# Classification Of Arrhythmia By Using Deep Learning With 2-d Ecg Spectral Image Representation

**SUBMITTED BY**

**PNT2022TMIDI3766**

**Monish V -622119104063**

**Iyyappan N – 622119104039**

**Jaganathan K – 622119104040**

**Mohammed Salim K - 622119104059**

## Project Report Format

# ECG- ImageBased Heartbeat Classification For Arrhythmia Detection Using IBM Watson Studio

## 1.Introduction:

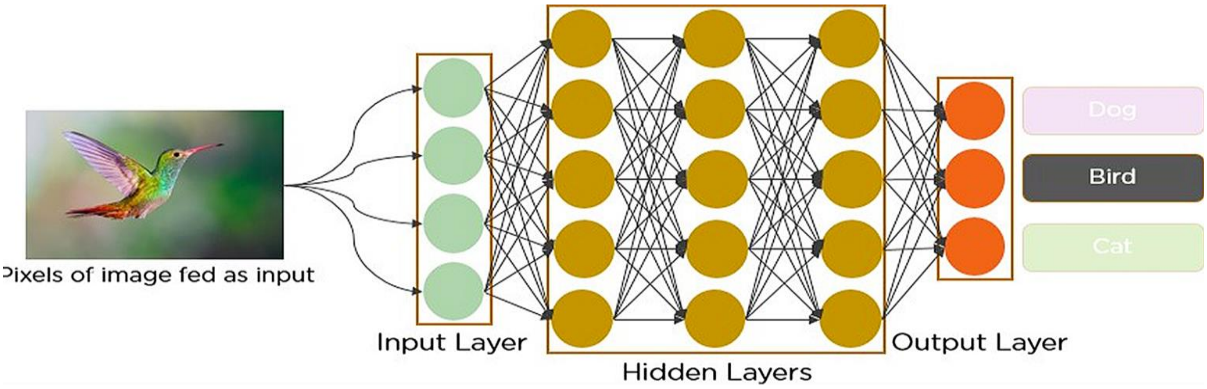
### 1.1. Overview:

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 convolution al 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.

### 1.2. **Purpose:**

In the past few decades, Deep Learning has proved to be a compelling tool because of its ability to handle large amounts of data. The interest to use hidden layers has surpassed traditional techniques, especially in pattern recognition. One of the most popular deep neural networks is Convolution al Neural Networks.



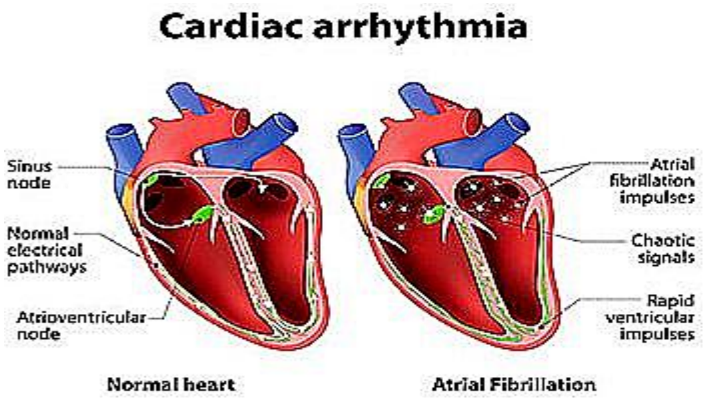
Indeep learning, a convolution al neural network(CNN/ConvNet) is a class of deep neural networks, most commonly applied to analyze visual imagery. Now when we think of a neural network we think about matrix multiplications but that is not the case with ConvNet. It uses a special technique called Convolution. Now in mathematics convolution is a mathematical operation on two functions that produces a third function that expresses how the shape of one is modified by the other.

## 2.Literature Survey:

### 2.1 Existing Problem:

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.

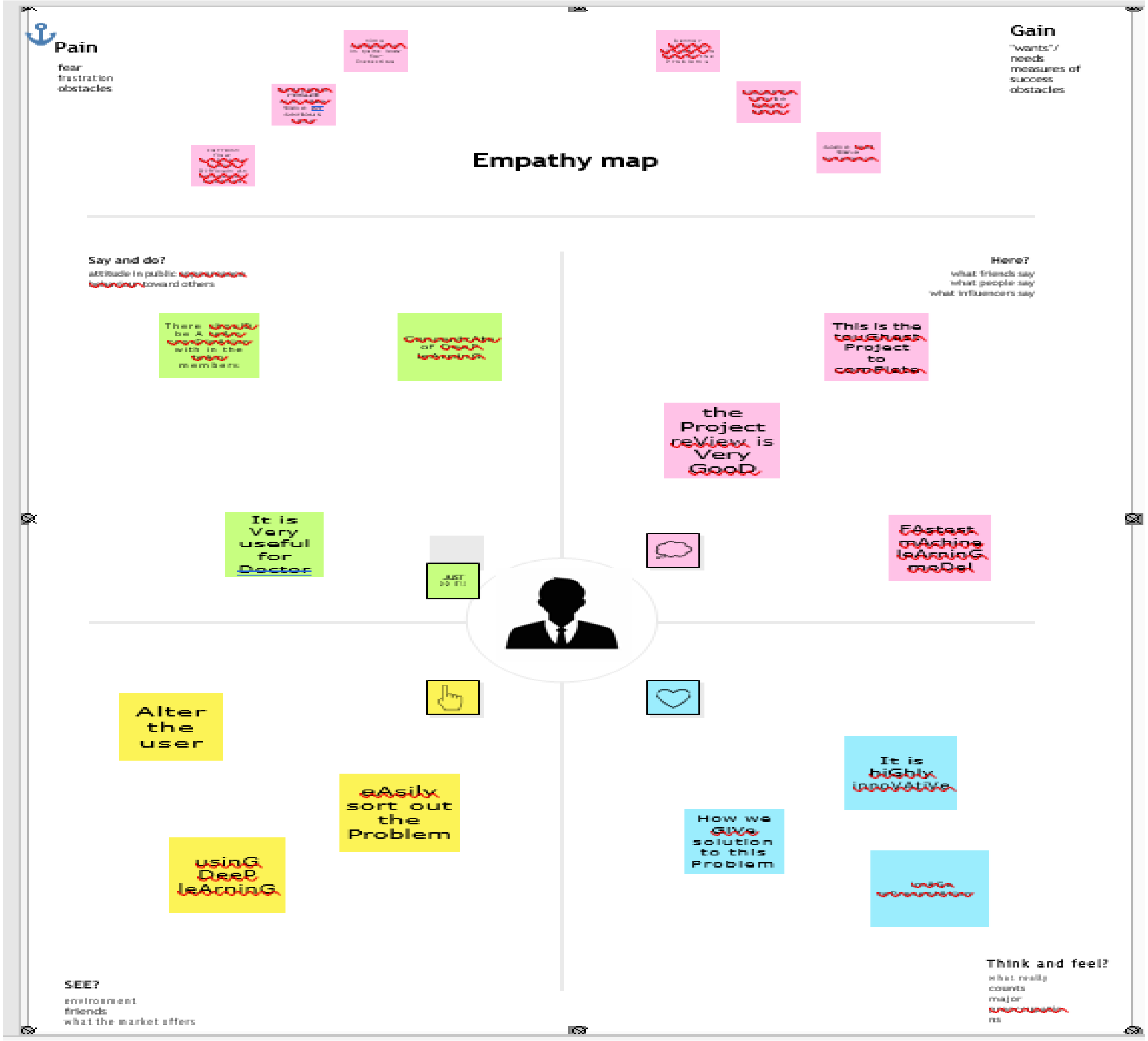


**2.2 Proposed Solution:**

An "ambulatory electrocardiogram" or an ECG) about the size of a postcard or digital camera that the patient will be using for 1 to 2 days, or up to 2 weeks. The test measures the movement of electrical signals or waves through the heart. These signals tell the heart to contract (squeeze) and pump blood. The patient will have electrodes taped to your skin. It's painless, although some people have mild skin irritation from the tape used to attach the electrodes to the chest.They can do everything but shower or bathe while wearing the electrodes. After the test period, patient will go back to see your doctor. They will be downloading the information.

## 3. IDEATION & PROPOSED SOLUTION

### 3.1 Empathy Map Canvas



## 3.2 Ideation & Brainstorming

Brainstorming provides a free and open environment that encourages everyone within a team to participate in the creative thinking process that leads to problem solving. Prioritizing volume over value, out-of-the-box ideas are welcome and built upon, and all participants are encouraged to collaborate, helping each other develop a rich amount of creative solutions.

## 3.3 Proposed Solution

| S.<br>No. | Parameter | Description |
|---|---|---|

| | | |
|---|---|---|
| 1. | Problem Statement (Problem to besolved) | 1. Electrocardiography (ECG) is a method for monitoring the human heart's electrical activity.<br>2. ECG signal is often used by clinical expertsin the collected time arrangement forthe evaluation of any rhythmiccircumstances of a topic.<br>3. The research was carried to make the assignment computerized by displaying the problem withencoder-decoder methods, byusing misfortune appropriation to predict standard or anomalous information. |
| 2. | Idea / Solution description | 1.Electrocardiogram signals have been widely used to identify arrhythmias due totheir non -invasive approach.<br>2.A better alternative is to utilize deep learning models for early automatic identification of cardiac arrhythmia, therebyenhancing diagnosis and treatment. |
| 3. | Novelty / Uniqueness | 1. When the cardiac arrhythmia problem occur, we can find out the pulsewave inminutes.<br>2. It is easy to find out the cardiac problem. |
| 4. | Social Impact/ Customer Satisfaction | 1. This can reducethe arrhythmia problem inthe beginning stage by the pulse wave.<br>2. The user can also use the as a surveillance.3.By the way monitor the patient. |
| 5. | Business Model (Revenue Model) | 1. This application willbe available in themulti-speciality hospital.<br>2. Government providing this type service. |
| 6. | Scalability of the Solution | 1.This application can monitor different phasesimultaneously and can detect cardiacarrhythmia with high accuracy. |

## 3.4 Problem Solution fit

Define CS, fit into CC

### 1. CUSTOMER SEGMENT(S)

CS

Who is your customer?
i.e. working parents of 0-5 y.o. kids

our main target customers are heart specialists(cardiologist), medical labs

### 6. CUSTOMER CONSTRAINTS

CC

What constraints prevent your customers from taking action or limit their choices of solutions? i.e. spending power, budget, no cash, network connection, available devices.

many cardiologist require vast experience to analayze the ECG reports and to identify the abnormal heartbeat.

### 5. AVAILABLE SOLUTIONS

AS

Which solutions are available to the customers when they face the problem or need to get the job done? What have they tried in the past? What pros & cons do these solutions have? i.e. pen and paper is an alternative to digital notetaking

usually experienced cardiologist look into the ECG scan pattern and identify the problem. recently computer aided dignostics has unraveled a new arena of opportunities. different methods to classify types of arrhythmia using machine learning and deep learning exists.The problem is that these architectures are too deep and they take quite some to train and take up some space as well.

Explore AS, differentiate

Focus on J&P, tap into BE, understand RC

### 2. JOBS-TO-BE-DONE / PROBLEMS

J&P

Which jobs-to-be-done (or problems) do you address for your customers? There could be more than one; explore different sides.

- classify different types of arrhythmia for diagnosis and treatment
- Try to gain insight frm the available ECG data about certain specific characterstics related to the disease and its treatment.

### 9. PROBLEM ROOT CAUSE

RC

What is the real reason that this problem exists? What is the back story behind the need to do this job?
i.e. customers have to do it because of the change in regulations.

The reports when analysed manually consumes more time.sometimes even false negative outcome is produced. so this may not be helpful for the patient.

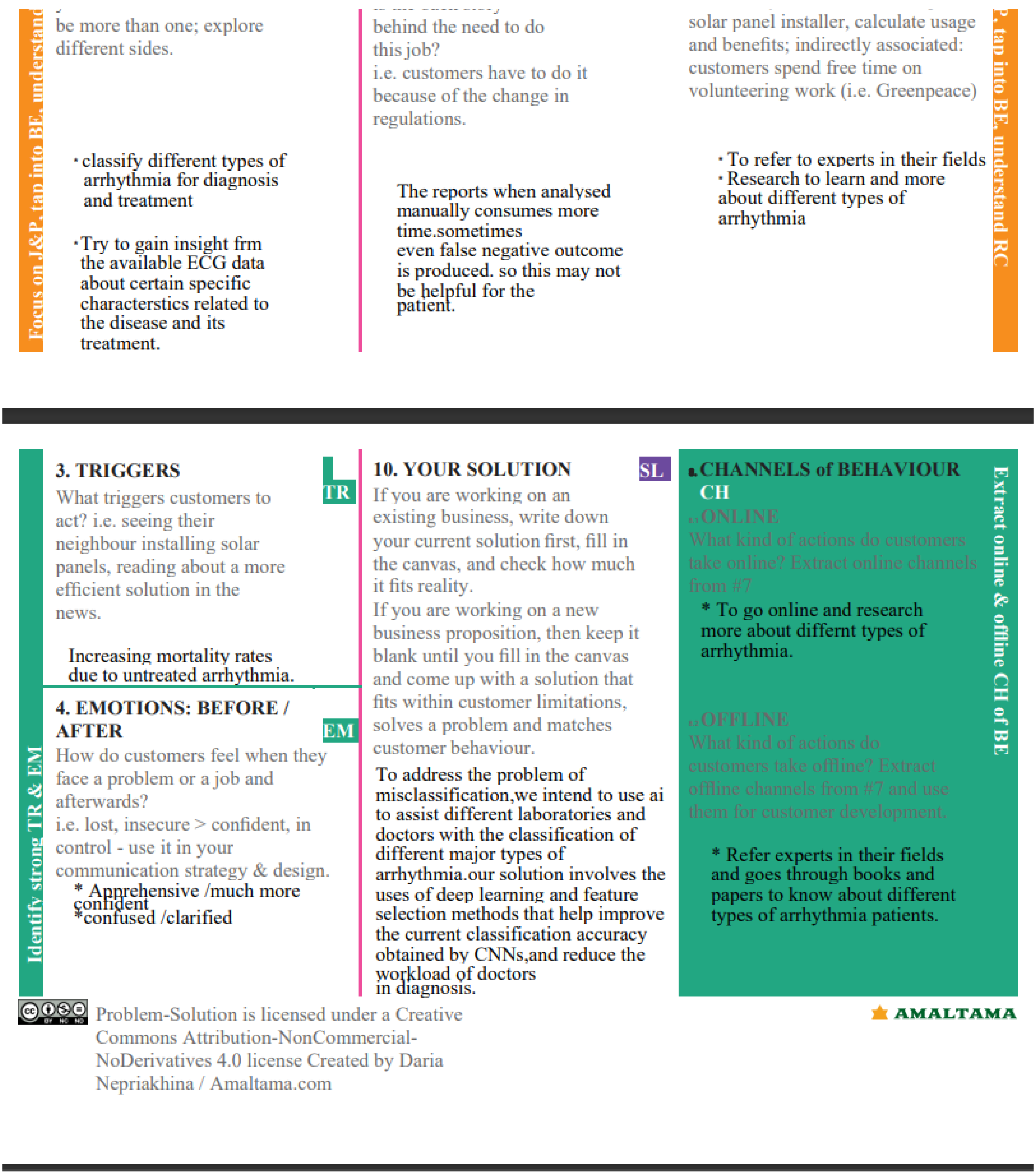
### 7. BEHAVIOUR

BE

What does your customer do to address the problem and get the job done?
i.e. directly related: find the right solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

- To refer to experts in their fields
- Research to learn and more about different types of arrhythmia

Focus on J&P, tap into BE, understand RC



Focus on J&P, tap into BE, understand

be more than one; explore different sides.

• classify different types of arrhythmia for diagnosis and treatment

•Try to gain insight frm the available ECG data about certain specific charactersties related to the disease and its treatment.

behind the need to do this job?
i.e. customers have to do it because of the change in regulations.

The reports when analysed manually consumes more time.sometimes even false negative outcome is produced. so this may not be helpful for the patient.

solar panel installer, calculate usage and benefits; indirectly associated: customers spend free time on volunteering work (i.e. Greenpeace)

• To refer to experts in their fields
• Research to learn and more about different types of arrhythmia

P, tap into BE, understand RC

Identify strong TR & EM

3. TRIGGERS

TR

What triggers customers to act? i.e. seeing their neighbour installing solar panels, reading about a more efficient solution in the news.

Increasing mortality rates due to untreated arrhythmia.

4. EMOTIONS: BEFORE / AFTER

EM

How do customers feel when they face a problem or a job and afterwards?
i.e. lost, insecure > confident, in control - use it in your communication strategy & design.

* Apprehensive /much more confident
*confused /clarified

10. YOUR SOLUTION

SL

If you are working on an existing business, write down your current solution first, fill in the canvas, and check how much it fits reality.
If you are working on a new business proposition, then keep it blank until you fill in the canvas and come up with a solution that fits within customer limitations, solves a problem and matches customer behaviour.

To address the problem of misclassification,we intend to use ai to assist different laboratories and doctors with the classification of different major types of arrhythmia.our solution involves the uses of deep learning and feature selection methods that help improve the current classification accuracy obtained by CNNs,and reduce the workload of doctors in diagnosis.

8.CHANNELS of BEHAVIOUR

CH

8.1 ONLINE

What kind of actions do customers take online? Extract online channels from #7

* To go online and research more about differnt types of arrhythmia.

8.2 OFFLINE

What kind of actions do customers take offline? Extract offline channels from #7 and use them for customer development.

* Refer experts in their fields and goes through books and papers to know about different types of arrhythmia patients.

Extract online & offline CH of BE

AMALTAMA

## 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<br>Confirmation via OTP |
| FR-3 | Get UserInput | Upload image as jpegUpload image as PNG |
| FR-4 | Save Image | Images are saved in theuploads folder |
| FR-5 | Chat withDoctor | Consult with Doctor |
| FR-6 | Report Generation | Get complete Report |

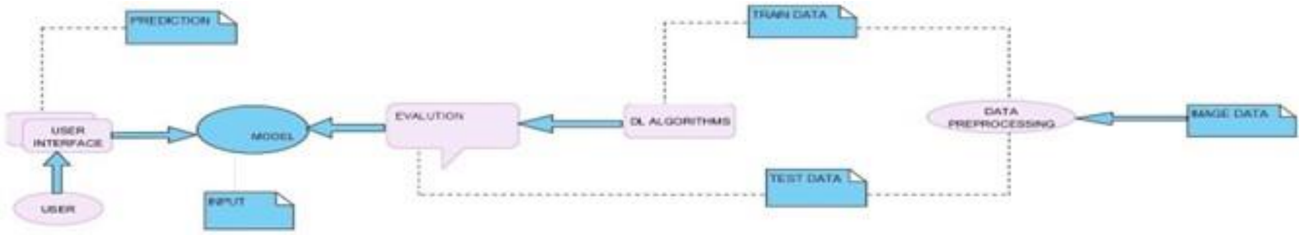
## 4.2 Non- Functional Requirement

| FR No. | Non-Functional Requirement | Description |
|---|---|---|
| NFR-1 | Usability | Classification of Arrhythmia with the help of AI. |
| NFR-2 | Security | User’s data cannot be accessed by Unauthorised people. |
| NFR-3 | Reliability | The systemperforms without failure. |
| NFR-4 | Performance | High accuracy. |
| NFR-5 | Availability | Anyone whois authorised. |
| NFR-6 | Scalability | Does not affect the performance even though used by manyusers. |

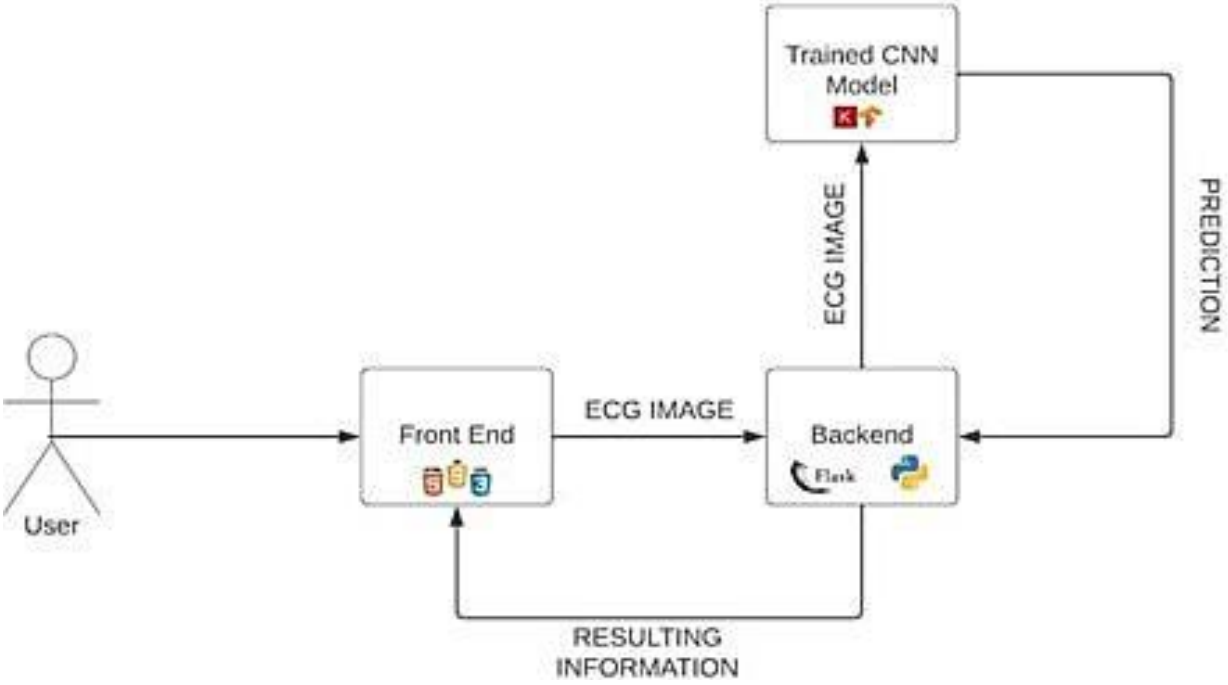
# 5.Project Design:

### 5.1 Data Flow Diagrams:

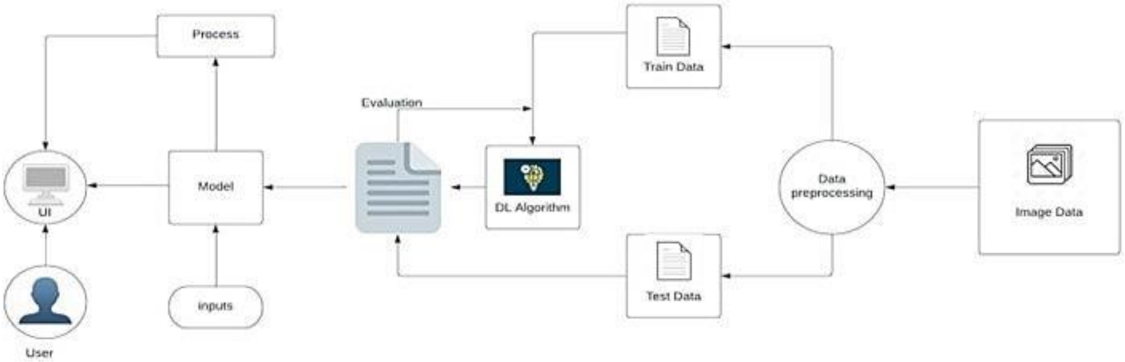
A Data Flow Diagram (DFD) is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It shows how data enters and leaves the system, what changes the information, and where datais stored.



## 5.2 Solution Architecture



### 5.2.1Technical Architecture



## 5.3User stories

| User Type | Functional Requirement(Epic) | User Story Number | User Story / Task | Acceptance criteria | Priority | Release |
|---|---|---|---|---|---|---|

| | | | | | | |
|---|---|---|---|---|---|---|
| Customer (Mobileuser) | Registration | USN-1 | As a user, I can register for the application by entering my email, password, and confirming my password. | I can accessmy account / dashboard | High | Sprint -1 |
| | | USN-2 | As a user, I can register for theapplication using gmail | I can access my account/dashboard | High | Sprint -1 |
| | Confirmation | USN-3 | As a user, I will receive confirmation email once I have registered for the application | I can receive confirmation email &click confirm | High | Sprint -1 |
| Data Input | Getting user input | USN-4 | As a user,I can sharemy input like the medical reports to theapplication | I can proceed withfurther steps withno error | High | Sprint -2 |

## 6.Project Planning & Scheduling

### 6.1 Sprint Planning & Estimation

| S.NO | MILESTONE | ACTIVITIES | DATE |
|---|---|---|---|

| | | | |
|---|---|---|---|
| 1. | Preparation Phase | Pre-requisites | 24 Aug 2022 |
| | | Prior knowledge | 25 Aug 2022 |
| | | Project Structure | 23 Aug 2022 |
| | | Project Flow | 23 Aug 2022 |
| | | Project Objectives | 22 Aug 2022 |
| | | Registrations | 26 Aug 2022 |
| | | Environment Set-up | 27 Aug 2022 |
| 2. | Ideation Phase | Literature Survey | 29 Aug 2022 - 03 Sept 2022 |
| | | Empathy Map | 05 Sept 2022 - 7 Sept 2022 |
| | | Problem Statement | 08 Sept 2022 - 10 Sept2022 |
| | | Ideation | 12 Sept 2022 - 16 Sept 2022 |
| 3. | Project Design Phase -I | Proposed Solution | 19 Sept 2022 - 23 Sept 2022 |
| | | Problem Solution Fit | 24 Sept 2022 - 26 Sept 2022 |
| | | Solution Architecture | 27 Sept 2022 - 30 Sept 2022 |
| 4. | Project Design Phase -II | Customer Journey | 03 Oct 2022 - 08 Oct 2022 |
| | | Requirement Analysis | 09 Oct 2022 - 11 Oct 2022 |

| | | | |
|---|---|---|---|
| | | Data Flow Diagrams | 11 Oct 2022 - 14 Oct 2022 |
| | | Technology Architecture | 15 Oct 2022 - 16 Oct 2022 |
| 5<br>. | Project Planning Phase | Milestones & Tasks | 17 Oct 2022 - 18 Oct 2022 |
| | | Sprint Schedules | 19 Oct 2022 - 22 Oct 2022 |
| 6<br>. | Project Development Phase | Sprint-1 | 24 Oct 2022 - 28 Oct 2022 |
| | | Sprint-2 | 30 Oct 2022 - 04 Nov 2022 |
| | | Sprint-3 | 06 Nov 2022- 11 Nov 2022 |
| | | Sprint-4 | 13 Nov 2022 - 18 Nov 2022 |

## 6.2 Sprint Delivery Schedule

| S.NO | MILESTONE | ACTIVITIES | DATE |
|---|---|---|---|
| 1. | Sprint-1 | 1. Download The Dataset<br>2. Import ImageDataGenerator Library<br>3. Configure ImageDataGenerator class<br>4. ImportLibraries<br>5. Initialize the Model | 24 Oct 2022 – 28 Oct 2022 |

|  |  |  |  |
|---|---|---|---|
| 2. | Sprint – 2 | 1. Register IBM Cloud<br>2. Apply ImageDataGenerator functionality to Trainset and Dataset<br>3. Test the model | 30 Oct 2022 – 04 Nov 2022 |
| 3. | Sprint – 3 | 1. Train the model on IBM<br>2. Create Html files<br>3. Train the Model | 06 Nov 2022 –11 Nov 2022 |
| 4. | print – 4 | 1. Configure The Learning Process<br>2. Build Python code<br>3. Adding Dense Layer<br>4. Adding CNN layer | 13 Nov 2022 –18 Nov 2022 |

## 7.Coding & Solutioning

### 7.1 Feature 1

### 7.1.1 Dataset Collection:

The dataset contains six classes:

a. Left Bundle Branch Block
b. Normal
c. Premature Atrial Contraction
d. Premature Ventricular Contractions
e. Right Bundle Branch Block
f. Ventricular Fibrillation

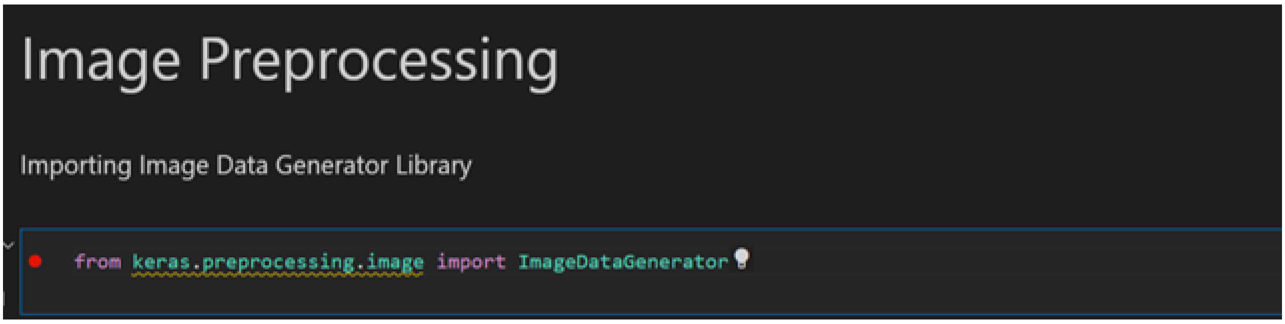
### 7.1.2 Image Preprocessing:

Image Pre-processing includes the following main tasks

1. **Import ImageDataGenerator Library:**

Image data augmentation is a technique that can be used to artificially expand the size of a training dataset by creating modified versions of images in the dataset.

The Keras deep learning neural network library provides the capability to fit models using image data augmentation via the ImageDataGenerator class.



# Image Preprocessing

Importing Image Data Generator Library

```
from keras.preprocessing.image import ImageDataGenerator
```

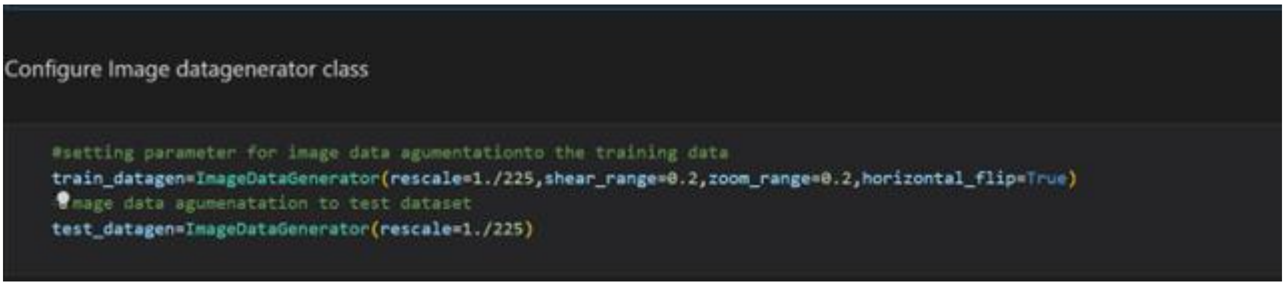
1. **Configure ImageDataGenerator Class:**

There are five main types of data augmentation techniques for image data; specifically:

-

1. Image shifts via the width_shift_range and height_shift_range arguments.
2. Image flips via the horizontal_flip and vertical_flip arguments.
3. Image rotates via the rotation_range argument
4. Image brightness via the brightness_range argument.
5. Image zooms via the zoom_range argument.

An instance of the ImageDataGenerator class can be constructed for train and test.



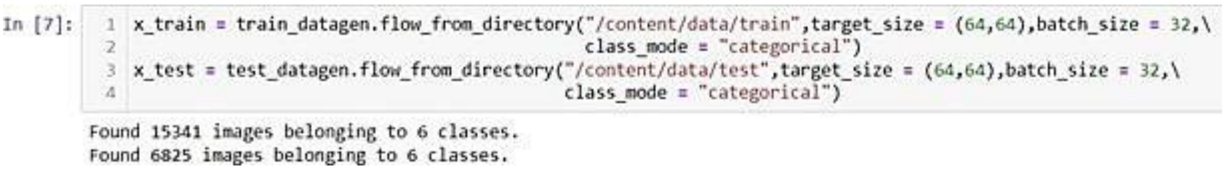
Configure Image datagenerator class

```
#setting parameter for image data agumentationto the training data
train_datagen=ImageDataGenerator(rescale=1./225,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
mage data agumenatation to test dataset
test_datagen=ImageDataGenerator(rescale=1./225)
```

1. **Applying ImageDataGenerator functionality to the trainset and test set:**

We will apply ImageDataGenerator functionality to Trainset and Testset by using the following code

This function will return batches of images from the subdirectories Left Bundle Branch Block, Normal, Premature Atrial Contraction, Premature Ventricular Contractions, Right Bundle Branch Block and Ventricular Fibrillation, together with labels 0 to 5{'Left Bundle Branch Block': 0, 'Normal': 1, 'Premature Atrial Contraction': 2, 'Premature Ventricular Contractions': 3, 'Right Bundle Branch Block': 4, 'Ventricular Fibrillation': 5}



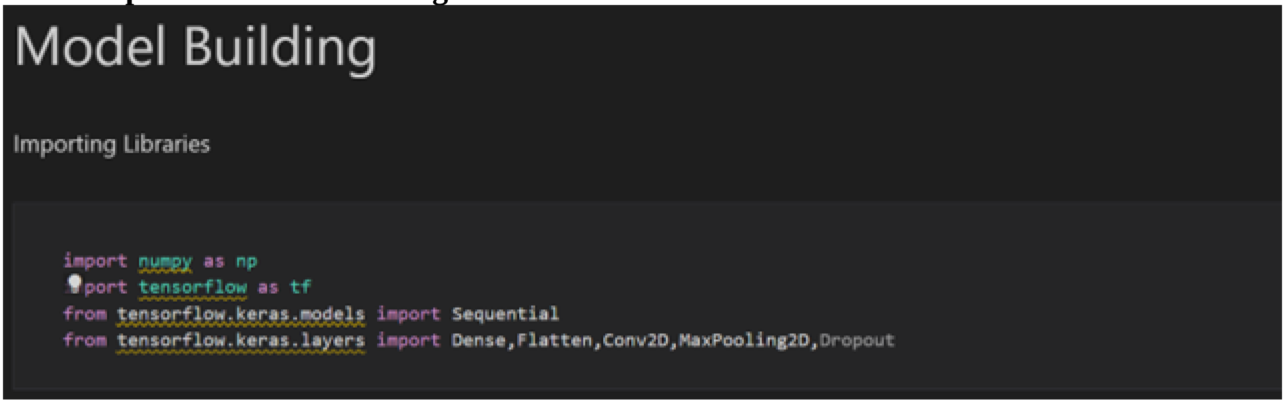
```
In [7]:  x_train = train_datagen.flow_from_directory("/content/data/train",target_size = (64,64),batch_size = 32,\
                                                     class_mode = "categorical")
         x_test = test_datagen.flow_from_directory("/content/data/test",target_size = (64,64),batch_size = 32,\
                                                   class_mode = "categorical")

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

We can see that for training there are 15341 images belonging to 6 classes and for testing there are 6825 images belonging to 6 classes.

## Model Building

We are ready with the augmented and pre-processed image data,we will begin our build our model by following the below steps:

1. **Import the model building Libraries:**



Model Building

Importing Libraries

```
import numpy as np
💡port tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense,Flatten,Conv2D,MaxPooling2D,Dropout
```

1. **Initializing the model:**

Keras has 2 ways to define a neural network:

1. Sequential
2. Function API

The Sequential class is used to define linear initializations of network layers which then, collectively, constitute a model. In our example below, we will use the Sequential constructor to create a model, which will then have layers added to it using the add () method.
Now, will initialize our model.

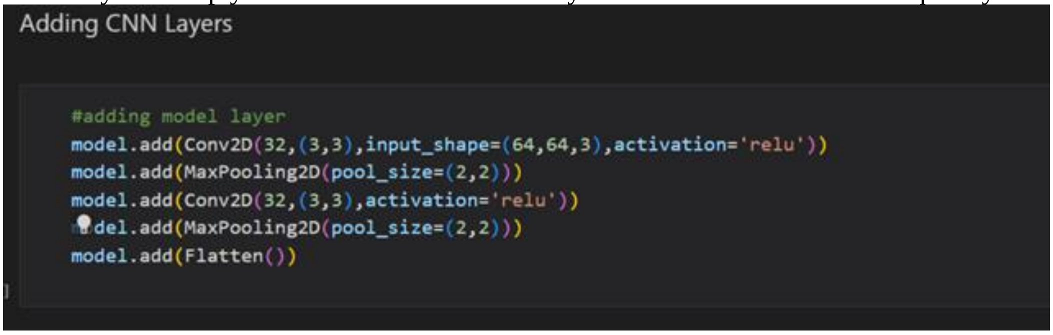
1. **Adding CNN Layers:**

We are adding a convolution layer with an activation function as “relu” and with a small filter size (3,3) and a number of filters as (32) followed by a max-pooling layer.
The Max pool layer is used to downsample the input.

The flatten layer flattens the input

2. **Adding Hidden Layers:**

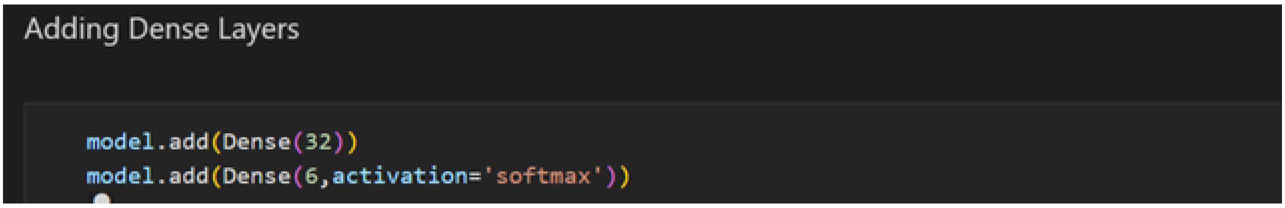
Dense layer is deeply connected neural network layer. It is most common and frequently used layer



Adding CNN Layers

```
#adding model layer
model.add(Conv2D(32,(3,3),input_shape=(64,64,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Conv2D(32,(3,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
```

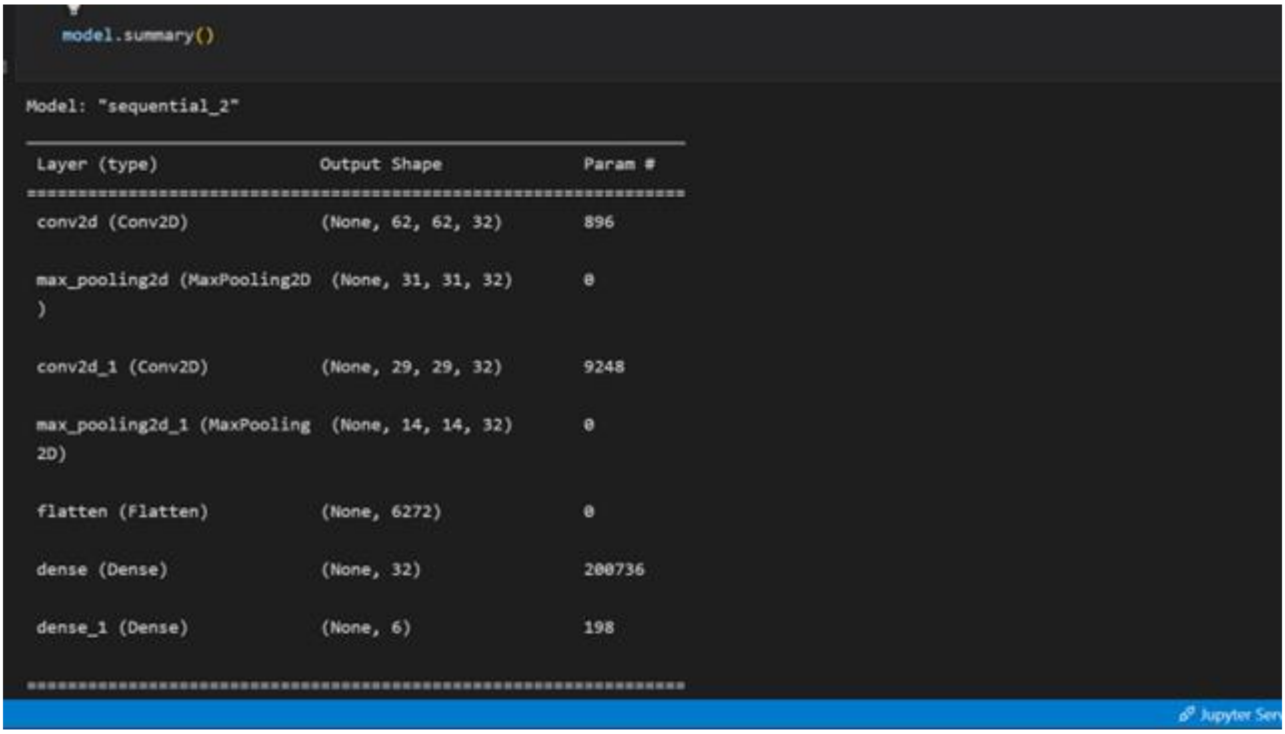
1. **Adding Output Layer:**



Adding Dense Layers

```
model.add(Dense(32))
model.add(Dense(6,activation='softmax'))
```

Understanding the model is very important phase to properly use it for training and prediction purposes. Keras provides a simple method,summary to get the full information about the model and its layers.



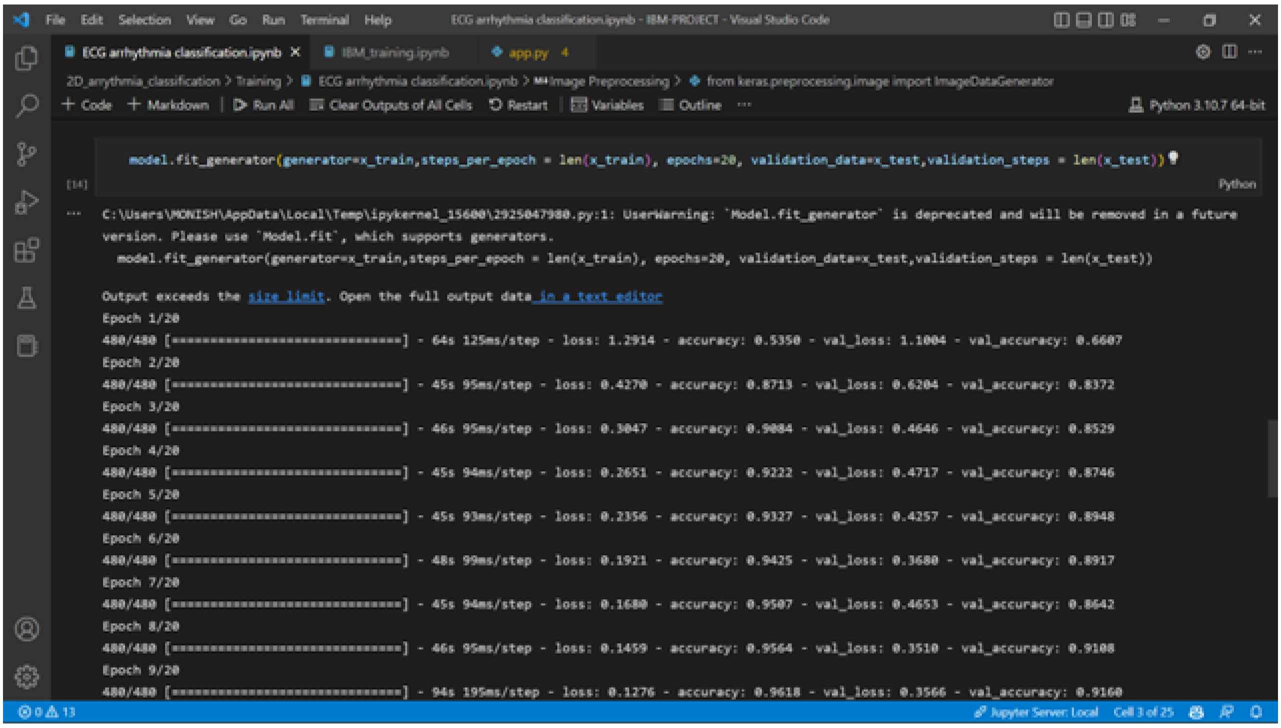
```
model.summary()
```

```
Model: "sequential_2"
_________________________________________________________________
 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

=================================================================
```

1. **Configure the Learning Process:**
1. The compilation is the final step in creating a model. Once the compilation is done, we can move on to the training phase. The loss function is used to find error or deviation in the learning process. Keras requires loss function during the model compilation process.
2. Optimization is an important process that optimizes the input weights by comparing the prediction and the loss function. Here we are using adam optimizer
3. Metrics is used to evaluate the performance of your model. It is similar to loss function, but not used in the training process.

2. **Training the model:**

We will train our model with our image dataset. fit_generator functions used to train a deep learning neural network.


File Edit Selection View Go Run Terminal Help
ECG arrhythmia classification.ipynb - IBM-PROJECT - Visual Studio Code
ECG arrhythmia classification.ipynb
IBM_training.ipynb
app.py 4
2D_arrythmia_classification > Training > ECG arrhythmia classification.ipynb > M↓Image Preprocessing > from keras.preprocessing.image import ImageDataGenerator
+ Code + Markdown | Run All Clear Outputs of All Cells Restart Variables Outline ···
Python 3.10.7 64-bit
[14]
Python
⊗ 0 ⚠ 13
Jupyter Server: Local Cell 3 of 25

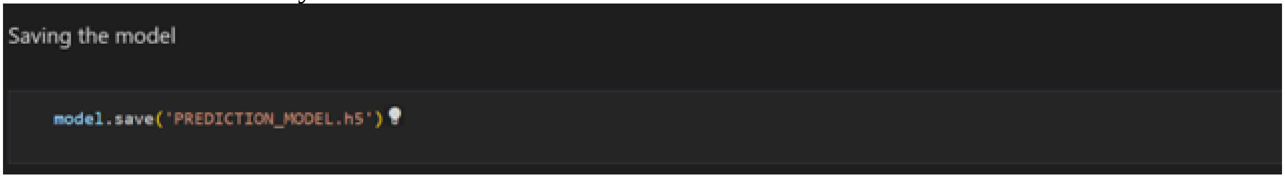
```
model.fit_generator(generator=x_train,steps_per_epoch = len(x_train), epochs=20, validation_data=x_test,validation_steps = len(x_test))
```

```
C:\Users\MONISH\AppData\Local\Temp\ipykernel_15600\2925047980.py:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future
version. Please use `Model.fit`, which supports generators.
  model.fit_generator(generator=x_train,steps_per_epoch = len(x_train), epochs=20, validation_data=x_test,validation_steps = len(x_test))

Output exceeds the size limit. Open the full output data in a text editor
Epoch 1/20
480/480 [==============================] - 64s 125ms/step - loss: 1.2914 - accuracy: 0.5350 - val_loss: 1.1004 - val_accuracy: 0.6607
Epoch 2/20
480/480 [==============================] - 45s 95ms/step - loss: 0.4270 - accuracy: 0.8713 - val_loss: 0.6204 - val_accuracy: 0.8372
Epoch 3/20
480/480 [==============================] - 46s 95ms/step - loss: 0.3047 - accuracy: 0.9084 - val_loss: 0.4646 - val_accuracy: 0.8529
Epoch 4/20
480/480 [==============================] - 45s 94ms/step - loss: 0.2651 - accuracy: 0.9222 - val_loss: 0.4717 - val_accuracy: 0.8746
Epoch 5/20
480/480 [==============================] - 45s 93ms/step - loss: 0.2356 - accuracy: 0.9327 - val_loss: 0.4257 - val_accuracy: 0.8948
Epoch 6/20
480/480 [==============================] - 48s 99ms/step - loss: 0.1921 - accuracy: 0.9425 - val_loss: 0.3680 - val_accuracy: 0.8917
Epoch 7/20
480/480 [==============================] - 45s 94ms/step - loss: 0.1680 - accuracy: 0.9507 - val_loss: 0.4653 - val_accuracy: 0.8642
Epoch 8/20
480/480 [==============================] - 46s 95ms/step - loss: 0.1459 - accuracy: 0.9564 - val_loss: 0.3510 - val_accuracy: 0.9108
Epoch 9/20
480/480 [==============================] - 94s 195ms/step - loss: 0.1276 - accuracy: 0.9618 - val_loss: 0.3566 - val_accuracy: 0.9160
```

1. **Saving the model:**

The model is saved with .h5 extension as follows

An H5 file is a data file saved in the Hierarchical Data Format (HDF). It contains multidimensional arrays of scientific data.



Saving the model

```
model.save('PREDICTION_MODEL.h5')
```

## 7.2 Feature 2

### Application Building:

In this section, we will be building a web application that is integrated into the model we built. A UI is provided for the uses where he has uploaded an image. The uploaded image is given to the saved model and prediction is showcased on the UI.

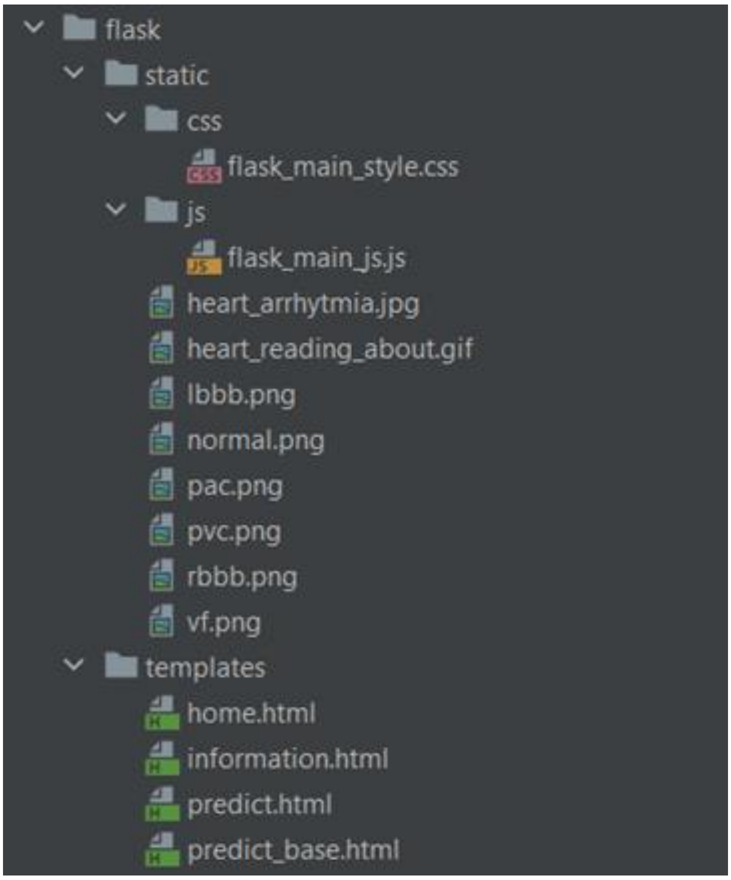
This section has the following tasks

1. **Building HTML Pages:**

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

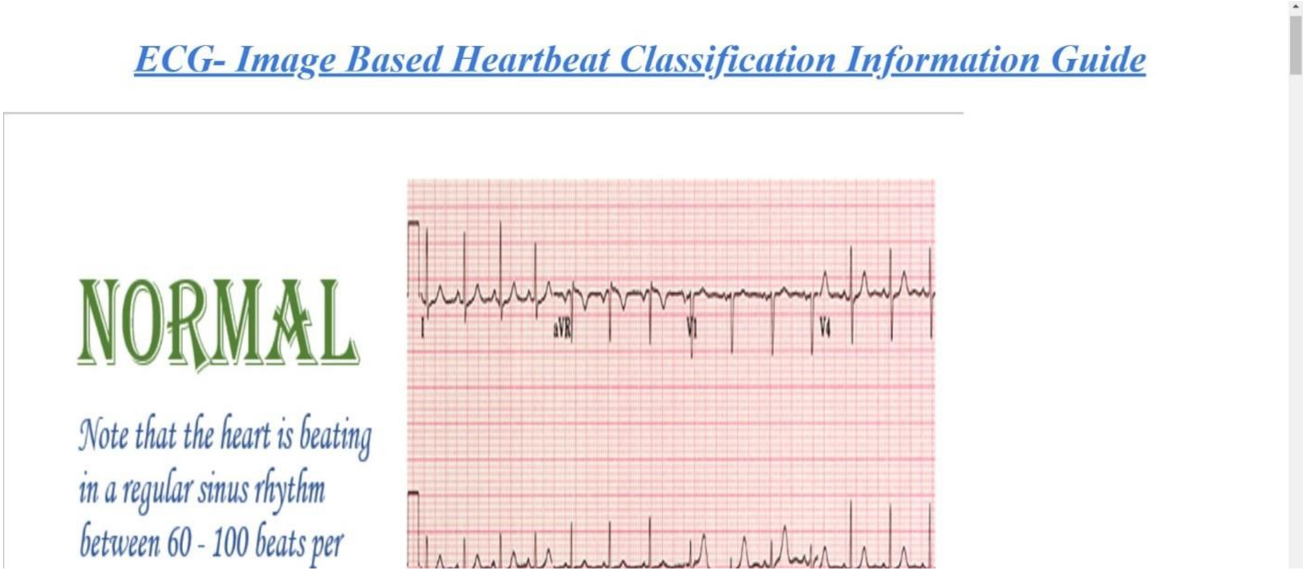
2. Here, we created 4 html pages- home.html, predict_base.html, predict.html, information.html

3. home.html displays the home page.



```
flask
    static
        css
            flask_main_style.css
        js
            flask_main_js.js
        heart_arrhytmia.jpg
        heart_reading_about.gif
        lbbb.png
        normal.png
        pac.png
        pvc.png
        rbbb.png
        vf.png
    templates
        home.html
        information.html
        predict.html
        predict_base.html
```

- information.html displays all important details to be known about ECG.

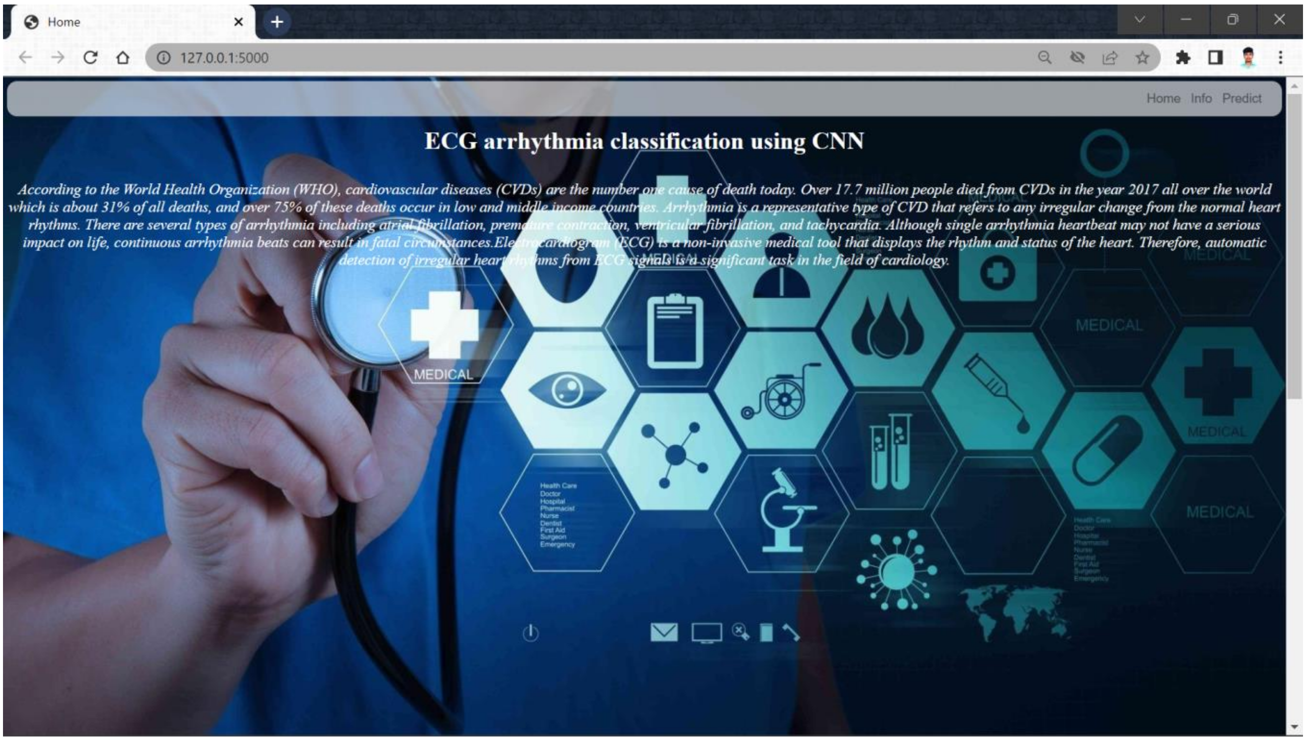


ECG- Image Based Heartbeat Classification Information Guide

NORMAL

Note that the heart is beating in a regular sinus rhythm between 60 - 100 beats per

1. predict-base.html and predict.html accept input from the user and predicts the value



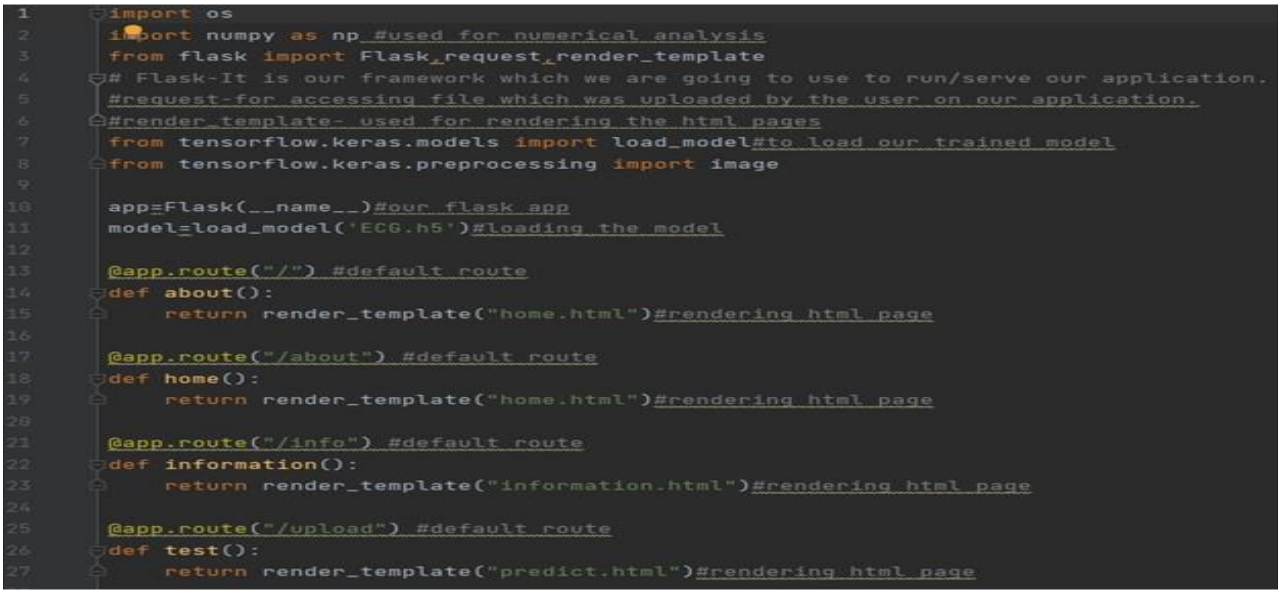
ECG arrhythmia classification using CNN

*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.*

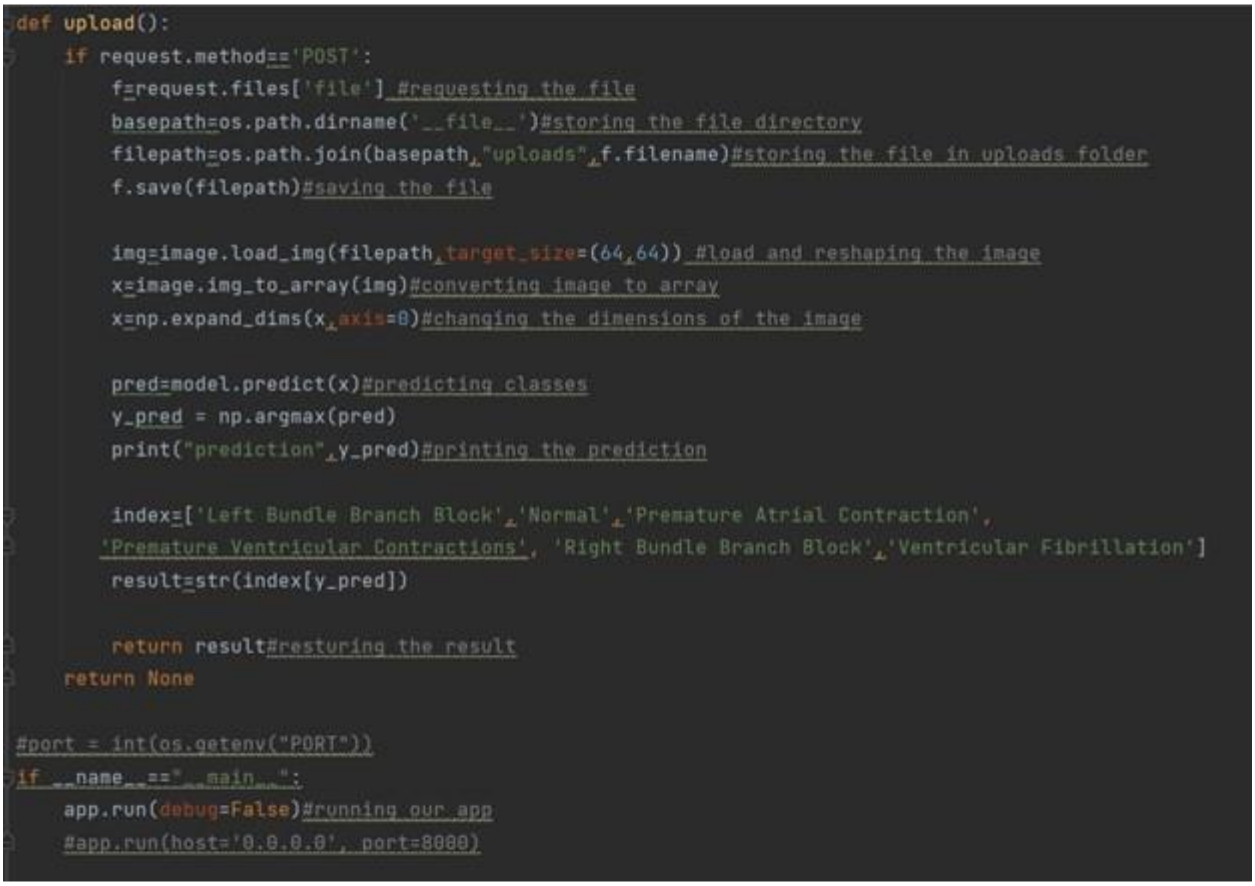
## Building server-side script:

We will buildthe flask file 'app.py' whichis a web framework writtenin python for server-side scripting.

1. The app starts running whenthe " name " constructor is called in main.
2. render_template is used to returnHTML file.
3. "GET" method is used to take inputfrom the user.
4. "POST" method is used to display the output to the user.



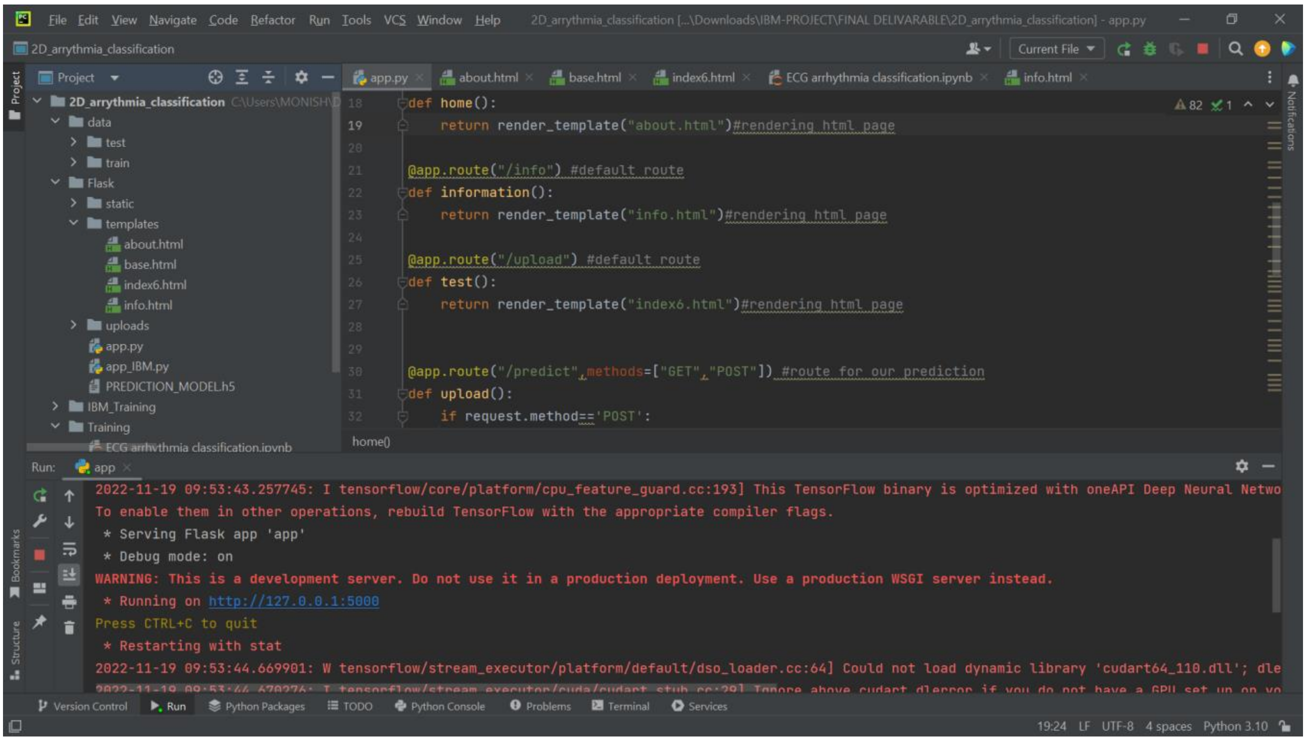
```
import os
import numpy as np #used for numerical analysis
from flask import Flask,request,render_template
# Flask-It is our framework which we are going to use to run/serve our application.
#request-for accessing file which was uploaded by the user on our application.
#render_template- used for rendering the html pages
from tensorflow.keras.models import load_model#to load our trained model
from tensorflow.keras.preprocessing import image

app=Flask(__name__)#our flask app
model=load_model('ECG.h5')#loading the model

@app.route("/") #default route
def about():
    return render_template("home.html")#rendering html page

@app.route("/about") #default route
def home():
    return render_template("home.html")#rendering html page

@app.route("/info") #default route
def information():
    return render_template("information.html")#rendering html page

@app.route("/upload") #default route
def test():
    return render_template("predict.html")#rendering html page
```



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

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

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

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

        return result#resturing the result
    return None

#port = int(os.getenv("PORT"))
if __name__=="__main__":
    app.run(debug=False)#running our app
    #app.run(host='0.0.0.0', port=8080)
```

### 1. Running The App:

C:\Users\MONISH\IBM_PROJECT\2D Arrythmia Classification\flask\app.py



File Edit View Navigate Code Refactor Run Tools VCS Window Help
2D_arrythmia_classification [...\Downloads\IBM-PROJECT\FINAL DELIVARABLE\2D_arrythmia_classification] - app.py
2D_arrythmia_classification
Current File
Project
app.py
about.html
base.html
index6.html
ECG arrhythmia classification.ipynb
info.html
2D_arrythmia_classification C:\Users\MONISH\D
data
test
train
Flask
static
templates
about.html
base.html
index6.html
info.html
uploads
app.py
app_IBM.py
PREDICTION_MODEL.h5
IBM_Training
Training
ECG arrhythmia classification.ipynb
home()
Run:
app
Version Control
Run
Python Packages
TODO
Python Console
Problems
Terminal
Services
19:24 LF UTF-8 4 spaces Python 3.10

```python
def home():
    return render_template("about.html")#rendering html page

@app.route("/info") #default route
def information():
    return render_template("info.html")#rendering html page

@app.route("/upload") #default route
def test():
    return render_template("index6.html")#rendering html page


@app.route("/predict",methods=["GET","POST"]) #route for our prediction
def upload():
    if request.method=='POST':
```

```
2022-11-19 09:53:43.257745: I tensorflow/core/platform/cpu_feature_guard.cc:193] This TensorFlow binary is optimized with oneAPI Deep Neural Netwo
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
 * Serving Flask app 'app'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on http://127.0.0.1:5000
Press CTRL+C to quit
 * Restarting with stat
2022-11-19 09:53:44.669901: W tensorflow/stream_executor/platform/default/dso_loader.cc:64] Could not load dynamic library 'cudart64_110.dll'; dle
2022-11-19 09:53:44.670276: I tensorflow/stream_executor/cuda/cudart_stub.cc:29] Ignore above cudart dlerror if you do not have a GPU set up on yo
```

Navigate to the localhost (http://127.0.0.1:5000/)where you can view your web page.

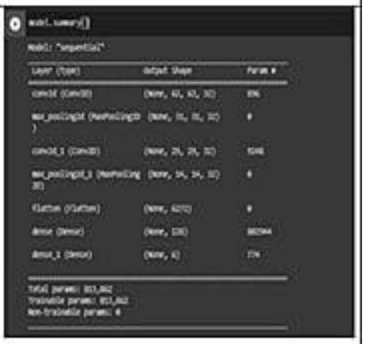
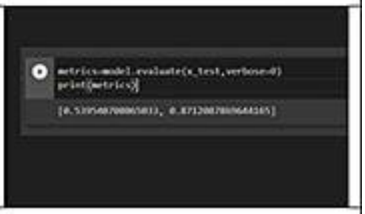
## 8.TESTING:

## 8.1PERFORMANCE TESTING:

**Model Performance Testing:**

Trainin
Validat

Project team shall fill the following information in model performance testing template.

| S.No. | Parameter | Values | Screenshot |
|---|---|---|---|
| 1. | Model Summary | - |  |
| 2. | Accuracy | Training Accuracy - 0.358437687158<br>Validation Accuracy - 0.910915732383 |  |
| 3. | Confidence Score (Only Yolo Projects) | Class Detected -<br>Confidence Score - | - |

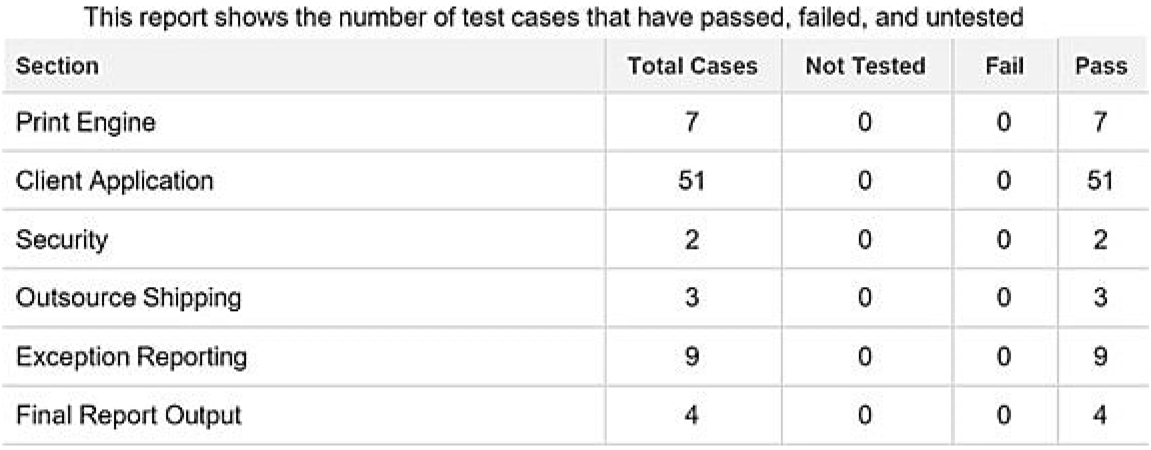
```
model.summary()

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, 128)               802944

 dense_1 (Dense)             (None, 6)                 774

=================================================================
Total params: 813,862
Trainable params: 813,862
Non-trainable params: 0
_________________________________________________________________
```

```
metrics=model.evaluate(x_test,verbose=0)
print(metrics)

[0.5395407000065013, 0.8712007880644165]
```

a. **USER ACCEPTANCE TESTING**

This report shows the count of the bugs at each severity level, and how they were fixed.

| Resolution | Severity 1 | Severity 2 | Severity 3 | Severity 4 | Subtotal |
|---|---|---|---|---|---|
| By Design | 5 | 4 | 2 | 3 | 14 |
| Duplicate | 1 | 0 | 3 | 0 | 4 |
| External | 2 | 3 | 0 | 1 | 6 |
| Fixed | 9 | 2 | 4 | 15 | 30 |
| Not Reproduced | 0 | 0 | 1 | 0 | 1 |
| Skipped | 0 | 0 | 1 | 1 | 2 |
| Won't Fix | 0 | 5 | 2 | 1 | 8 |
| Totals | 17 | 14 | 13 | 21 | 65 |

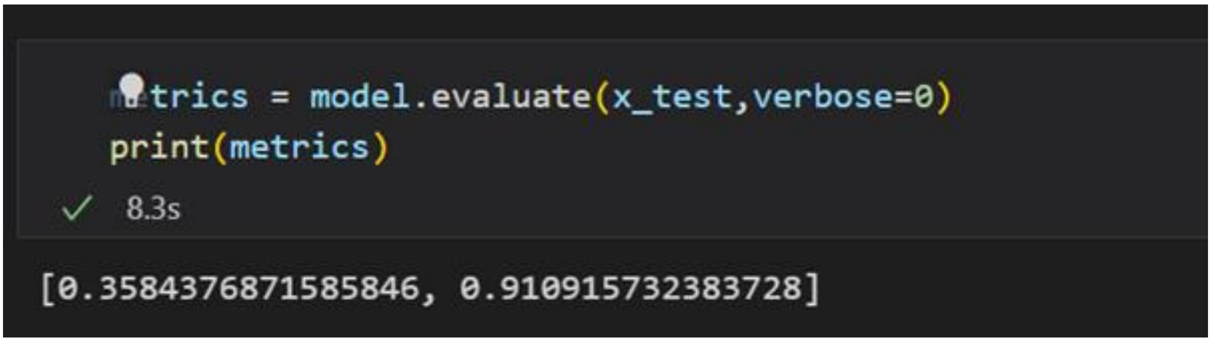


This report shows the number of test cases that have passed, failed, and untested

| Section | Total Cases | Not Tested | Fail | Pass |
|---|---|---|---|---|
| Print Engine | 7 | 0 | 0 | 7 |
| Client Application | 51 | 0 | 0 | 51 |
| Security | 2 | 0 | 0 | 2 |
| Outsource Shipping | 3 | 0 | 0 | 3 |
| Exception Reporting | 9 | 0 | 0 | 9 |
| Final Report Output | 4 | 0 | 0 | 4 |

# 9.RESULT

## 9.1 PERFORMANCE METRICS:



```
metrics = model.evaluate(x_test,verbose=0)
print(metrics)
```

✓ 8.3s

```
[0.3584376871585846, 0.910915732383728]
```

# 10. Advantages & Disadvantages:

## 10.1Advantages:

i. The proposed model predicts Arrhythmia in images with a high accuracy rate of nearly96%

1. The early detection of Arrhythmia gives better understanding of disease causes, initiates therapeutic interventions and enables developing appropriate treatments.

a. **Disadvantages:**

- Not useful for identifying the different stages of Arrhythmiadisease.
- Not useful in monitoring motor symptoms

## Applications :

i. It is useful for identifying the arrhythmia disease at an earlystage.
ii. It is useful in detecting cardiovascular disorders

.

# 11.Conclusion:

1. Cardiovascular disease is a major health problem in today's world. The early diagnosis of cardiac arrhythmia highly relies on the ECG.

2. Unfortunately, the expert level of medical resources is rare, visually identify the ECG signal is challenging and time-consuming.
3. The advantages of the proposed CNN network have been put to evidence.
4. 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.

## 12.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

## 13.Appendix

### ECG Arrythmia classification.ipnb

### Image Preprocessing

Importing Image Data Generator Library

```
from keras.preprocessing.image import ImageDataGenerator
```

Configure Image datagenerator class

```
#setting parameter for image data agumentationto the training data

train_datagen=ImageDataGenerator(rescale=1./225,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)

#image data agumenatation to test dataset

test_datagen=ImageDataGenerator(rescale=1./225)
```

### Model Building

Importing Libraries

```
import numpy as np

import tensorflow as tf

from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense,Flatten,Conv2D,MaxPooling2D,Dropout
model= Sequential()
```

Adding CNN Layers

```
#adding model layer

model.add(Conv2D(32,(3,3),input_shape=(64,64,3),activation='relu'))

model.add(MaxPooling2D(pool_size=(2,2)))

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

model.add(MaxPooling2D(pool_size=(2,2)))

model.add(Flatten())
```

Adding Dense Layers

```
model.add(Dense(32))

model.add(Dense(6,activation='softmax'))



model.summary()
```

Configuring the learning process

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

Train the Model

```
model.fit_generator(generator=x_train,steps_per_epoch = len(x_train), epochs=20,
validation_data=x_test,validation_steps = len(x_test))

model.save('PREDICTION_MODEL.h5')
```

Test the Model

```
model=tf.keras.models.load_model(r'C:/Users/MONISH\Downloads/IBM-
PROJECT/2D_arrythmia_classification/Training/PREDICTION_MODEL.H5')

img=tf.keras.utils.load_img(r'C:/Users/MONISH\Downloads/IBM-PROJECT/2D_arrythmia_classification/Flask/uploads/PAC.png',target_size=(64,64))

x=tf.keras.utils.img_to_array(img)

x=np.expand_dims(x,axis=0)

pred = model.predict(x)

y_pred=np.argmax(pred)

y_pred
```

```
model=tf.keras.models.load_model(r'C:/Users/MONISH\Downloads/IBM-PROJECT/2D_arrythmia_classification/Training/PREDICTION_MODEL.H5')

img=tf.keras.utils.load_img(r'C:/Users/MONISH\Downloads/IBM-PROJECT/2D_arrythmia_classification/Flask/uploads/PAC.png',target_size=(64,64))

x=tf.keras.utils.img_to_array(img)

x=np.expand_dims(x,axis=0)

pred = model.predict(x)

y_pred=np.argmax(pred)

y_pred
```

# App.py

```
import os
import numpy as np
from flask import Flask,request,render_template
from keras.models import load_model
from keras.utils import load_img
from keras.utils import img_to_array
```

```
 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("index6.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=load_img(filepath,target_size=(64,64))
        x=img_to_array(img)
        x=np.expand_dims(x,axis=0)
        pred=model.predict_classes(x)
        print("prediction",pred)
        index=['Left Bundle Branch Block','Normal','Premature Atrial Contraction','Premature Ventricular Contraction','Right Bundle Branch Block','Ventricular Fibrillation']
        result=str(index[pred[0]])
        return result
    return None
port=int(os.getenv("PORT"))
if __name__=="__main__":
    app.run(debug=False)
```

## HTML

## About html

```
<!DOCTYPE html>

<html>

<head>

<title>Home</title>

<style>

body

{

    background-image: url("https://getwallpapers.com/wallpaper/full/6/b/b/923092-best-dentist-wallpapers-2000x1153.jpg");

    background-size: cover;

}
```

```
.pd{

padding-bottom:100%;}

.navbar

{

margin: 0px;

padding:20px;

background-color:white;

opacity:0.6;

color:black;

font-family:'Roboto',sans-serif;

font-style: italic;

border-radius:20px;

font-size:25px;

}

a

{

color:grey;

float:right;

text-decoration:none;

font-style:normal;

padding-right:20px;

}

a:hover{

background-color:black;

color:white;

border-radius:15px;0

font-size:30px;

padding-left:10px;
}
p
{
color:white;
font-style:italic;
font-size:30px;
}
</style>
</head>
<body>
<div class="navbar">
<a href="/upload" >Predict</a>
<a href="/info">Info</a>
<a href="/about">Home</a>
<br>
</div>
<br>
<center><b class="pd"><font color="white" size="15" font-family="Comic Sans MS" >ECG arrhythmia classification using CNN</font></b></center>
<div>
<br>
<center>
<p>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.

</p>

</center>

</div>

</body>
</html>
```

## Base.html

```
html lang="en">



<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

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

    <title>Predict</title>

    <link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css" rel="stylesheet">

    <script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>

    <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>

    <script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>

    <link href="{{ url_for('static', filename='css/main.css') }}" rel="stylesheet">

<style>

.bar

{

margin: 0px;

padding:20px;

background-color:white;
opacity:0.6;

color:black;

font-family:'Roboto',sans-serif;

font-style: italic;

border-radius:20px;

font-size:25px;

}

a

{

color:grey;

float:right;

text-decoration:none;

font-style:normal;

padding-right:20px;

}

a:hover{

background-color:black;

color:white;

border-radius:15px;0

font-size:30px;

padding-left:10px;

}

body

{

    background-image: url("https://img.freepik.com/free-photo/medical-stethoscope-isolated-with-black-background-medical-concept-stethoscope-black-background-with-space-text-health-concept-medical-conceptual_1391-769.jpg?w=996&t=st=1668317501~exp=1668318101~hmac=0a7798c89866846d33766327d1953225ada8858ad9f4394f9c6eaea01dc4e837");

    background-size:cover;

}
```

```
</style>
</head>

<body>
<div class="bar">
<a href="/upload" >Predict</a>
<a href="/info">Info</a>
<a href="/about">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>
```

## Index6.html

```
{% extends "base.html" %} {% block content %}


<h2 style="color:white;font-family:Times New Roman;font-size:60"><center>ECG Arrhythmia
Classification</center></h2>


<div>
```

```
    <form id="upload-file" method="post" enctype="multipart/form-data">

    <center>    <label for="imageUpload" class="upload-label">

            Choose...

        </label>

        <input type="file" name="file" id="imageUpload" accept=".png, .jpg, .jpeg">

    </center></form>



   <center> <div class="image-section" style="display:none;">

        <div class="img-preview">

            <div id="imagePreview">

            </div></center>

        </div>

        <center><div>

            <button type="button" class="btn btn-primary btn-lg " id="btn-predict">Predict!</button>

        </div></center>

    </div>



    <div class="loader" style="display:none;"></div>



    <h3 style="color:white" id="result">

        <span> </span>

    </h3>



</div>



</div>



{% endblock %}
```

**Info.html**

```
<!DOCTYPE html>
<html>
<head>
<style>
.navbar
{
margin: 0px;
padding:20px;
background-color:white;
opacity:0.6;
color:black;
font-family:'Roboto',sans-serif;
font-style: italic;
border-radius:20px;
font-size:25px;
}
a
{
color:grey;
float:right;
text-decoration:none;
font-style:normal;
padding-right:20px;
}
a:hover{
background-color:black;
```

```
color:white;
border-radius:15px;0
font-size:30px;
padding-left:10px;
}

img{
width:550px;
height:400px;
padding:10px;
margin-top:0px;
}
img:hover{
border-radius:100px;
border-color:grey;
border-shadow:10px;

}
body{
   background-image: url("https://i.pinimg.com/originals/a5/70/af/a570af57ecbcfbbdd2429f2af11c8579.gif");
    background-size: cover;
}
h1{
font-size:60px;
text-align:center;
color:white;
font-style:italic;
font-weight:bolder;
```

```
}
div{
margin-left:50px;
}
img{
width:1100px;
height:600px;
padding:10px;
margin-top:0px;
}
img:hover{
border-radius:100px;
border-color:grey;
border-shadow:10px;
}
</style>
<title>Info</title>
</head>
<body>
<div class="navbar">
<a href="/upload" >Predict</a>
<a href="/info">Info</a>
<a href="/about">Home</a>
<br>
</div>
<div>
<h1><u>ECG</u></h1>
<div>
```

```
<span><img src="/static/normal.png" title="normal"></span>

<span><img src="/static/vf.png" title="vf"></span>

<span><img src="/static/pac.png" title="pac"></span>

<span><img src="/static/pvc.png" title="pvc"></span>

<span><img src="/static/rbbb.png" title="rbbb"></span>

<span><img src="/static/lbbb.png" title="lbbb"></span>

<div>

</body>
</html>
```

**Train The Model On IBM Watson.ipynb**

Importing Image Data Generator Library

In [2]:

```
pwd

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='Rb_98YMyoN-zzee4ztUdxx8U4xexG4hGYrxQKq1GidT0',
    ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
    config=Config(signature_version='oauth'),
    endpoint_url='https://s3.private.us.cloud-object-storage.appdomain.cloud')

bucket = 'arrythmiaclassification-donotdelete-pr-iyy1hnmipwjeum'
object_key = 'data - Copy.zip'

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

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

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)

ls
```

```
from keras.preprocessing.image import ImageDataGenerator
```

Configure Image datagenerator class

```
#setting parameter for image data agumentationto the training data
train_datagen=ImageDataGenerator(rescale=1./225,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
#image data agumenatation to test dataset
test_datagen=ImageDataGenerator(rescale=1./225)
```

Applying ImageDataGenerator Functionality to train and test Dataset

```
#performing data agumentation to train the dataset

x_train=train_datagen.flow_from_directory(directory=r'/home/wsuser/work/data -
Copy/train',target_size=(64,64),batch_size=32,class_mode='categorical')
#performing agumentation to test the dataset
x_test=test_datagen.flow_from_directory(directory=r'/home/wsuser/work/data -
Copy/test',target_size=(64,64),batch_size=32,class_mode='categorical')
```

## Model Building

Importing Libraries

```
import numpy as np
import tensorflow
from tensorflow.keras.models import Sequential
from tensorflow.keras import layers

from tensorflow.keras.layers import Dense,Flatten

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

model= keras.Sequential()
```

Adding CNN Layers

```
#adding model layer
model.add(Conv2D(32,(3,3),input_shape=(64,64,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Conv2D(32,(3,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
```

Adding Dense Layers

```
model.add(Dense(32))
model.add(Dense(6,activation='softmax'))

model.summary()
```

Configuring the learning process

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

## Train 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))
```

## Saving the model

```
model.save('ECG.h5')

!pip install watson-machine-learning-client


from ibm_watson_machine_learning import APIClient
wml_credentials = {
    "url":"https://us-south.ml.cloud.ibm.com",
    "apikey":"VFECcIuGn6VBQtE-m_AjCrHtiyr7oMqmlk9Cc9xXG0_u"
}
client = APIClient(wml_credentials)

client


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'])

space_uid=guid_from_space_name(client,'arrythmia-prediction')
print("Space UID ="+space_uid)


client.set.default_space(space_uid)


client.software_specifications.list(200)


software_sapce_uid=client.software_specifications.get_uid_by_name("tensorflow_rt22.1-py3.9")

software_sapce_uid


ls


!tar -zcvf arrythmia-classification.tgz ECG.h5


model_details=client.repository.store_model(model='arrythmia-classification.tgz',meta_props={
    client.repository.ModelMetaNames.NAME: "arrythmia-classification",
    client.repository.ModelMetaNames.TYPE: 'tensorflow_2.7',
    client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_sapce_uid
})

model_id = client.repository.get_model_id(model_details)

model_id
```

```
ls
```

Test the Model

```
from tensorflow.keras.models import load_model
from keras.preprocessing import image
model=load_model('/Users/anshumanr/Documents/Externship/Project/Training/ECG.h5')
img=image.load_img("/Users/anshumanr/Documents/Externship/Project/Flask/uploads/PAC.png",target_size=(64,64))
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
pred = model.predict(x)
y_pred=np.argmax(pred)
y_pred
```

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

# Train The Model On IBM Watson-1.ipynb

```
!pip install watson-machine-learning-client --upgrade
```

```
!pip install ibm_watson_machine_learning
```

```
from ibm_watson_machine_learning import APIClient
wml_credentials={
   "url":"https://us-south.ml.cloud.ibm.com",
   "apikey":"VFECcIuGn6VBQtE-m_AjCrHtiyr7oMqmlk9Cc9xXG0_u"
}
client=APIClient(wml_credentials)
```

```
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'])
```

```
space_uid=guid_from_space_name(client,'arrythmia-prediction')
print("Space UID ="+space_uid)
Space UID =9c0d2961-8749-422a-91cc-7024e8706d70
```

```
client.set.default_space(space_uid)
```

```
client.repository.download("9b2bf570-e6ab-4257-8662-215c5655becd",'my_model.tar.gb')
```

# IMAGE PREPROCESSING.ipynb

Importing Image Data Generator Library

```
!pip install keras
```

```
!pip install numpy

!pip install tensorflow
```

```
from keras.preprocessing.image import ImageDataGenerator
```

Configure Image datagenerator class

```
#setting parameter for image data agumentationto the training data
train_datagen=ImageDataGenerator(rescale=1./225,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
#image data agumenatation to test dataset
test_datagen=ImageDataGenerator(rescale=1./225)
```

Applying ImageDataGenerator Functionality to train and test Dataset

```
#performing data agumentation to train the dataset
x_train=train_datagen.flow_from_directory(directory=r'C:\Users\User1\Downloads\data\data\train',target_size=(64,64),batch_size=32,class_mode='categorical')
#performing agumentation to test the dataset
x_test=test_datagen.flow_from_directory(directory=r'C:\Users\User1\Downloads\data\data\test',target_size=(64,64),batch_size=32,class_mode='categorical')
```

## Github Respository:

IBM-Project-33857-1660227986/Project Designing & Planning at main · IBM-EPBL/IBM-Project-33857-1660227986 (github.com)

## Demo video link

https://youtu.be/IPalxImsTLQ