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

Team ID:PNT2022TMID16980

INDEX

1. INTRODUCTION

1.1 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 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.

1.2 Purpose

The electrocardiogram (ECG) recordings are widely used for diagnosing and predicting cardiac arrhythmia for diagnosing heart diseases. Towards this end, clinical experts might need to look at ECG recordings over a longer period of time for detecting cardiac arrhythmia. The ECG is a one-dimensional (1-D) signal representing a time series, which can be analyzed using machine learning techniques for automated detection of certain abnormalities. Recently, deep learning techniques have been developed, which provide significant performance in radiological image analysis [4,5]. Convolutional neural networks (CNNs) have recently been shown to work for multi-dimensional (1-D, 2-D, and in certain cases, 3-D) inputs but were initially developed for problems dealing with images represented as two-dimensional inputs [6]. For time series data, 1-D CNNs are proposed but

are less versatile when compared to 2-D CNNs. Hence, representing the time series data in a 2-D format could benefit certain machine learning tasks [7,8]. Hence, for ECG signals, a 2-D transformation has to be applied to make the time series suitable for deep learning methods that require 2-D images as input.

2. LITERATURE SURVEY

2.1 Existing problem

The ECG records the electrical signals of the human heart and is mostly used for clinical diagnosis of cardiac arrhythmias. More than 300 million ECGs are obtained worldwide

every year. The huge diagnostic workload leads to inefficiency and misdiagnosis of cardiac arrhythmias based on ECG. So the combination of extensive digitization of ECG data and automatic classification algorithms has attracted more and more attention. In the early research on the automatic classification of cardiac arrhythmia, most algorithms based on machine learning are usually divided into two parts: feature engineering and classification. Specifically, researchers first manually extracted a large number of ECG features with medical meaning, such as wavelet features, composite features, heart rate variability statistical feature, related statistical features, higher order statistical features and morphological features. Meanwhile, the principal component analysis and independent component analysis use mathematical methods to extract ECG features from high-dimensional space to low-dimensional space. After feature engineering, support vector machine, self-organizing map, clustering and other machine learning algorithms are used to analyze artificial features and give the prediction result. Although machine learning has broad research applications in the classification of cardiac arrhythmia, there are still some problems that need to be solved. For example, feature engineering based on subjective factors leads to the elimination of some potentially important features, which may affect the final classification performance.

2.2 References

1. Mc Namara, K.; Alzubaidi, H.; Jackson, J.K. Cardiovascular disease as a leading cause of death: How are pharmacists getting involved? Integr. Pharm. Res. Pract. 2019, 8, 1. [CrossRef] [PubMed]

2. Lackland, D.T.; Weber, S.M.A. Global burden of cardiovascular disease and stroke: hypertension at the core. Can. J. Cardiol. 2015, 31, 569–571. [CrossRef] [PubMed]

3. Mustaqeem, A.; Anwar, S.M.; Majid, M. A modular cluster based collaborative recommender system for cardiac patients. Artif. Intell. Med. 2020, 102, 101761. [CrossRef] [PubMed]

4. Irmakci, I.; Anwar, S.M.; Torigian, D.A.; Bagci, U. Deep Learning for Musculoskeletal Image Analysis. arXiv 2020, arXiv:2003.00541.

5. Anwar, S.M.; Majid, M.; Qayyum, A.; Awais, M.; Alnowami, M.; Khan, M.K. Medical image analysis using convolutional neural networks: A review. J. Med. Syst. 2018, 42, 226. [CrossRef]

6. Gu, J.; Wang, Z.; Kuen, J.; Ma, L.; Shahroudy, A.; Shuai, B.; Liu, T.; Wang, X.; Wang, G.; Cai, J.; et al. Recent advances in convolutional neural networks. Pattern Recognit. 2018, 77, 354–377. [CrossRef]

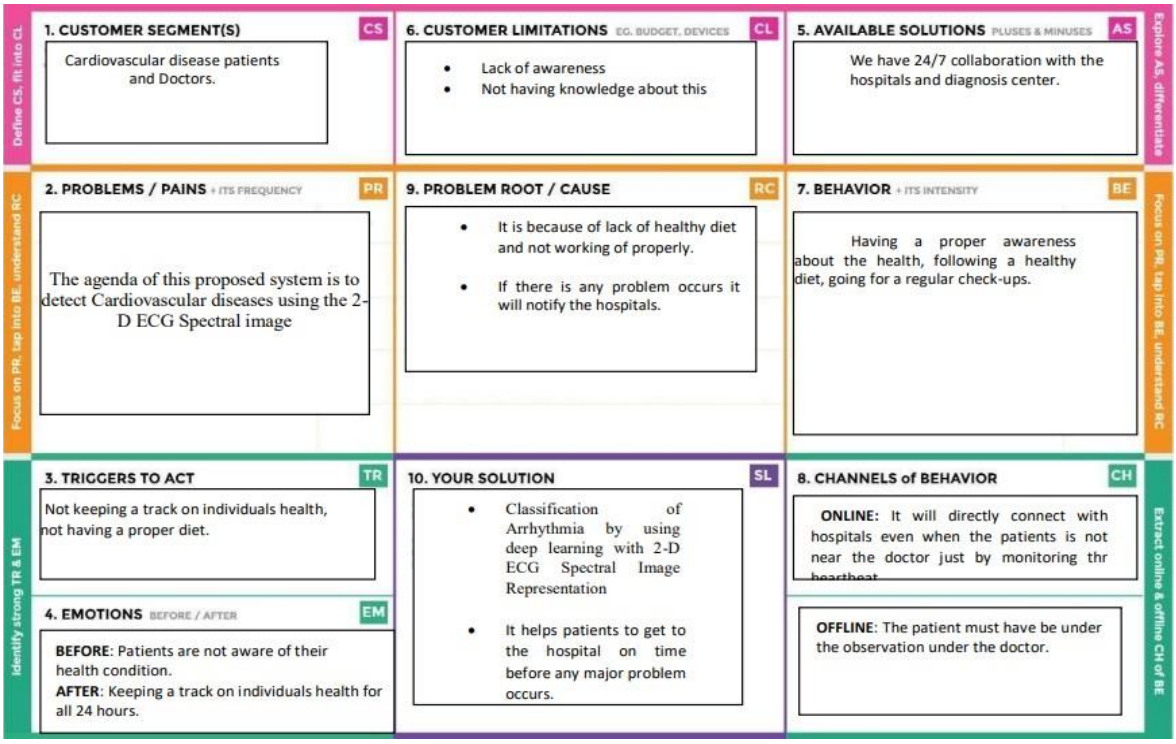
7. Wu, Y.; Yang, F.; Liu, Y.; Zha, X.; Yuan, S. A comparison of 1-D and 2-D deep convolutional neural networks in ECG classification. arXiv 2018, arXiv:1810.07088.

8. Zhao, J.; Mao, X.; Chen, L. Speech emotion recognition using deep 1D & 2-D CNN LSTM networks. Biomed. Signal Process. Control 2019, 47, 312–323.

9. Ortega, S.; Fabelo, H.; Iakovidis, D.K.; Koulaouzidis, A.; Callico, G.M. Use of hyperspectral/multispectral imaging in gastroenterology. Shedding some–different–light into the dark. J. Clin. Med. 2019, 8, 36. [CrossRef]

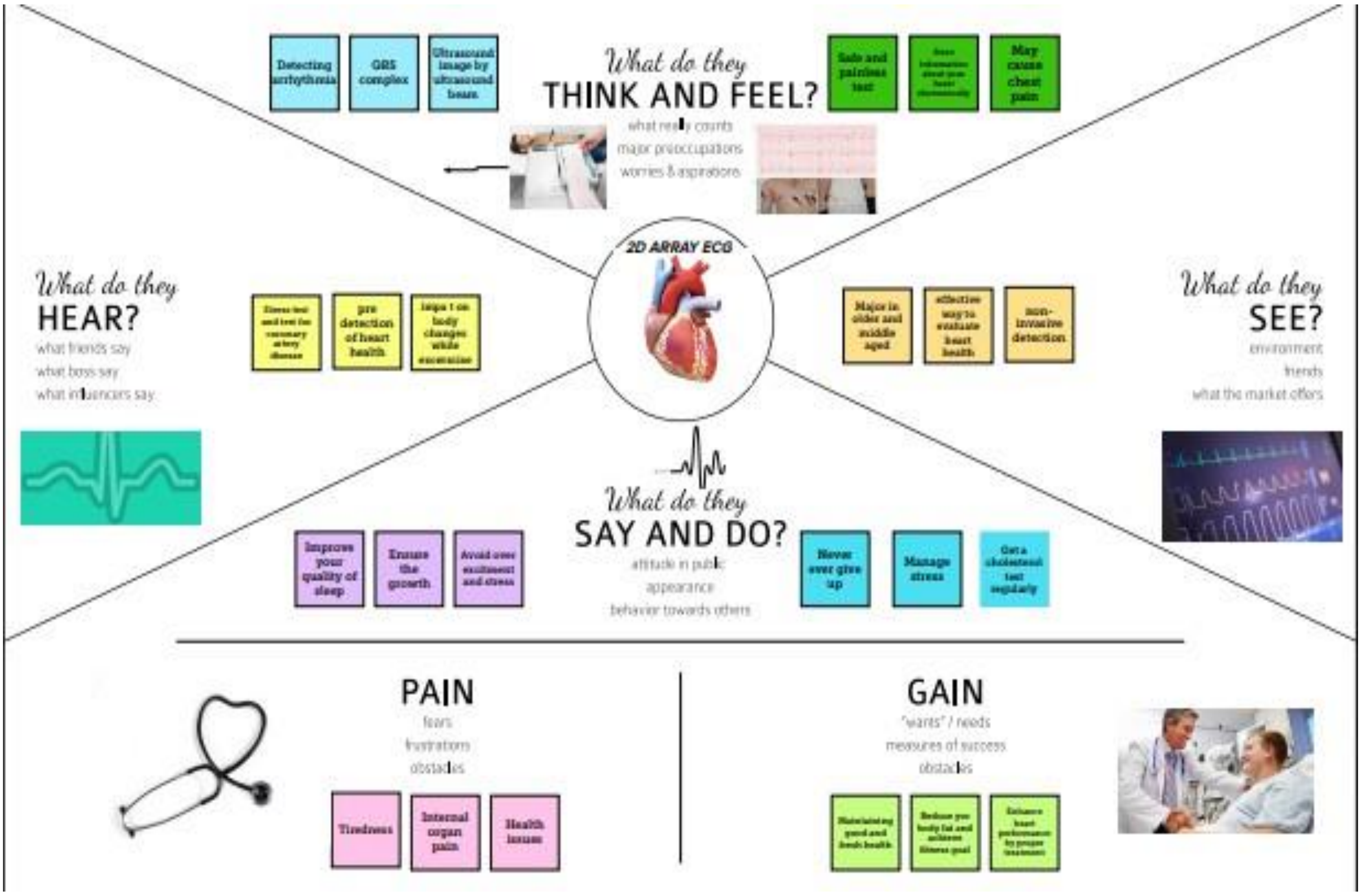
10. Feng, Y.-Z.; Sun, D.-W. Application of Hyperspectral Imaging in Food Safety Inspection and Control: A Review. Crit. Rev. Food Sci. Nutr. 2012, 52, 1039–1058. [CrossRef]

2.3 Problem Statement Definition



3. IDEATION & PROPOSED SOLUTION

3.1 Empathy Map Canvas

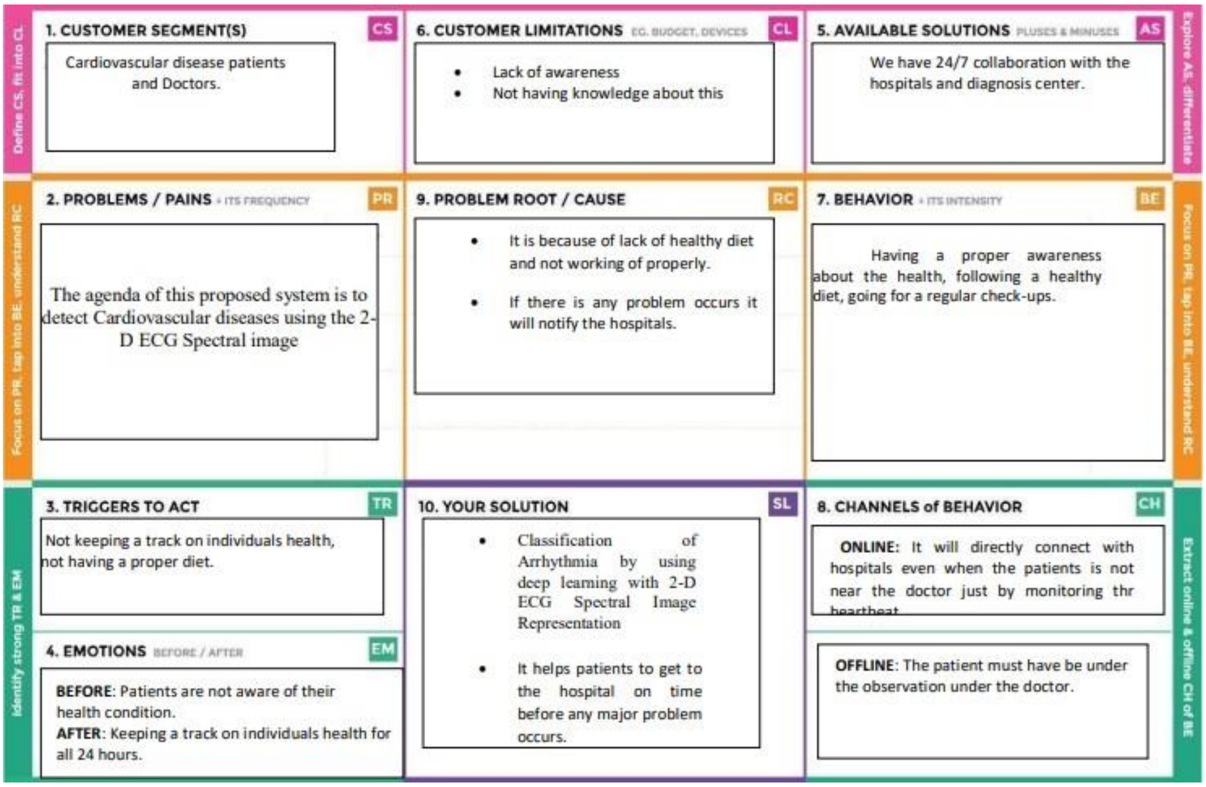


3.2 Ideation & Brainstorming

3.3 Proposed Solution

S.NO	PARAMETER	DESCRIPTION
1	Problem Statement	1. The developing application must be very efficient and useful for the user friendly. 2. The agenda of this proposed system is to detect Cardiovascular diseases using the 2-D ECG Spectral image.
2	Idea/solution description	1. This will caution them about the irregular pattern of their heartbeat(Arrhythmia). 2. We are proposing that the automated detection of such pattern to clinical consultation.
3	Novelty/uniqueness	1. Spectrograms(2-D images) are employed which are generated by the 1-D ECG signal using STFT. In addition, data augmentation was used for the 2-D image representation of ECG signals. 2. The method consists of five steps signal processing, generation of spectrograms, augmentation of data, extraction features from the data(using CNN model), classification based on features.
4	Social Impact/Customer Satisfaction	1. The main purpose of this application is to make people awareness on their general health. 2. Can collaborate with doctors and hospitals.
5	Business model	1. By approaching the government to organize awareness camps. 2. By collaborating with diagnosing centers.
6	Scalability of the solution	1. It is very essential for everyone to keep a track on individuals health. 2. It helps in monitoring one's health.

3.4 Problem Solution Fit



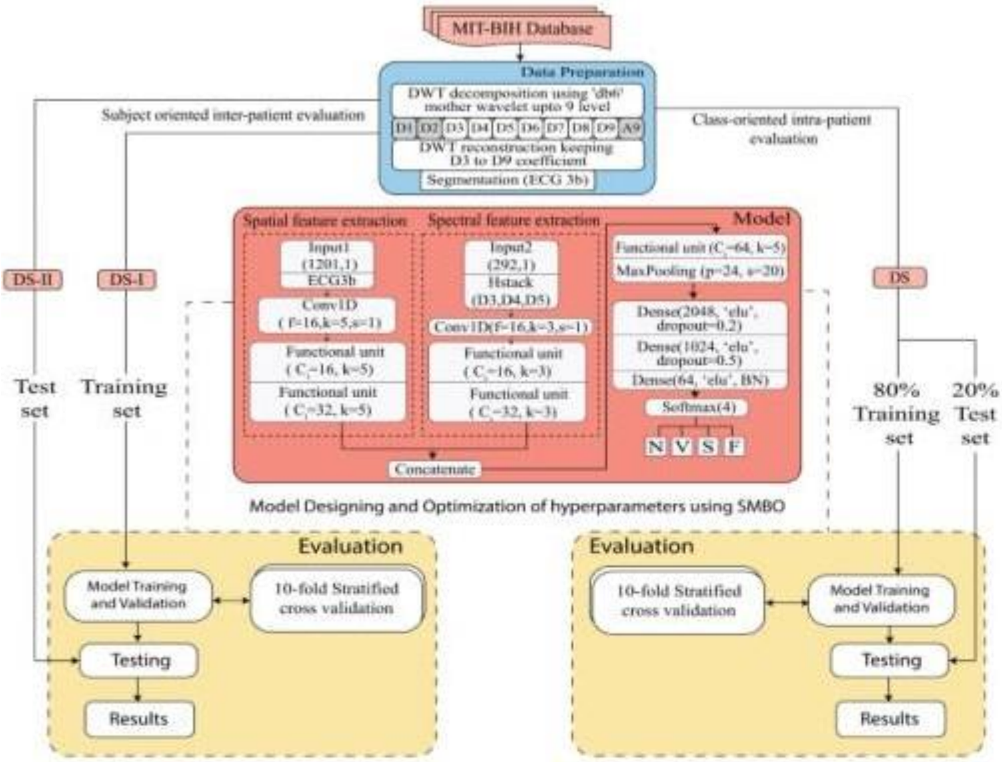
4. REQUIREMENT ANALYSIS

4.1 Functional Requirement

FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Download the app Registration through Gmail Create an account Follow the instructions Connect the app to the hospital
FR-2	User Confirmation	Confirmation through Email Monitoring starts after confirmation
FR-3	Interface	Good interface for the user to operate
FR-4	Accessing datasets	Health history of the patient is noted Personal details about the patient is recorded Information about the hospital is registered
FR-5	Mobile application	AI, Heart beat sensor and Blood pressure sensor can be accessed by the hospital through this mobile Application.

5. PROJECT FLOW

5.1 Data Flow Diagrams



5.2 Solution and technical architecture

Solution Architecture

Solution Architecture: Solution architecture is a complex process – with many sub-processes – that bridges the gap between business problems and technology solutions. Its goals are to:

➢ Having a proper awareness about the health, following a healthy diet, going for a regular check-ups

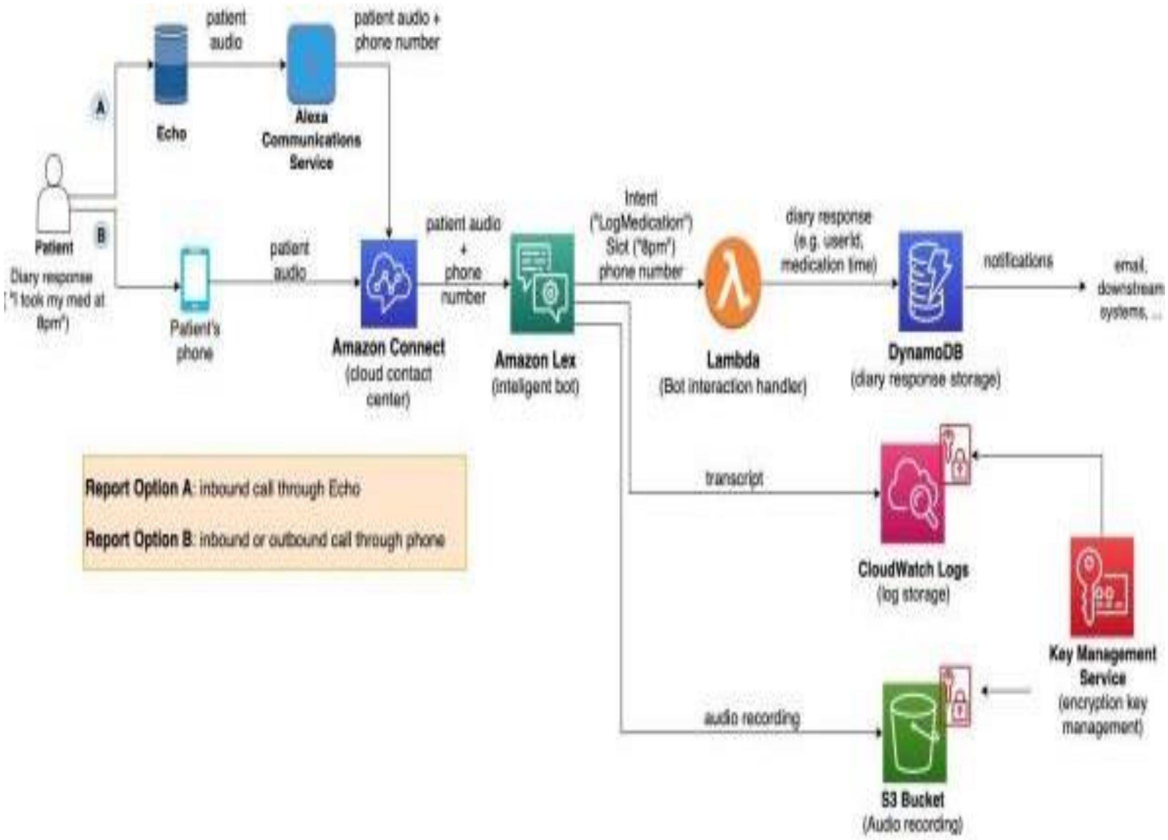
➢ It is because of lack of healthy diet and not working of properly.

➢ If there is any problem occurs it will notify the hospitals.

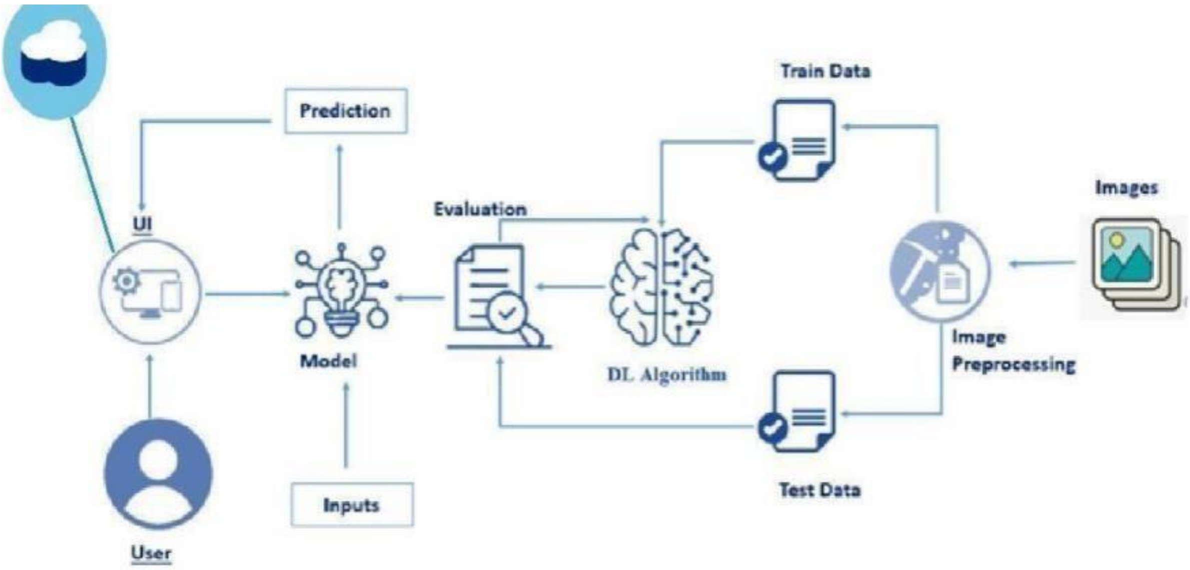
➢ Not keeping a track on individuals health, not having a proper diet.

➢ It will directly connect with hospitals even when the patients is not near the doctor just by monitoring the health.

Solution Architecture Diagram



Technical Architecture

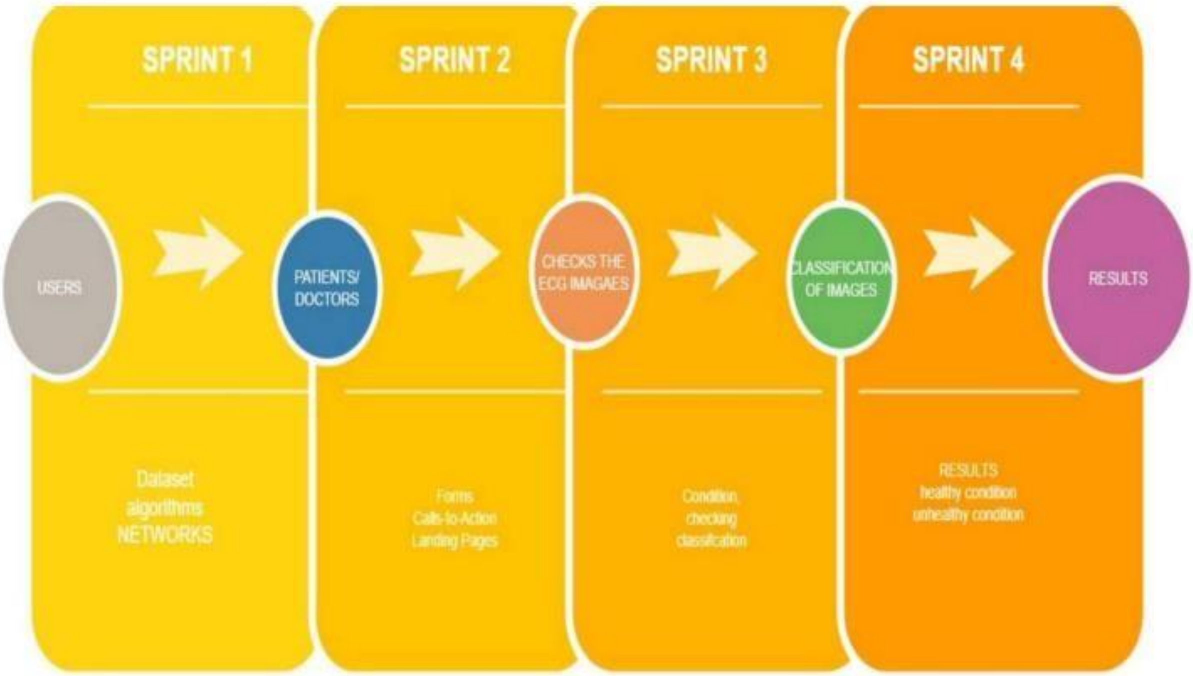


5.3 User stories

User Type	Functional Requirement (Epic)	User Story Number	User Story / Task	Acceptance criteria	Priority	Release
Customer (Mobile 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	
	Login	USN-2	As a user, I can log into the applicationby entering email & password	I can access theapplication	High	
	Dashboard	USN-3	As a user I can enter my income andexpenditure details.	I can view my dailyexpenses	High	
Customer Care Executive		USN-4	As a customer care executive, I can solvethe log in issues and other issuesof the application.	I can provide support or solution at any time 24*7	Medium	
Administrator	Application	USN-5	As an administrator I can upgradeor update the application.	I can fix the bug which arises for the customersand users of the application	Medium	

6. PROJECT PLANNING AND SCHEDULING

6.1 Sprint Delivery Schedule



7 SOLUTIONING

7.1Feature 1

The system will be able to predict the gesture such as which alphabet or number the person is trying to say. Various techniques like preprocessing, feature extraction are applied. CNN was used for classification. The web application is been developed using PHP and bootstrap for the Frontend and Python for Backend. The user captured image is passed and captured images features are extracted. Extracted features will be matched with the training model, depending on nearby match the predicted output is obtained.

7.2 Feature 2

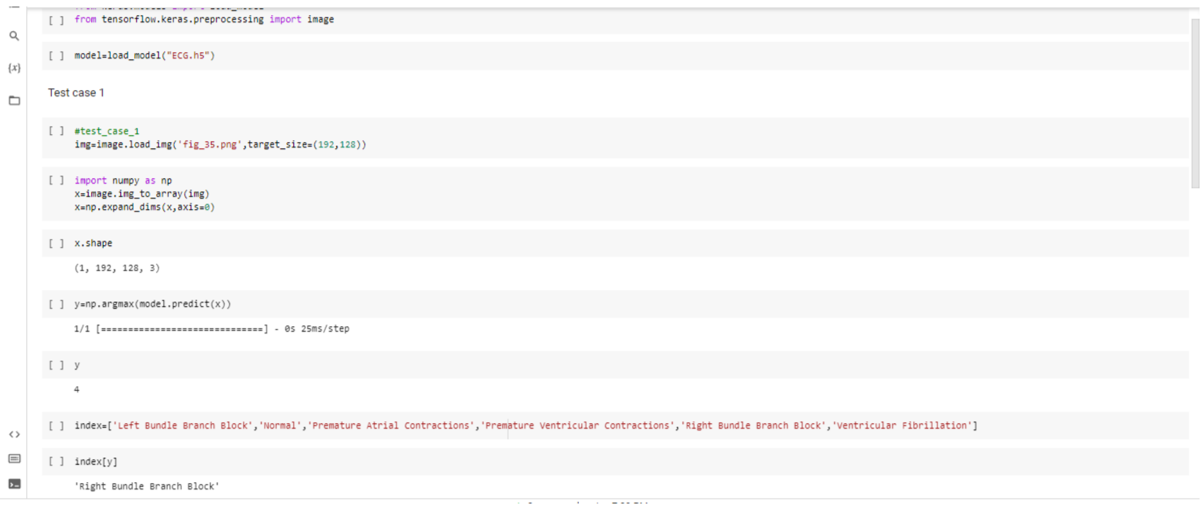
CNN's are used for image classification and recognition because of their high accuracy. The CNN follows a hierarchical model which works on building a network, like a funnel, and finally gives out a fully-connected layer where all the neurons are connected and the output is processed. Hence we are using a Convolutional Neural Network as a important feature.

8 TESTING

8.1Test cases

A test plan documents strategy that will be used to verify and ensure that a product or a system meets its design specification and other requirements. A test plan is usually prepared or by with significant input from the engineer. This document describes the plans for testing the architectural prototype of system. In my project the system has to be tested to get the desired output. I use different speed for testing the system

.



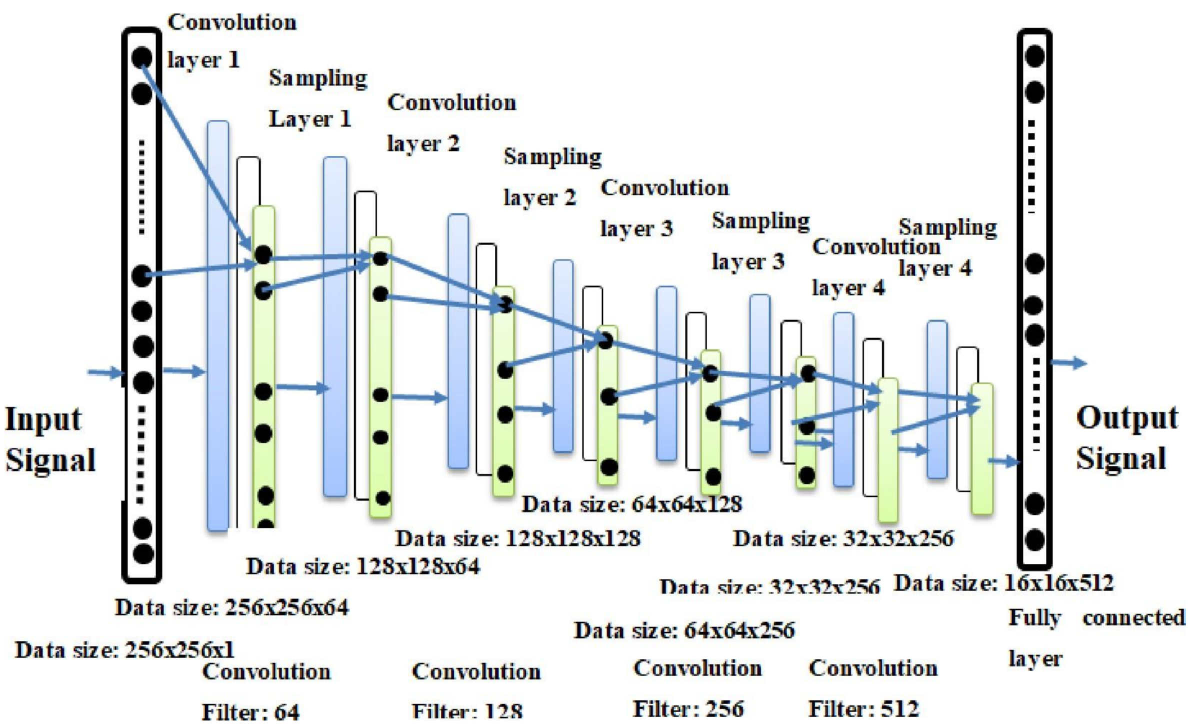
```
[ ] from tensorflow.keras.preprocessing import image

[ ] model=load_model("ECG.h5")
```

Test case 1

```
[ ] #test_case_1
    img=image.load_img('fig_35.png',target_size=(192,128))

[ ] import numpy as np
    x=image.img_to_array(img)
    x=np.expand_dims(x,axis=0)

[ ] x.shape
    (1, 192, 128, 3)

[ ] y=np.argmax(model.predict(x))
    1/1 [==============================] - 0s 25ms/step

[ ] y
    4

[ ] index=['Left Bundle Branch Block','Normal','Premature Atrial Contractions','Premature Ventricular Contractions','Right Bundle Branch Block','Ventricular Fibrillation']

[ ] index[y]
    'Right Bundle Branch Block'
```

8.2 User Acceptance Testing

User interface design(UI) or user interface engineering is the design of user interfaces for machines and software, such as computer, home appliances, mobile devices and other electronic devices, with the focus on maximizing usability and the user experience. The goal for user interface design is to make the user's interaction as simple and efficient as possible, in terms of accomplishing user goals(user centered design).



Test case 2

```
#test_case_2
```

```
img=image.load_img('fig_5749.png',target_size=(192,128))
```

```
import numpy as np
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
```

```
y=np.argmax(model.predict(x))
```

```
1/1 [==============================] - 0s 23ms/step
```

```
index[y]
```

```
'Premature Ventricular Contractions'
```

test case 3

```
#test_case_3
```

```
img=image.load_img('fig_3497.png',target_size=(192,128))
import numpy as np
x=image.img_to_array(img)
x=np.expand_dims(x,axis=0)
y=np.argmax(model.predict(x))

y
index[y]
```

9.RESULTS

9.1 Performance metrices

The two significant optimization parameters in the proposed 2-D CNN model are the learning rateand the batch size of the data used. To improve the performance, these two optimization parametersmust be selected carefully to obtain the best accuracy in the automatic classification of arrhythmia using the ECG signals. The proposed model was evaluated in different experiments with various values of learning parameters. For a smaller value of the learning rate (i.e., less than 0.0005), the speed of the convergence was very slow. However, when the value of the learning rate was large (i.e., greater than 0.001), the speed of convergence improved. At the same time, asymmetrical changes were observed in the accuracy rate. Henceforth, we selected an optimum value of 0.001 for the learning rate, as this value can attain better accuracy for the proposed model (i.e., optimum value)

```
/opt/conda/envs/Python-3.9/lib/python3.9/site-packages/tensorflow/python/keras/engine/training.py:1940: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future versio
n. Please use `Model.fit`, which supports generators.
  warnings.warn('`Model.fit_generator` is deprecated and '
Epoch 1/10
480/480 [==============================] - 466s 968ms/step - loss: 1.2137 - accuracy: 0.5979 - val_loss: 0.8051 - val_accuracy: 0.6967
Epoch 2/10
480/480 [==============================] - 463s 965ms/step - loss: 0.4495 - accuracy: 0.8578 - val_loss: 0.6057 - val_accuracy: 0.7962
Epoch 3/10
480/480 [==============================] - 462s 962ms/step - loss: 0.3797 - accuracy: 0.8824 - val_loss: 0.6401 - val_accuracy: 0.8166
Epoch 4/10
480/480 [==============================] - 462s 963ms/step - loss: 0.3556 - accuracy: 0.8902 - val_loss: 0.6136 - val_accuracy: 0.8252
Epoch 5/10
480/480 [==============================] - 459s 955ms/step - loss: 0.3442 - accuracy: 0.8937 - val_loss: 0.6836 - val_accuracy: 0.8019
Epoch 6/10
480/480 [==============================] - 461s 960ms/step - loss: 0.3178 - accuracy: 0.9053 - val_loss: 0.7337 - val_accuracy: 0.7704
Epoch 7/10
480/480 [==============================] - 458s 955ms/step - loss: 0.3188 - accuracy: 0.9010 - val_loss: 0.6205 - val_accuracy: 0.8421
Epoch 8/10
480/480 [==============================] - 459s 956ms/step - loss: 0.2825 - accuracy: 0.9171 - val_loss: 0.6164 - val_accuracy: 0.8344
Epoch 9/10
480/480 [==============================] - 457s 953ms/step - loss: 0.2486 - accuracy: 0.9271 - val_loss: 0.5976 - val_accuracy: 0.8299
Epoch 10/10
480/480 [==============================] - 455s 948ms/step - loss: 0.2375 - accuracy: 0.9327 - val_loss: 0.5734 - val_accuracy: 0.8533
Out[25]: <tensorflow.python.keras.callbacks.History at 0x7f956ae6fd60>

In [26]: model.save('ECG.h5')

In [27]: !tar -zcvf image-Classification-model_new.tgz ECG.h5

ECG.h5

In [28]: ls -1

data/
ECG.h5
image-Classification-model_new.tgz
```

```
In [6]: model=load_model("ECG.h5")

In [15]: #test_case_1
         img=image.load_img('fig_35.png',target_size=(192,128))

In [16]: import numpy as np
         x=image.img_to_array(img)
         x=np.expand_dims(x,axis=0)

In [17]: x.shape

Out[17]: (1, 192, 128, 3)

In [18]: y=np.argmax(model.predict(x))
         1/1 [==============================] - 0s 25ms/step

In [19]: y

Out[19]: 4

In [20]: index=['Left Bundle Branch Block','Normal','Premature Atrial Contractions','Premature Ventricular Contractions','Right Bundle Bra

In [21]: index[y]

Out[21]: 'Right Bundle Branch Block'

In [23]: #test_case_2

In [27]: img=image.load_img('fig_5749.png',target_size=(192,128))
```

```
In [23]: #test_case_2

In [27]: img=image.load_img('fig_5749.png',target_size=(192,128))

In [28]: import numpy as np
         x=image.img_to_array(img)
         x=np.expand_dims(x,axis=0)

In [29]: y=np.argmax(model.predict(x))
         1/1 [==============================] - 0s 23ms/step

In [30]: index[y]

Out[30]: 'Premature Ventricular Contractions'

In [31]: #test_case_3

In [33]: img=image.load_img('fig_3497.png',target_size=(192,128))
         import numpy as np
         x=image.img_to_array(img)
         x=np.expand_dims(x,axis=0)
         y=np.argmax(model.predict(x))

         y
         index[y]
         1/1 [==============================] - 0s 27ms/step

Out[33]: 'Normal'
```

10.ADVANTAGES AND DISADVANTAGES

Advantages:

Your healthcare provider will be able to explain your results to you such as,

- Heart arrhythmias, such as premature ventricular complexes or atrial fibrillation
- Whether you have conduction abnormalities, which result from issues regarding how the electrical impulse spreads across the heart (such as with a bundle branch block)

- Signs of an ongoing or a prior myocardial infarction (heart attack)
- Whether you have signs of severe coronary artery disease (CAD), such as stable angina or unstable angina
- If your heart muscle has become abnormally thickened, as in hypertrophic cardiomyopathy
- Signs of congenital electrical abnormalities, such as Brugada syndrome
- Electrolyte imbalances, particularly elevated or decreased levels of potassium, calcium, or magnesium
- Congenital (from birth) heart defects
- Infections involving the heart, such as pericarditis, which is an infection of the protective tissue surrounding the heart

Disadvantages

- The ECG reveals the heart rate and rhythm only during the few seconds it takes to record the tracing.
- If an arrhythmia (heart rhythm irregularity) occurs only intermittently, an ECG might not pick it up, and ambulatory monitoring may be required.
- The ECG is often normal or nearly normal with many types of heart disease, such as coronary artery disease.
- Sometimes, abnormalities that appear on the ECG turn out to have no medical significance after a thorough evaluation is done.

11.CONCLUSION

The study presented results of a review on different methods for classifying arrhythmia on ECG signals. The objective of the review method was to investigate the most powerful Deep Learning methods detecting various types of arrhythmia. Technical details of the common methods were discussed. The GRU/LSTM, CNN, and LSTM, showed outstanding results for correct classification of Atrial Fibrillation, Supraventricular Ectopic Beats, and Ventricular Ectopic Beats, respectively. It is also concluded that the use of a proper type of Deep Learning method can considerably improve the classification performance for the corresponding application.

12.FUTURE SCOPE

According to the best classification methods represented, CNN-based have proven to be effective for arrhythmia classification. Recent trend of research in this scope shows that dynamic classification methods that are capable to learn both short and long term contents of the signal in an efficient way, would be employed for such applications. CNN has shown excellent performance in classifying different types of arrhythmia. This powerful method would be one of the most efficient learning tool for this application.

13.APPENDIX

Deep Learning Method	Computational Complexity
MLP	medium-complexity
CNN	high-complexity
DBN	low-complexity
RNN	medium-complexity
LSTM	medium-complexity
GRU	low-complexity

13.1 SOURCE CODE:

```
 pwd
!pip install keras==2.2.4
!pip install tensorflow==2.5.0import os,
types

import pandas as pd

from botocore.client import Configimport ibm_boto3

def____iter____(self): return 0#

@hidden_cell
# The following code accesses a file in your IBM Cloud Object Storage. It includesyour 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='sZmW7ChAxF_z7fqdh9QjWZaoANyi2onbO3YJsULM0GGe',
        ibm_auth_endpoint="https://iam.cloud.ibm.com/oidc/token",
        config=Config(signature_version='oauth'), endpoint_url='https://s3.private.eu.cloud-object-
        storage.appdomain.cloud')

bucket = 'classificationofecg-donotdelete-pr-pvvx2hiz4wniw3'
object_key = 'Classification of Arrhythmia by Using Deep Learning with 2-D ECGSpectral Image
Representation.zip'

streaming_body_2 = 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 thepossibilities to load the data.
# ibm_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/# pandas
documentation: http://pandas.pydata.org/

from io import BytesIOimport
zipfile
unzip=zipfile.ZipFile(BytesIO(streaming_body_2.read()),'r')file_paths=unzip.namelist()
for path in file_paths:
        unzip.extract(path)
from   tensorflow.keras.preprocessing.image import ImageDataGenerator#image_augmentation
train_ds=ImageDataGenerator(rescale=1./255,
                                                        shear_range=0.2,
                                                        zoom_range=0.2,
                                                        horizontal_flip=True,
```

```
vertical_flip=True) test_ds=ImageDataGenerator(rescale=1./255)


x_train=train_ds.flow_from_directory(r'data/train',
        target_size=(192,128),
        class_mode='categorical',batch_size=32)
#Found 15341 images belonging to 6 classes.

x_train.class_indices
x_test=test_ds.flow_from_directory(r'data/test',
        target_size=(192,128),
        class_mode='categorical',batch_size=32)
x_train.class_indices#sprint-
2
#create model
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D,MaxPooling2D,Flatten,Densemodel=Sequential()
#add layers model.add(Convolution2D(32,(3,3),input_shape=(192,128,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2))) model.add(Convolution2D(32,(3,3),activation='relu'))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten()) model.add(Dense(32))
model.add(Dense(6,activation='softmax'))
model.summary()
model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])
model.fit_generator(generator=x_train,steps_per_epoch=len(x_train),epochs=10,valid
ation_data=x_test,validation_steps=len(x_test))
model.save('ECG.h5')
!tar -zcvf image-Classification-model_new.tgz ECG.h5ls -1
#testing the model
from tensorflow.keras.models import load_model from
tensorflow.keras.preprocessing import image
model=load_model("ECG.h5")
img1=image.load_img(r'data/test/Premature Ventricular Contractions/VEBfig_13.png')img1
img1=img1.resize((128,192))
x=image.img_to_array(img1) x
import numpy as np
x=np.expand_dims(x,axis=0)
y=np.argmax(model.predict(x))y
index=['Left Bundle Branch Block','Normal','Premature Atrial Contractions','Premature Ventricular
Contractions','Right Bundle BranchBlock','Ventricular Fibrillation']
index[y] client.repository.download(model_id,'my_model.tar.gz')import
tensorflow as tf
tf.____version_____
!pip install keras==2.2.4
```

```
#deployment
!pip install   watson-machine-learning--Client from
ibm_watson_machine_learning import APIClient
wml_credentials={
        "url":"https://us-south.ml.cloud.ibm.com", "apikey":"jODT-
        AnyGz3AWuG_kZdrQUOBNM5whihNrQnnLZ-h1x3U"
}
client=APIClient(wml_credentials)client
def guid_space_name(client,img_class): space=client.spaces.get_details()
        return(next(item for item in space['resources'] if
item['entity']['name']==ecg_deploy)['metadata']['id'])
space_uid=guid_space_name(client,'ecg_deploy') print("Space
UID"+space_uid) client.set.default_space(space_uid)
software_space_uid=client.software_specifications.get_uid_by_name('tensorflow_1.15-py3.6')
software_space_uid model_details=client.repository.store_model(model='ECG.h5',meta_props={
        client.repository.ModelMetaNames.NAME:"CNN",
        client.repository.ModelMetaNames.TYPE:'KERAS_2.2.4',
        client.repository.ModelMetaNames.SOFTWARE_SPEC_UID:software_space_uid
})
model_id=client.repository.get_model_uid(model_details) model_id
client.repository.download(model_id,'my_model.tar.gz')
client.repository.download(model_id,'fruit-training.ter.gz')
```

13.2 Github & Project demo link

https://github.com/IBM-EPBL/IBM-Project-6834-1658839482

DEMO LINK

https://youtu.be/8gAc-6oG8Rk