c096a92  base
tensorflow_1.15-py3.6-ddl       111e41b3-de2d-5422-a4d6-bf776828c4b7  base
autoai-kb_rt22.2-py3.10         125b6d9a-5b1f-5e8d-972a-b251688ccf40  base
runtime-22.1-py3.9              12b83a17-24d8-5082-900f-0ab31fbfd3cb  base
scikit-learn_0.22-py3.6         154010fa-5b3b-4ac1-82af-4d5ee5abbc85  base
default_r3.6                    1b70aec3-ab34-4b87-8aa0-a4a3c8296a36  base
pytorch-onnx_1.3-py3.6          1bc6029a-cc97-56da-b8e0-39c3880dbbe7  base
kernel-spark3.3-r3.6            1c9e5454-f216-59dd-a20e-474a5cdf5988  base
pytorch-onnx_rt22.1-py3.9-edt   1d362186-7ad5-5b59-8b6c-9d0880bde37f  base
tensorflow_2.1-py3.6            1eb25b84-d6ed-5dde-b6a5-3fbdf1665666  base
spark-mllib_3.2                 20047f72-0a98-58c7-9ff5-a77b012eb8f5  base
tensorflow_2.4-py3.8-horovod    217c16f6-178f-56bf-824a-b19f20564c49  base
runtime-22.1-py3.9-cuda         26215f05-08c3-5a41-a1b0-da66306ce658  base
do_py3.8                        295addb5-9ef9-547e-9bf4-92ae3563e720  base
autoai-ts_3.8-py3.8             2aa0c932-798f-5ae9-abd6-15e0c2402fb5  base
tensorflow_1.15-py3.6           2b73a275-7cbf-420b-a912-eae7f436e0bc  base
kernel-spark3.3-py3.9           2b7961e2-e3b1-5a8c-a491-482c8368839a  base
pytorch_1.2-py3.6               2c8ef57d-2687-4b7d-acce-01f94976dac1  base
spark-mllib_2.3                 2e51f700-bca0-4b0d-88dc-5c6791338875  base
pytorch-onnx_1.1-py3.6-edt      32983cea-3f32-4400-8965-dde874a8d67e  base
spark-mllib_3.0-py37            36507ebe-8770-55ba-ab2a-eafe787600e9  base
spark-mllib_2.4                 390d21f8-e58b-4fac-9c55-d7ceda621326  base
autoai-ts_rt22.2-py3.10         396b2e83-0953-5b86-9a55-7ce1628a406f  base
xgboost_0.82-py3.6              39e31acd-5f30-41dc-ae44-60233c80306e  base
pytorch-onnx_1.2-py3.6-edt      40589d0e-7019-4e28-8daa-fb03b6f4fe12  base
pytorch-onnx_rt22.2-py3.10      40e73f55-783a-5535-b3fa-0c8b94291431  base
default_r36py38                 41c247d3-45f8-5a71-b065-8580229facf0  base
autoai-ts_rt22.1-py3.9          4269d26e-07ba-5d40-8f66-2d495b0c71f7  base
autoai-obm_3.0                  42b92e18-d9ab-567f-988a-4240ba1ed5f7  base
pmml-3.0_4.3                    493bcb95-16f1-5bc5-bee8-81b8af80e9c7  base
spark-mllib_2.4-r_3.6           49403dff-92e9-4c87-a3d7-a42d0021c095  base
xgboost_0.90-py3.6              4ff8d6c2-1343-4c18-85e1-689c965304d3  base
pytorch-onnx_1.1-py3.6          50f95b2a-bc16-43bb-bc94-b0bed208c60b  base
autoai-ts_3.9-py3.8             52c57136-80fa-572e-8728-a5e7cbb42cde  base
spark-mllib_2.4-scala_2.11      55a70f99-7320-4be5-9fb9-9edb5a443af5  base
spark-mllib_3.0                 5c1b0ca2-4977-5c2e-9439-ffd44ea8ffe9  base
autoai-obm_2.0                  5c2e37fa-80b8-5e77-840f-d912469614ee  base
spss-modeler_18.1               5c3cad7e-507f-4b2a-a9a3-ab53a21dee8b  base
cuda-py3.8                      5d3232bf-c86b-5df4-a2cd-7bb870a1cd4e  base
autoai-kb_3.1-py3.7             632d4b22-10aa-5180-88f0-f52dfb6444d7  base
pytorch-onnx_1.7-py3.8          634d3cdc-b562-5bf9-a2d4-ea90a478456b  base
------------------------------  ------------------------------------  ----
Note: Only first 50 records were displayed. To display more use 'limit' parameter.
```

In [29]:
```
software_spec_uid = client.software_specifications.get_uid_by_name("tensorflow_rt22.1-py3.9")
software_spec_uid
```

Out[29]:
```
'acd9c798-6974-5d2f-a657-ce06e986df4d'
```

In [31]:
```
model_details = client.repository.store_model(model='image-classification-model_new.tgz',meta_props={
client.repository.ModelMetaNames.NAME:"CNN",
client.repository.ModelMetaNames.TYPE:"tensorflow_rt22.1",
```

```
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 [32]:
```
model_id
```

Out[32]:
```
'cb3b8767-c2a5-4782-b8b3-46857ffe23ff'
```

In [36]:
```
client.repository.download(model_id,'my_model.tar.gz')
```

```
Successfully saved model content to file: 'my_model.tar.gz'
```

Out[36]:
```
'/home/wsuser/work/my_model.tar.gz'
```

### Predicting our results

In [38]:
```
from tensorflow.keras.utils import load_img,img_to_array
from tensorflow.keras.models import load_model
from keras.preprocessing import image
model=load_model('/home/wsuser/work/ECG.h5')
 #loading the model for testing
```

In [40]:
```
img = load_img("Dataset/test/Left Bundle Branch Block/fig_5934.png",target_size= (64,64))#loading of the image
x = img_to_array(img)#image to array
x = np.expand_dims(x,axis = 0)#changing the shape
#pred = model.predict_classes(x)#predicting the classes
#pred
preds=model.predict(x)
pred=np.argmax(preds,axis=1)
preds
```

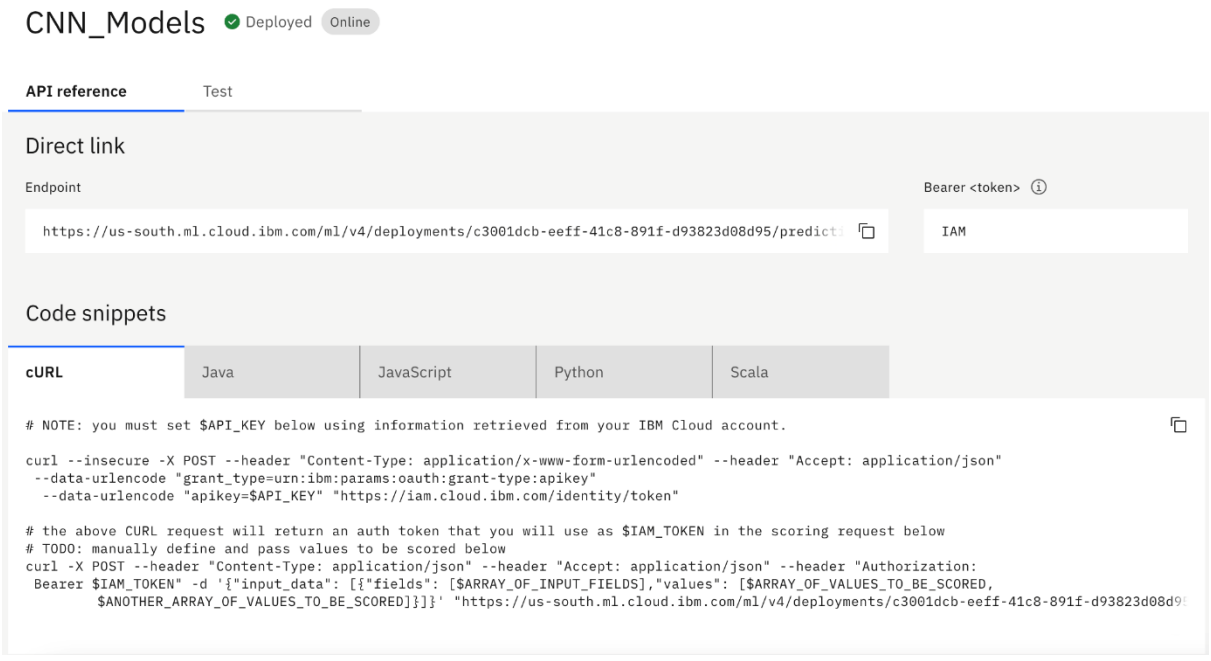
Out[40]:
```
array([[1., 0., 0., 0., 0., 0.]], dtype=float32)
```

In [41]:
```
index=['Left Bundle Branch Block','Normal','Premature Atrial Contraction',
       'Premature Ventricular Contractions', 'Right Bundle Branch Block','Ventricular Fibrillation']
result=str(index[pred[0]])
result
```

Out[41]:
```
'Left Bundle Branch Block'
```

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 LINK:**

**https://github.com/IBM-EPBL/IBM-Project-27231-1660051100**

**PROJECT DEMO LINK:**

**https://drive.google.com/file/d/189Uj0LWouO95nDIWlW0Xfz1QTTIOVGyD/view?usp=share_link**

(if the video is not loaded in drive please do download the video from the link)