Team ID	**PNT2022TMID00351**
Project Name	Classification of arrhythmia by using deep learning with 2-d ECG Spectral Image Representation

Classification of arrhythmia by using deep learning with 2-d ECG Spectral Image Representation

1) INTRODUCTION :

Cardiovascular diseases (CVDs) are the leading cause of human death, with over 17 million people known to lose their lives annually due to CVDs. A normal heartbeat varies with age, body size, activity, and emotions. In cases where the heartbeat feels too fast or slow, the condition is known as palpitations. An arrhythmia does not necessarily mean that the heart is beating too fast or slow, it indicates that the heart is following an irregular beating pattern. A classification model to identify CVDs at their early stage could effectively reduce the mortality rate by providing a timely treatment.

PROJECT OVERVIEW:

The electrocardiogram (ECG) is one of the most extensively employed signals used in the diagnosis and prediction of cardiovascular diseases (CVDs). The ECG signals can capture the heart's rhythmic irregularities, commonly known as arrhythmia. A careful study of ECG signals is crucial for precise diagnosis of patients' acute and chronic heart conditions.
In our proposed system, a first-time user should register in the webapp using his/her respective mobile number or email Id, by doing so they will receive a one-time password which will let the user to login and set a password for future uses.

The backend process consists of an algorithm responsible for classifying the arrhythmia and also predicting if the heart rhythm is irregular or not. We are using machine learning to facilitate the same. The algorithm employs a two-dimensional (2-D) convolutional neural network (CNN) model for the classification of ECG signals into eight classes; namely, normal beat, premature ventricular contraction beat, paced beat, right bundle branch block beat, left bundle branch block beat, atrial premature contraction beat, ventricular flutter wave beat, and ventricular escape beat. The one-dimensional ECG time series signals are transformed into 2-D spectrograms through short-time Fourier transform. The 2-D CNN model consisting of four convolutional layers and four pooling layers is designed for extracting robust features from the input spectrograms. The model has achieved 96.39% training accuracy and 97% testing accuracy.

PURPOSE :

Our proposed system has an upper hand over the existing systems as it is able to compute within the same time constraint for any range of dataset. The computational efficiency is not compromised for increasing dataset. This solely brings out our system as an efficient and reliable model to bank on.

2. LITERATURE SURVEY

1) Classification of ECG Heartbeat Arrhythmia

Published in: January 2020 by Jagadeeswararao Annam,
Srinivas Kalyanapu , Sureshbabu
https://www.researchgate.net/publication/341907602_Classification_of_ECG_Heartbeat_Arrhythmia_A_Review

Manual identification of ECG heart-beat classes by cardiologists is time consuming and cumbersome. These professionals rely on computer based methods for determination of these heart-disease types. In this work, existing literature is organized into a proposed taxonomy based on dichotomies involving full time series-based versus feature-based, AAMI versus Non-AAMI, and inter-patient versus intra-patient based distinctions. The basic contributions of this work are systematic review of literature on heart-beat abnormality detection, identifying research gaps and the research issues unmet sofar in the literature to propose novel approaches for addressing these gaps.

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

Published in: April 2020 by Amin Ullah, S.Anwar
https://www.researchgate.net/publication/340817081_Classification_of_Arrhythmia_by_Using_Deep_Learning_with_2-D_ECG_Spectral_Image_Representation
The electrocardiogram (ECG) is one of the most extensively employed signals used in the diagnosis and prediction of cardiovascular diseases (CVDs). The ECG signals can capture the heart's rhythmic irregularities, commonly known as arrhythmias. A careful study of ECG signals is crucial for precise diagnoses of patients' acute and chronic heart conditions. In this study, we propose a two-dimensional (2-D) convolutional neural network (CNN) model for the classification of ECG signals into eight classes; namely, normal beat, premature ventricular contraction beat, paced beat, right bundle branch block beat, left bundle branch block beat, atrial premature contraction beat, ventricular flutter wave beat, and ventricular escape beat. The one-dimensional ECG time series signals are transformed into 2-D spectrograms through short-time Fourier transform. The 2-D CNN model consisting of four convolutional layers and four pooling layers is designed for extracting robust features from the input spectrograms. Our proposed methodology is evaluated on a publicly available MIT-BIH arrhythmia dataset. We achieved a state-of-the-art average classification accuracy of 99.11%, which is better than those of recently reported results in classifying similar types of arrhythmias. The performance is significant in other indices as well, including sensitivity and specificity, which indicates the success of the proposed method.

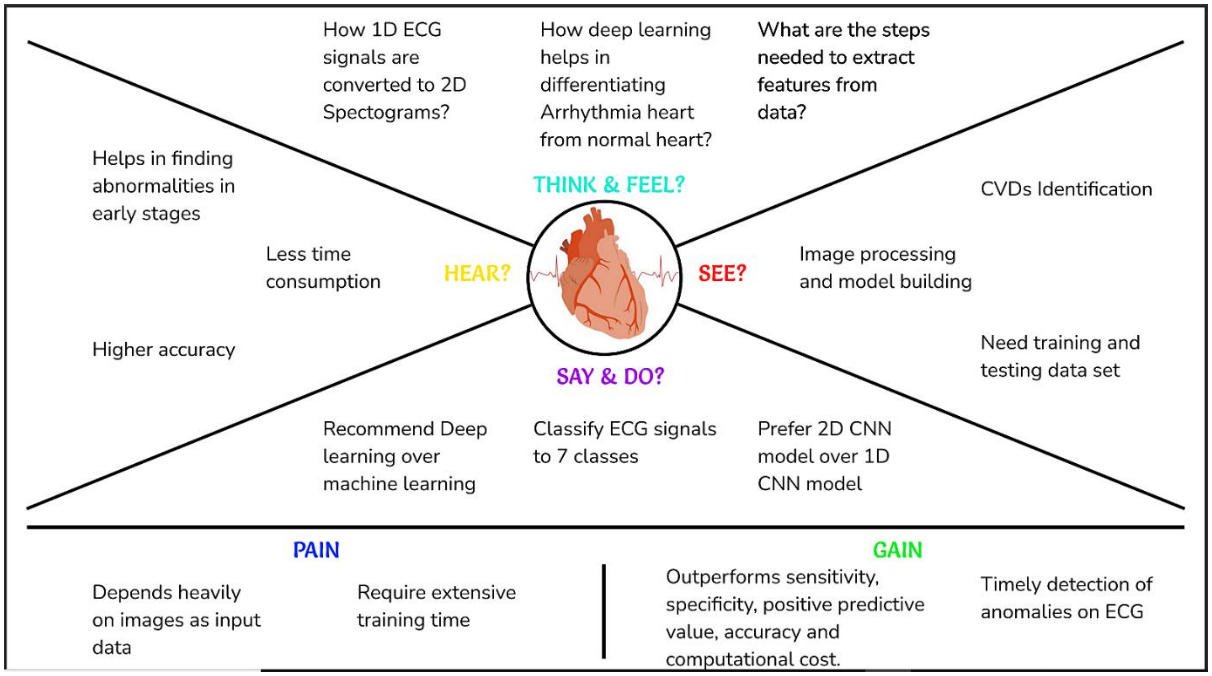
3) Classification of cardiac arrhythmia using a convolutional neural network and bi-directional long short-term memory

Published in: May 26, 2022 by Shahab Ul Hassan, Mohd S Mohd Zahid , Khaleel Husain
https://journals.sagepub.com/doi/full/10.1177/20552076221102766

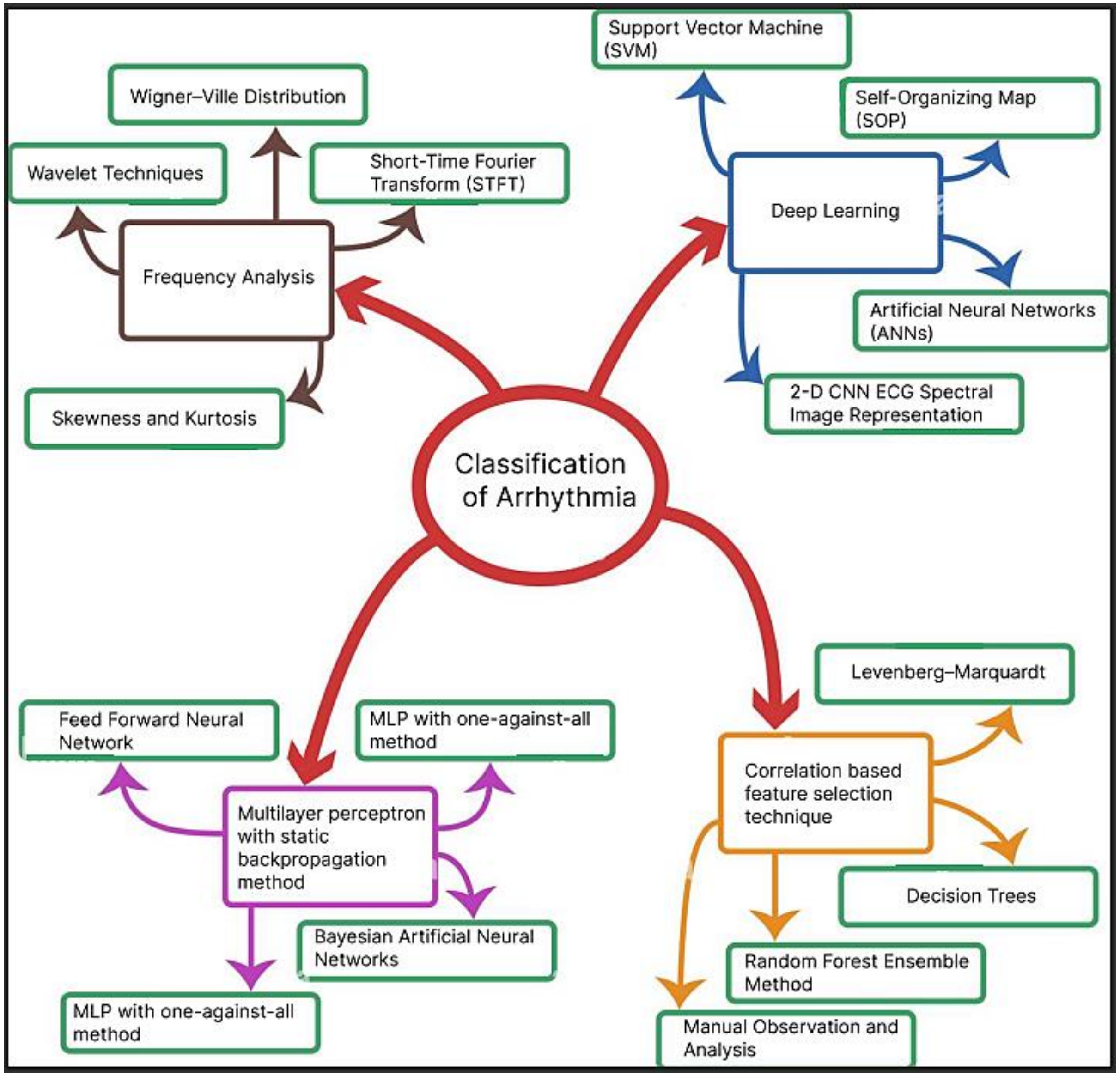
Cardiac arrhythmia is a leading cause of cardiovascular disease, with a high fatality rate worldwide. The timely diagnosis of cardiac arrhythmias, determined by irregular and fast heart rate, may help lower the risk of strokes. Electrocardiogram signals have been widely used to identify arrhythmias due to their non-invasive approach. However, the manual process is error-prone and time-consuming. A better alternative is to utilize deep learning models for early automatic identification of cardiac arrhythmia, thereby enhancing diagnosis and treatment. In this article, a novel deep learning model, combining convolutional neural network and bi-directional long short-term memory, is proposed for arrhythmia classification. Specifically, the classification comprises five different classes: non-ectopic (N), supraventricular ectopic (S), ventricular ectopic (V), fusion (F), and unknown (Q) beats. The proposed model is trained, validated, and tested using MIT-BIH and St-Petersburg data sets separately. Also, the performance was measured in terms of precision, accuracy, recall, specificity, and f1-score. The results show that the proposed model achieves training, validation, and testing accuracies of 100%, 98%, and 98%, respectively with the MIT-BIH data set. Lower accuracies were shown for the St-Petersburg data set. The performance of the proposed model based on the MIT-BIH data set is also compared with the performance of existing models based on the MIT-BIH data set.

3. IDEATION PHASE

Empathy Map Canvas



IDEATION AND BRAINSTORMING


Support Vector Machine (SVM)
Wigner–Ville Distribution
Self-Organizing Map (SOP)
Wavelet Techniques
Short-Time Fourier Transform (STFT)
Deep Learning
Frequency Analysis
Artificial Neural Networks (ANNs)
2-D CNN ECG Spectral Image Representation
Skewness and Kurtosis
Classification of Arrhythmia
Levenberg–Marquardt
Feed Forward Neural Network
MLP with one-against-all method
Correlation based feature selection technique
Multilayer perceptron with static backpropagation method
Decision Trees
Bayesian Artificial Neural Networks
Random Forest Ensemble Method
MLP with one-against-all method
Manual Observation and Analysis

Proposed Solution Template:

S.No.	**Parameter**	**Description**
1.	Problem Statement	Classification of Arrhythmia by Using Deep Learning with 2-D ECG Spectral Image Representation.
2.	Idea / Solution description	Spectrograms are employed generated from 1-D ECG signals using STFT. In addition, data augmentation and CNN-based methods are used with 2-D spectrograms as input.
3.	Novelty / Uniqueness	Can achieve better accuracy and efficiency even with large datasets.
4.	Social Impact / Customer Satisfaction	Reduced hardware complexity and less time consumption.
5.	Business Model (Revenue Model)	Capital equipment revenue model.
6.	Scalability of the Solution	Since we use improvised algorithms for training and processing large datasets,it performs computations in a cost-effective and time-saving way.

Project Design Phase-I - Solution Fit Template

Project Title: Classification of Arrhythmia by Using Deep Learning with 2-D ECG Spectral Image Representation
Team ID: PNT2022TMID00351

1. CUSTOMER SEGMENT(S) CS

Who is your customer?

Our potential customers include Doctors,radiologists and nurses.

6. CUSTOMER CONSTRAINTS CC

What constraints prevent your customers from taking action or limit their choices of solutions?

Hardware complexity, Budget, Time consuming process in training the model are the constraints that prevent customers from making actions.

5. AVAILABLE SOLUTIONS AS

Which solutions are available to the customers when they face the problem or need to get the job

Artificial neural networks (ANNs) :

Pros: Good fault tolerance and ability to parallel processing

Cons: Require lots of computational power and lesser accuracy of 90.6%

Support vector machines (SVMs) :

Pros: Best algorithm when classes are separable and suitable for extreme case binary classification

Cons: Training time is higher and does not perform well when target classes re overlapping

2. JOBS-TO-BE-DONE / PROBLEMS J&P

Which jobs-to-be-done (or problems) do you address for your customers?

To deal with patients suffering from Cardiovascular diseases (CVDs) such as Tachycardia, Bradycardia , Diabetes, High BP, Hyperthyroidism etc.

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?

To improve accuracy. Feed-forward neural network has an accuracy of 96.95%, 1-D CNN model has an average accuracy of 97.03% but our proposed model has an accuracy of 98.06%.

7. BEHAVIOUR BE

What does your customer do to address the problem and get the job done?

Our customer opt for efficient algorithm to address the problem and get the job done.

3. TRIGGERS TR

What triggers customers to act?

Prompts other doctors to use by seeing their fellow doctors using it in their hospitals.

4. EMOTIONS: BEFORE / AFTER EM

How do customers feel when they face a problem or a job and afterwards?

The customers feel unreliable and time consuming which is overcome in our model which makes them feel reliable and satisfied thereby saving their time

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 behavior.

1. Spectrograms (2-D images) are employed, which are generated from the 1-D ECG signal using STFT. In addition, data augmentation was used for the 2-D image representation of ECG signals.
2. A state-of-the-art performance was achieved in ECG arrhythmia classification by using the proposed CNN-based method with 2-D spectrograms as input.

Our solution is to perform fine tuning with large database to achieve higher accuracy and robustness

8. CHANNELS of BEHAVIOUR CH

8.1 ONLINE

What kind of actions do customers take online?

Mobile app and website.

8.2 OFFLINE

What kind of actions do customers take offline and use them for customer development?

Newspaper advertisement .

4. REQUIREMENT ANALYSIS

Functional Requirements:

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Gmail. Registration through Phone number.
FR-2	User Confirmation	Confirmation via Email Confirmation via OTP
FR-3	User Login	Login using password created during registration.
FR-4	Uploading ECG data	Registered user should be able to upload ECG reports on the application for analysis
FR-5	Sensing of abnormality	The trained ECG model should be able to identify arrhythmic heart beats and alert users in case of any abnormalities.
FR-6	Informing the type of arrhythmia	Further the model should accurately classify the kind of arrhythmia inorder to carry out the appropriate treatment rapidly .

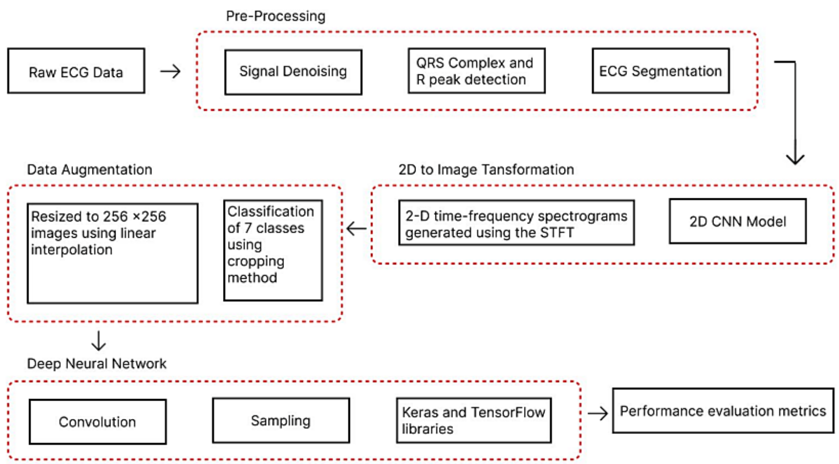
Non-functional Requirements:

FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The proposed model is a web application that can be used on a computer or mobile phones with internet connectivity.
NFR-2	**Security**	Unique account creation using phone number or Gmail and verification of password.
NFR-3	**Reliability**	Since it uses an optimised algorithm, it is capable of processing large datasets as well as able to handle several user requests simultaneously.
NFR-4	**Performance**	It achieves 99.61% specificity, 99.11% average accuracy and 98.59% positive predictive value (precision)
NFR-5	**Availability**	Since the 1-D ECG data is converted to 2-D spectral images,it increases the versatility of the model making it's processing fast and available to a number of users at a time.
NFR-6	**Scalability**	The experimental setup consists of an eighth-generation ASUS server with 32GB internal RAM, 500 GB external SSD hard drive with the addition of internal hard drive, and NVIDIA 1080

		GPU with 11 GB memory.Thus with the help of these hardwares, we achieve enhanced scalability.

5. PROJECT DESIGN

Data Flow Diagram:



Technical Architecture:

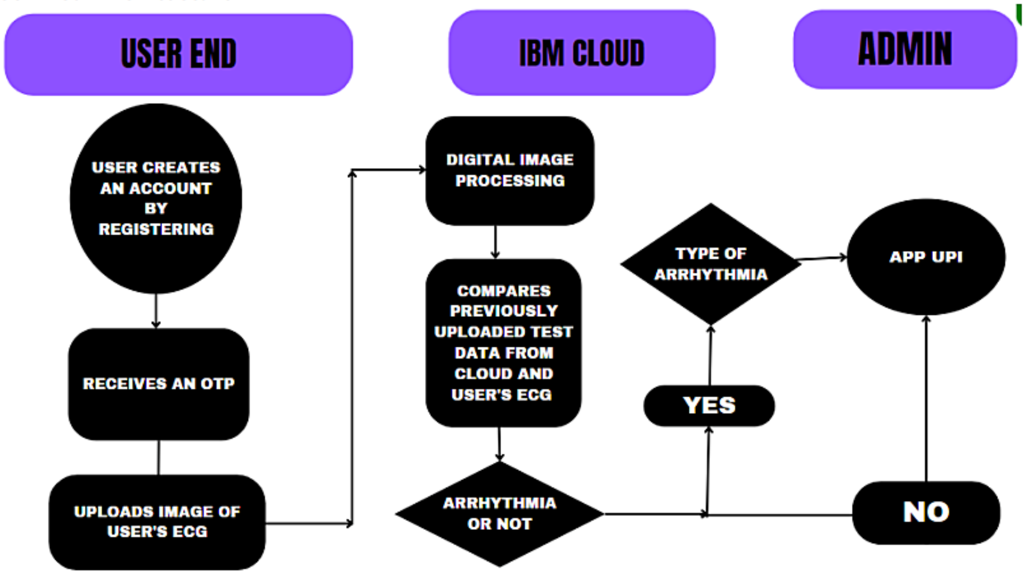


Table-1 : Components & Technologies:

S.No	Component	Description	Technology
1.	User Interface	Web App	HTML, CSS, JavaScript
2.	Application Logic-1	Logic for a process in the application	Python
3.	Application Logic-2	Logic for a process in the application	IBM Watson STT service
4.	Application Logic-3	Logic for a process in the application	IBM Watson Assistant
5.	Database	Data Type, Configurations etc.	MySQL
6.	Cloud Database	Database Service on Cloud	IBM DB2, IBM Cloudant
7.	File Storage	File storage requirements	IBM Block Storage
8.	External API-1	Purpose of External API used in the application	Contact information API
9.	External API-2	Purpose of External API used in the application	Image recognition API
10.	Machine Learning Model	Purpose of Machine Learning Model	Electrocardiogram pattern recognition model
11.	Infrastructure (Server / Cloud)	Application Deployment on Local System / Cloud Local Server Configuration: Cloud Server Configuration :	Cloud Foundry, Kubernetes

Table-2: Application Characteristics:

S.No	Characteristics	Description	Technology
1.	Open-Source Frameworks	List the open-source frameworks used	Kubernetes, Anaconda
2.	Security Implementations	List all the security / access controls implemented, use of firewalls etc.	Google cloud vision API
3.	Scalable Architecture	Justify the scalability of architecture (3 – tier, Micro-services)	Technology used

S.No	Characteristics	Description	Technology
4.	Availability	Justify the availability of application (e.g. use of load balancers, distributed servers etc.)	Technology used
5.	Performance	Design consideration for the performance of the application (number of requests per sec, use of Cache, use of CDN's) etc.	Technology used

User Stories:

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority
Customer (Web user)	Registration	USN-1	As a user, I can register for the application by entering my email, password, and confirming my password.	I can access my account / dashboard.	High
		USN-2	As a user, I will receive confirmation email once I have registered for the application.	I can receive confirmation email & click confirm.	High
		USN-3	As a user, I can register for the application through Gmail or through Phone number.	I can receive OTP/confirmation email & click confirm.	Medium
	Login	USN-4	As a user, I can log into the application by entering email & password.	I can log in to Dashboard.	High
	Dashboard	USN-5	As a user, I can view the reports of my ECG.	I can keep a track of my reports.	High
Customer Care Executive	Registration	USN-6	As a Customer Care Executive, I can register for the application by entering my email, password, and confirming my password.	I can access users account / dashboard. Also I can receive confirmation email & click confirm.	High
	Login	USN-7	As a Customer Care Executive, I can log into the application by entering email & password.	I can access to the queries raised by users and look out for the underlying issues.	High
	Query resolving	USN-8	As a Customer Care Executive, I need to solve user queries regarding the reports.	I can send a mail with detailed explanation.	High
Administrator	Add / remove user	USN-9	As an admin, I can add or remove user details manually.	I can remove or delete user reports under valid reasons.	High
	Data management	USN-10	As an admin, I need to assure the evaluations are precise and data are being managed accordingly.	I can update the features of the web page.	Medium

6)PROJECT PLANNING AND SOLUTIONS:

S.NO	MILESTONE	ACTIVITIES	DATE
1.	Preparation Phase	Pre-requisites	22 Aug 2022
		Prior knowledge	22 Aug 2022
		Project Structure	22 Aug 2022
		Project Flow	21 Aug 2022

		Project Objectives	20 Aug 2022
		Environment Set-up	25 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 Sept 2022
		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.	Project Planning Phase	Milestones & Tasks	17 Oct 2022 - 18 Oct 2022
		Sprint Schedules	19 Oct 2022 - 22 Oct 2022
6.	Project Development Phase	Sprint-1	24 Oct 2022 - 28 Oct 2022
		Sprint-2	30 Oct 2022 - 04 Nov 2022
		Sprint-3	06 Nov 2022- 10 Nov 2022
		Sprint-4	13 Nov 2022 - 19 Nov 2022
	Sprint-1	• Download The Dataset • Import ImageDataGenerator Library • Configure ImageDataGenerator class • Import Libraries • Initialize the Model	24 Oct 2022 – 28 Oct 2022

	Sprint – 2	• Register IBM Cloud • Apply ImageDataGenerator functionality to Trainset and Dataset • Test the model	30 Oct 2022 – 04 Nov 2022
	Sprint – 3	• Train the model on IBM • Create Html files • Train the Model	05 Nov 2022 – 10 Nov 2022
	Sprint – 4	• Configure The Learning Process • Build Python code • Adding Dense Layer • Adding CNN layer	13 Nov 2022 – 19 Nov 2022

7)**CODING AND SOLUTIONS:**

Feature 1:

CONVERTING IMAGE TO ARRAY:

```
import requests
from tensorflow.keras.preprocessing import image
from tensorflow.keras.models import load_model
import numpy as np
import pandas as pd
import tensorflow as tf
import os
from werkzeug.utils import secure_filename
from tensorflow.python.keras.backend import set_session
from flask import Flask, render_template, redirect, url_for, request
```

In [2]:

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

In [3]:

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

In [4]:

```
@app.route('/about')
def home():
    return render_template('about.html')
@app.route('/info')
def information():
    return render_template('info.html')
```

In [6]:

```
@app.route('/index')
def test():
    return render_template('index.html')
@app.route("/predict", methods=["GET", "POST"]) #route for our prediction
def upload():
    if request.method== 'POST':
        f=request.files['image'] #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_classes(x) #predicting classes
        print("prediction", 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[pred[0]])
        return result
```

FEATURE 2:
CLASSIFICATION OF ARRYTHMIA:

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

In [2]:

```
train_datagen = ImageDataGenerator (rescale = 1./255, shear_range= 0.2,zoom_range= 0.2, horizontal_flip = True)
```

In [3]:

```
test_datagen =ImageDataGenerator (rescale = 1./255)
```

In [4]:

```
x_train = train_datagen.flow_from_directory(r'C:\Users\PC\Desktop\IBM\data\train',target_size = (64,64), batch_size = 32, class_mode = 'categorical')
x_test = test_datagen.flow_from_directory(r'C:\Users\PC\Desktop\IBM\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.
```

In [5]:

```
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
model=Sequential()
```

In [7]:

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

In [8]:

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

In [9]:

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

In [10]:

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

In [11]:

```
model.add(Flatten())
```

In [12]:

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

In [13]:

```
model.summary()
```

FEATURES 3:

PRINTING THE PREDICTION:

```
import
requests
        from tensorflow.keras.preprocessing import image
        from tensorflow.keras.models import load_model
        import numpy as np
        import pandas as pd
        import tensorflow as tf
        import os
        from werkzeug.utils import secure_filename
        from tensorflow.python.keras.backend import set_session
        from flask import Flask, render_template, redirect, url_for, request


        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('/index')
        def test():
            return render_template('index.html')

        @app.route("/predict", methods=["GET", "POST"]) #route for our prediction
        def upload():
            if request.method== 'POST':
                f=request.files['image'] #requesting the file
                basepath=os.path.dirname(__file__)#storing the file directory
```

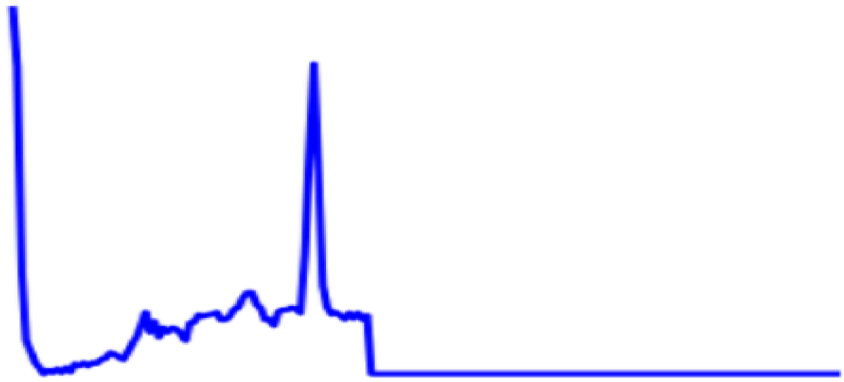
```
        filepath=os.path.join(basepath, 'uploads',f.filename) #storing the
file in uploads folder
        f.save(filepath) #saving the file

        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_classes(x) #predicting classes
        print("prediction", 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[pred[0]])
        return result

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

8) TESTING

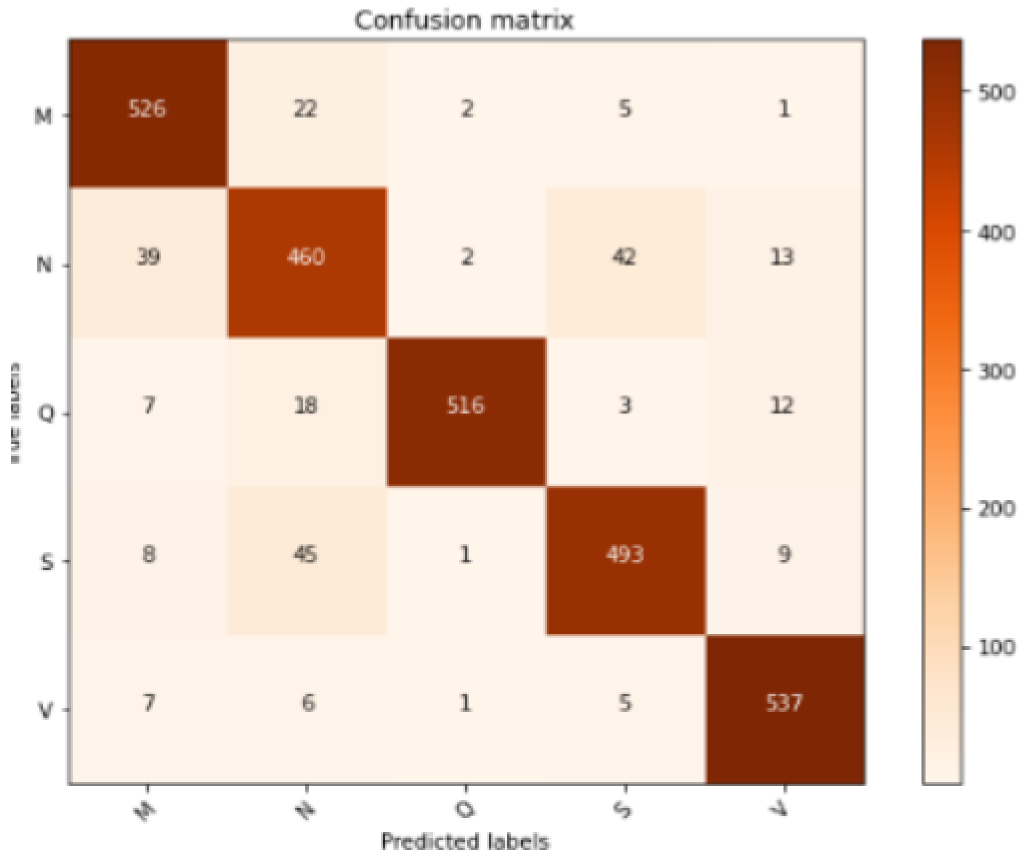
The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with 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 oftest. Each test type addresses a specific testing requirement.

9) RESULTS:

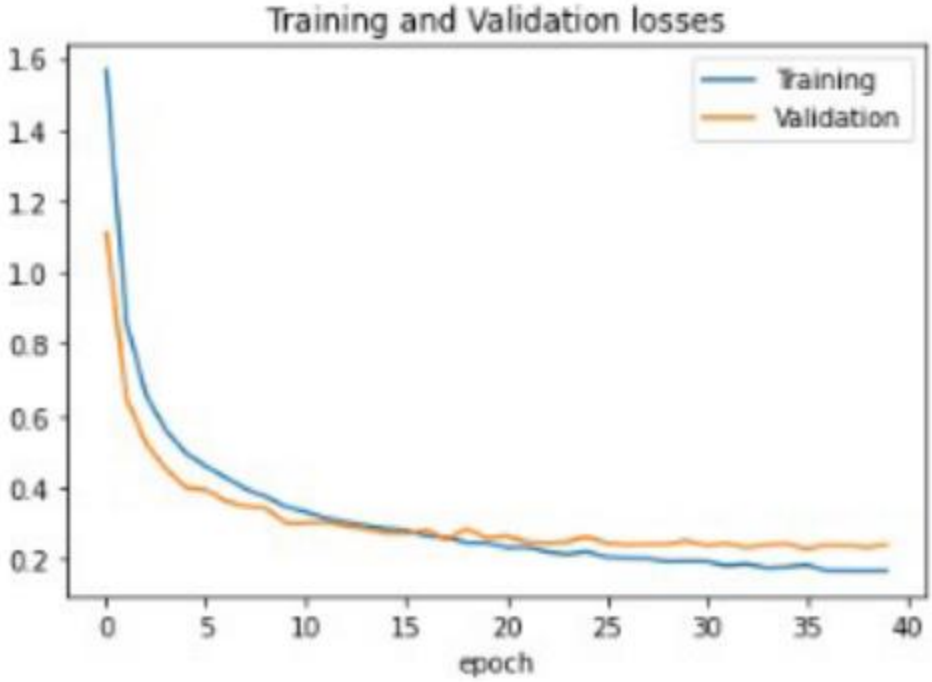
1) **TESTING IMAGE**



2) **CONFUSION MATRIX**

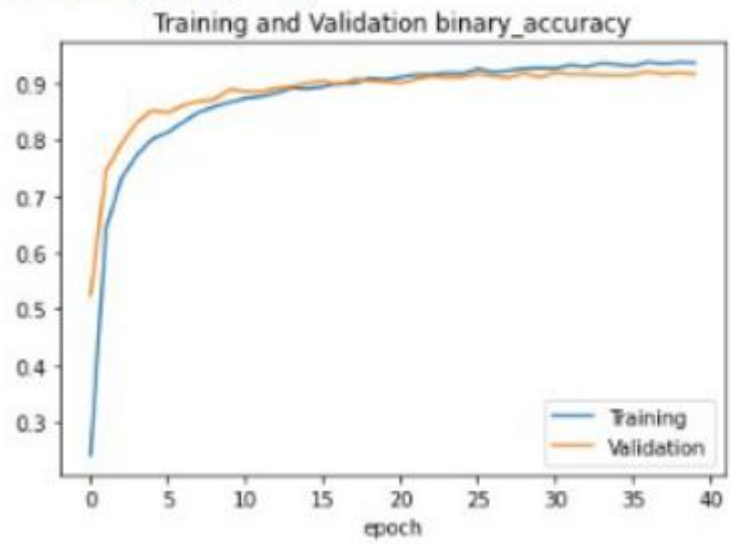


3) **TRAINING AND VALIDATION LOSSES**



4) **TESTING ACCURACY**

Text(0.5, 0, 'epoch')



9)RESULT:

It stood out as a solution which works efficiently and flawlessly when we increase our dataset, unlike the other models which suffer during the increasing the dataset phase which can make the system unreliable.

10)ADVANTAGES AND DISADVANTAGES:

The accuracy can reach upto 98.00%. The 2d cnn has the biggest pro as its ability to extract features automatically without any human intervention. Also, it has higher accuracy and can do fine tuning. It has high robustness as well. The disadvantage lies in the cost and also the data storage and security problems which can be under control if treated properly and securely.

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%.

12) FUTURE SCOPE :

This can be a great source of alternative for the existing systems and give results more precisely if incorporated in the actual medical field without any doubt. It has high accuracy and can work perfectly fine when the dataset becomes large which is the need of the future generations to come.

13)APPENDIX:

https://github.com/IBM-EPBL/IBM-Project-12438-1659451313-GITHUB LINK

https://drive.google.com/drive/folders/13DkcgXF6K0i1j4bP_kOn74zPHwgSZe3X?usp=sharing - VIDEO LINK