IBM-Project-26636-1660031573

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

TEAM DETAILS:

Team ID: PNT2022TMID21442

College Name: Thiagarajar College of Engineering

Department: ECE

TEAM MEMBERS:

- VIJAY ANAND B (TEAM LEADER)
- ABHISHEKVARUN M
- ARJUNE RAM KUMAR M
- ASWIN R

TABLE OF CONTENT

1. INTRODUCTION

1.1 PROJECT 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 convolutional neural network (CNN), in which we classify ECG into seven categories, one being normal and the other six being different types of arrhythmia using deep two-dimensional CNN with grayscale ECG images. We are creating a web application where the user selects the image which is to be classified.

1.2 PURPOSE:

An electrocardiogram (ECG) measures the electric activity of the heart and has been widely used for detecting heart diseases due to its simplicity and non-invasive nature. By analyzing the electrical signal of each heartbeat, i.e., the combination of action impulse waveforms produced by different specialized cardiac tissues found in the heart, it is possible to detect some of its abnormalities.

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 arrhythmias including atrial fibrillation, premature contraction, ventricular fibrillation, and tachycardia

2.2 REFERENCES

Bibliography

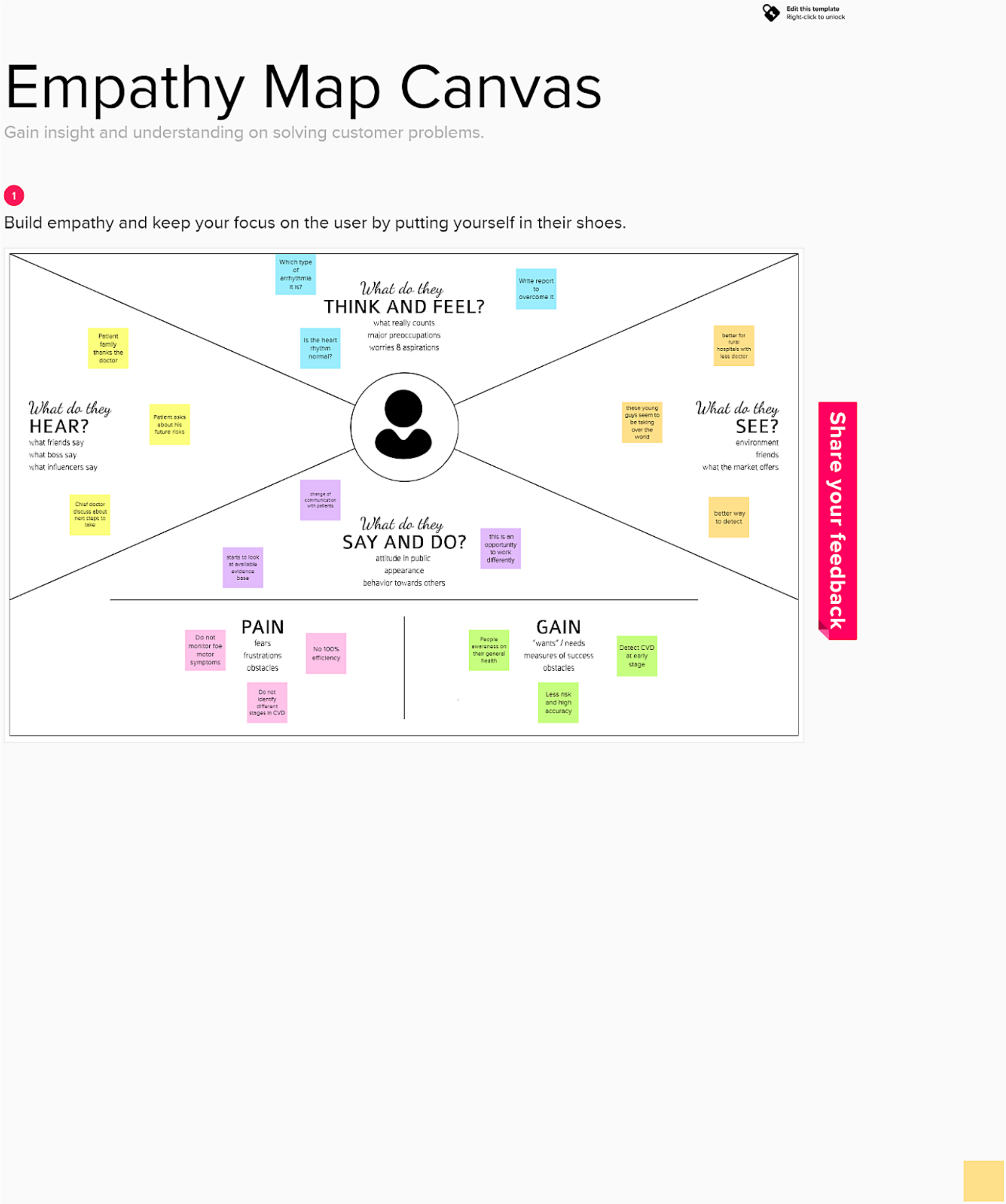
- https://www.frontiersin.org/articles/10.3389/fncom.2020.5640

15/full
- https://www.sciencedirect.com/science/article/pii/S0169260715003314

2.3 PROBLEM STATEMENT DEFINITION

3. IDEATION & PROPOSED SOLUTION

3.1 EMPATHY MAP CANVAS



Empathy Map Canvas

Gain insight and understanding on solving customer problems.

1

Build empathy and keep your focus on the user by putting yourself in their shoes.

What do they
THINK AND FEEL?
what really counts
major preoccupations
worries & aspirations

Which type of arrhythmia it is?

Is the heart rhythm normal?

Write report to overcome it

What do they
HEAR?
what friends say
what boss say
what influencers say

Patient family thanks the doctor

Patient asks about his future risks

Chief doctor discuss about next steps to take

What do they
SEE?
environment
friends
what the market offers

better for rural hospitals with less doctor

these young guys seem to be taking over the world

better way to detect

What do they
SAY AND DO?
attitude in public
appearance
behavior towards others

change of communication with patients

starts to look at available evidence base

this is an opportunity to work differently

PAIN
fears
frustrations
obstacles

Do not monitor foe motor symptoms

No 100% efficiency

Do not identify different stages in CVD

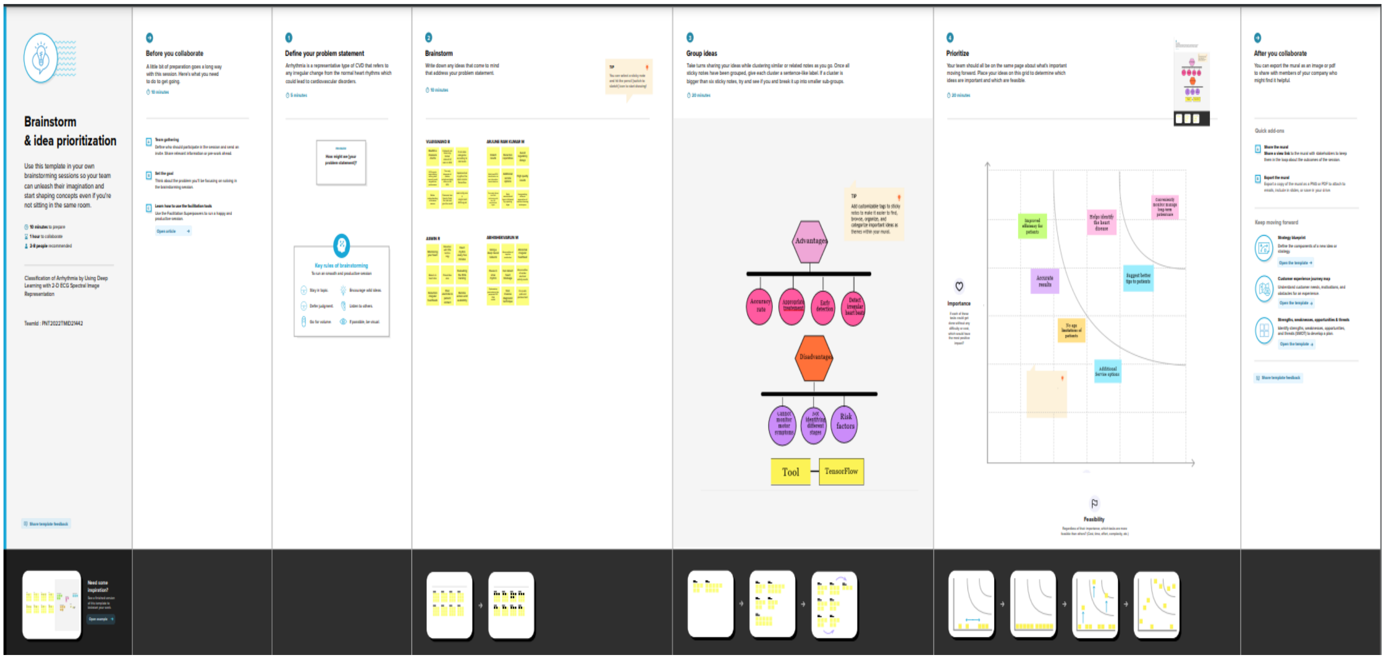
GAIN
"wants" / needs
measures of success
obstacles

People awareness on their general health

Detect CVD at early stage

Less risk and high accuracy

3.2 IDEATION AND BRAINSTORMING



Brainstorm & idea prioritization

Use this template in your own brainstorming sessions so your team can unleash their imagination and start shaping concepts even if you're not sitting in the same room.

10 minutes to prepare
1 hour to collaborate
2-8 people recommended

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

TeamId : PNT2022TMID21442

Before you collaborate

A little bit of preparation goes a long way with this session. Here's what you need to do to get going.

10 minutes

A Team gathering
Define who should participate in the session and send an invite. Share relevant information or pre-work ahead.

B Set the goal
Think about the problem you'll be focusing on solving in the brainstorming session.

C Learn how to use the facilitation tools
Use the Facilitation Superpowers to run a happy and productive session.

Open article →

1 Define your problem statement

Arrhythmia is a representative type of CVD that refers to any irregular change from the normal heart rhythms which could lead to cardiovascular disorders.

5 minutes

How might we [your problem statement]?

Key rules of brainstorming
To run an smooth and productive session

Stay in topic.
Encourage wild ideas.
Defer judgment.
Listen to others.
Go for volume.
If possible, be visual.

2 Brainstorm

Write down any ideas that come to mind that address your problem statement.

10 minutes

TIP
You can select a sticky note and hit the pencil [switch to sketch] icon to start drawing!

VIJAYANAND B

ARJUNE RAM KUMAR M

ASWIN R

ABHISHEKVARUN M

3 Group ideas

Take turns sharing your ideas while clustering similar or related notes as you go. Once all sticky notes have been grouped, give each cluster a sentence-like label. If a cluster is bigger than six sticky notes, try and see if you and break it up into smaller sub-groups.

20 minutes

TIP
Add customizable tags to sticky notes to make it easier to find, browse, organize, and categorize important ideas as themes within your mural.

4 Prioritize

Your team should all be on the same page about what's important moving forward. Place your ideas on this grid to determine which ideas are important and which are feasible.

20 minutes

Improved efficiency for patients
Helps identify the heart disease
Conveniently monitor manage long-term patientcare
Accurate results
Suggest better tips to patients
No age limitations of patients
Additional Service options

Importance
If each of these tasks could get done without any difficulty or cost, which would have the most positive impact?

Feasibility
Regardless of their importance, which tasks are more feasible than others? (Cost, time, effort, complexity, etc.)

After you collaborate

You can export the mural as an image or pdf to share with members of your company who might find it helpful.

Quick add-ons

A Share the mural
Share a view link to the mural with stakeholders to keep them in the loop about the outcomes of the session.

B Export the mural
Export a copy of the mural as a PNG or PDF to attach to emails, include in slides, or save in your drive.

Keep moving forward

Strategy blueprint
Define the components of a new idea or strategy.
Open the template →

Customer experience journey map
Understand customer needs, motivations, and obstacles for an experience.
Open the template →

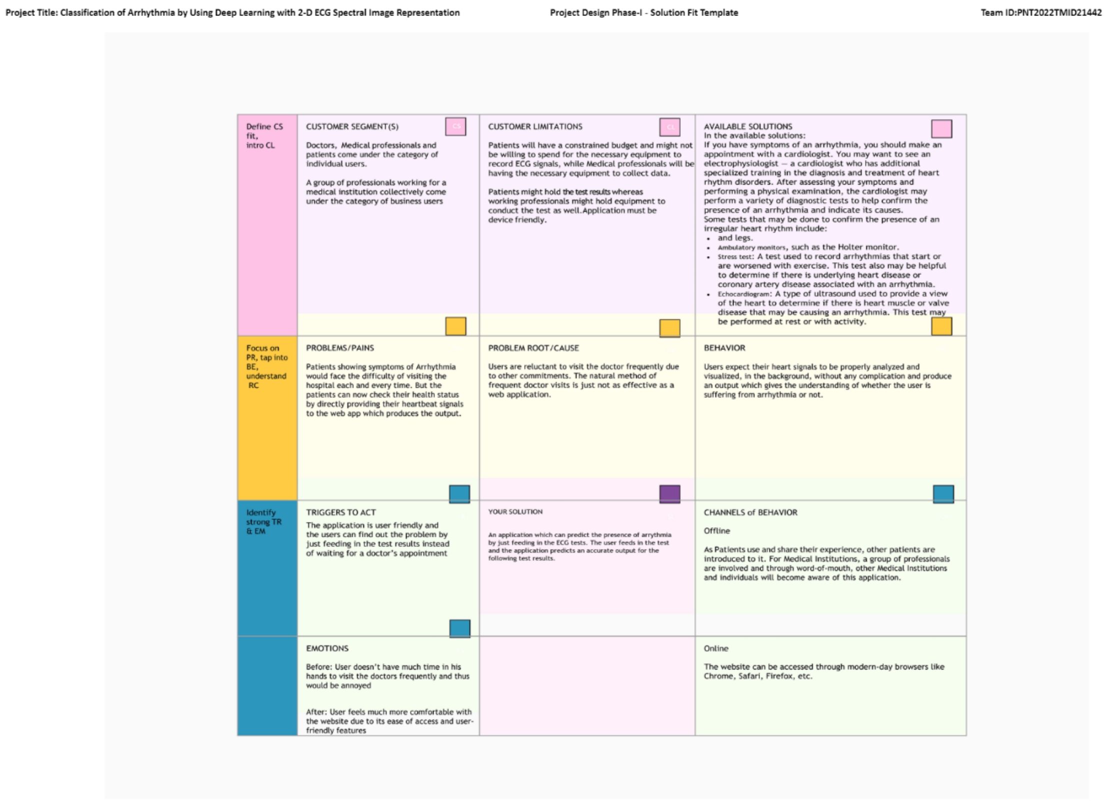
Strengths, weaknesses, opportunities & threats
Identify strengths, weaknesses, opportunities, and threats (SWOT) to develop a plan.
Open the template →

Share template feedback

3.3 PROPOSED SOLUTION

An "ambulatory electrocardiogram" or an ECG) about the size of a postcard or digital camera that you'll use for 1 to 2 days, or up to 2 weeks. The test measures the movement of electrical signals or waves through your heart. These signals tell the heart to contract (squeeze) and pump blood. You'll 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. You can do everything but shower or bathe while wearing the electrodes. After the test period, you'll go back to see your doctor. They'll download the information.

3.4 PROBLEM SOLUTION FIT

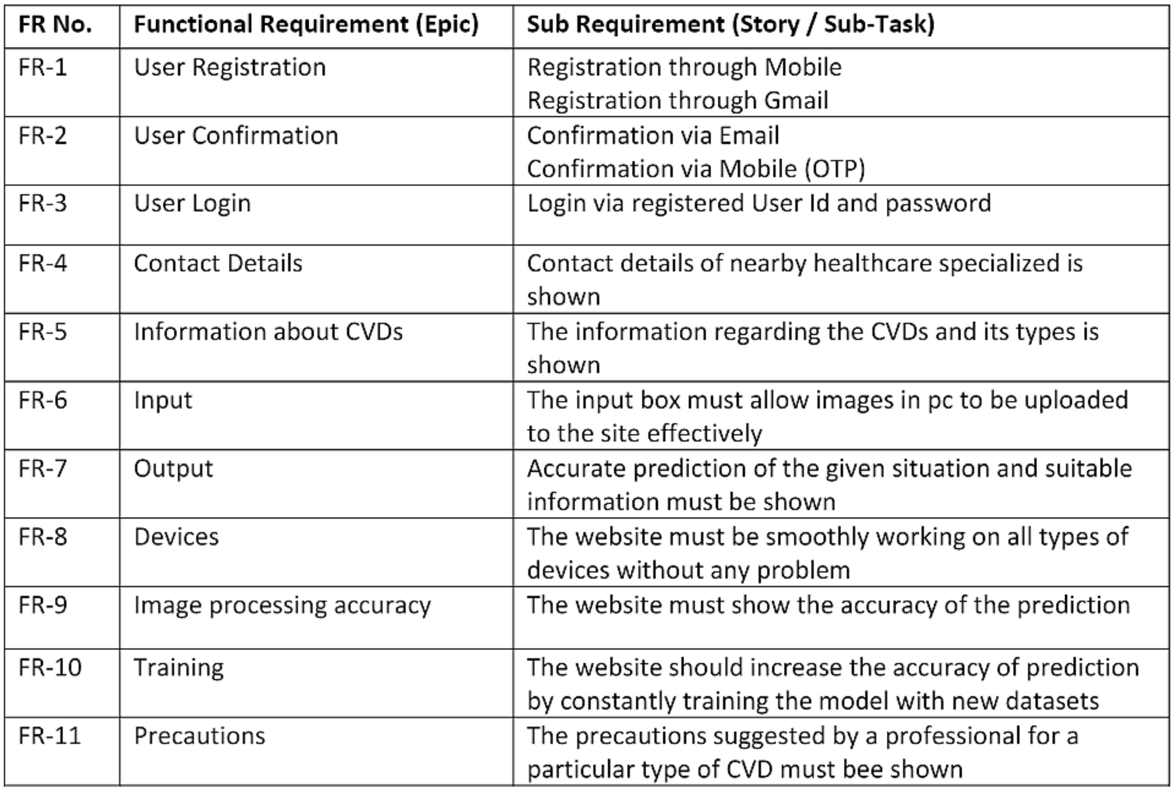


Project Title: Classification of Arrhythmia by Using Deep Learning with 2-D ECG Spectral Image Representation Project Design Phase-I - Solution Fit Template Team ID:PNT2022TMID21442

Define CS fit, intro CL	CUSTOMER SEGMENT(S) Doctors, Medical professionals and patients come under the category of individual users. A group of professionals working for a medical institution collectively come under the category of business users	CUSTOMER LIMITATIONS Patients will have a constrained budget and might not be willing to spend for the necessary equipment to record ECG signals, while Medical professionals will be having the necessary equipment to collect data. Patients might hold the test results whereas working professionals might hold equipment to conduct the test as well.Application must be device friendly.	AVAILABLE SOLUTIONS In the available solutions: If you have symptoms of an arrhythmia, you should make an appointment with a cardiologist. You may want to see an electrophysiologist — a cardiologist who has additional specialized training in the diagnosis and treatment of heart rhythm disorders. After assessing your symptoms and performing a physical examination, the cardiologist may perform a variety of diagnostic tests to help confirm the presence of an arrhythmia and indicate its causes. Some tests that may be done to confirm the presence of an irregular heart rhythm include: • and legs. • Ambulatory monitors, such as the Holter monitor. • Stress test: A test used to record arrhythmias that start or are worsened with exercise. This test also may be helpful to determine if there is underlying heart disease or coronary artery disease associated with an arrhythmia. • Echocardiogram: A type of ultrasound used to provide a view of the heart to determine if there is heart muscle or valve disease that may be causing an arrhythmia. This test may be performed at rest or with activity.
Focus on PR, tap into BE, understand RC	PROBLEMS/PAINS Patients showing symptoms of Arrhythmia would face the difficulty of visiting the hospital each and every time. But the patients can now check their health status by directly providing their heartbeat signals to the web app which produces the output.	PROBLEM ROOT/CAUSE Users are reluctant to visit the doctor frequently due to other commitments. The natural method of frequent doctor visits is just not as effective as a web application.	BEHAVIOR Users expect their heart signals to be properly analyzed and visualized, in the background, without any complication and produce an output which gives the understanding of whether the user is suffering from arrhythmia or not.
Identify strong TR & EM	TRIGGERS TO ACT The application is user friendly and the users can find out the problem by just feeding in the test results instead of waiting for a doctor's appointment	YOUR SOLUTION An application which can predict the presence of arrythmia by just feeding in the ECG tests. The user feeds in the test and the application predicts an accurate output for the following test results.	CHANNELS of BEHAVIOR Offline As Patients use and share their experience, other patients are introduced to it. For Medical Institutions, a group of professionals are involved and through word-of-mouth, other Medical Institutions and individuals will become aware of this application.
	EMOTIONS Before: User doesn't have much time in his hands to visit the doctors frequently and thus would be annoyed After: User feels much more comfortable with the website due to its ease of access and user-friendly features		Online The website can be accessed through modern-day browsers like Chrome, Safari, Firefox, etc.

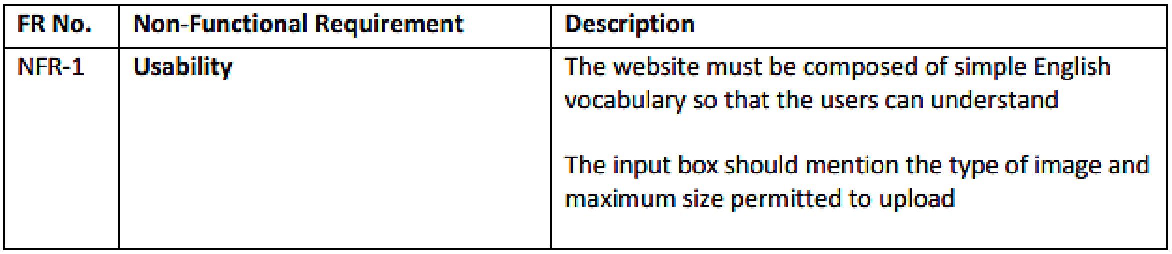
4. REQUIREMENT ANALYSIS

4.1 FUNCTIONAL REQUIREMENT

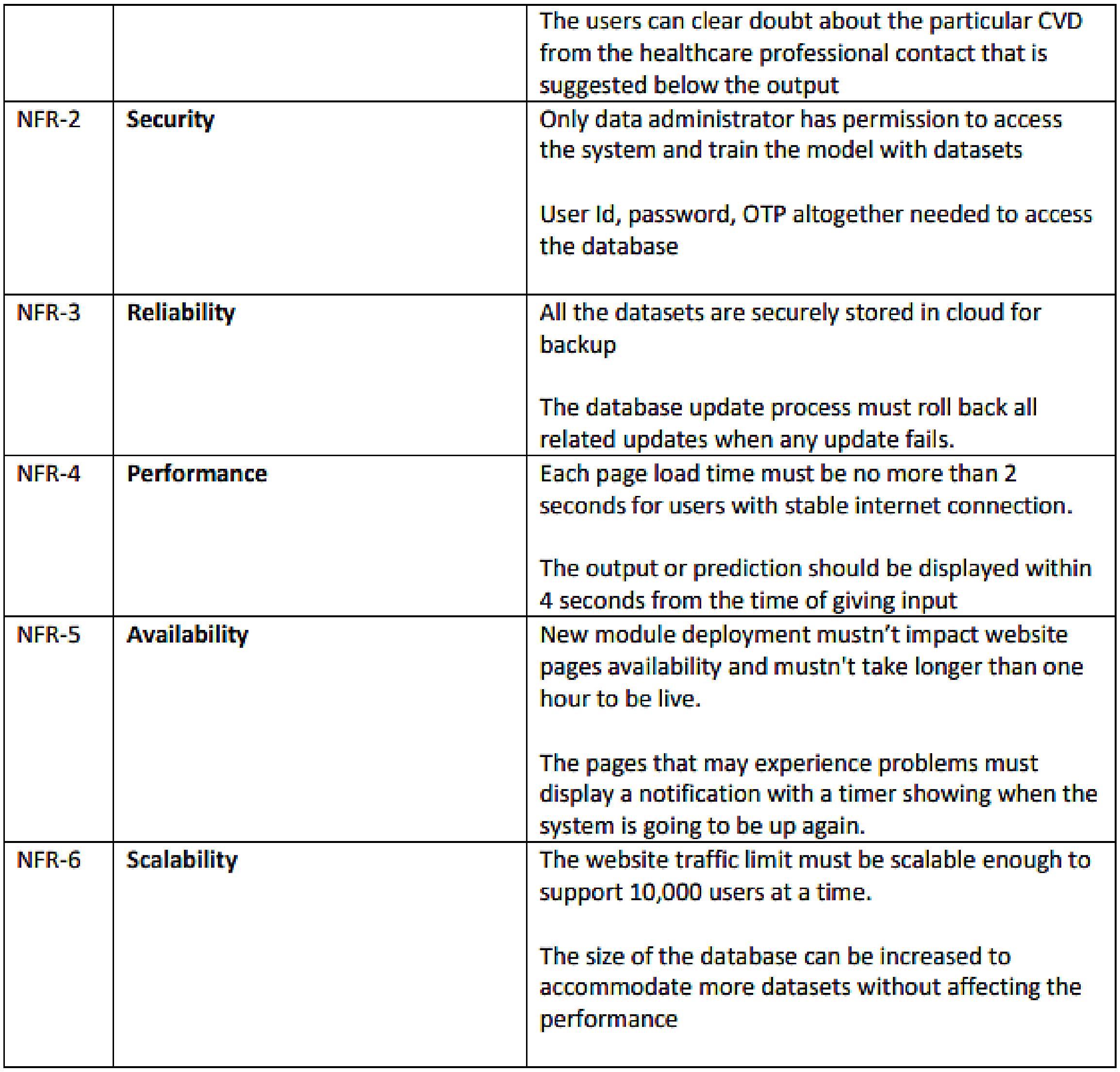


FR No.	Functional Requirement (Epic)	Sub Requirement (Story / Sub-Task)
FR-1	User Registration	Registration through Mobile Registration through Gmail
FR-2	User Confirmation	Confirmation via Email Confirmation via Mobile (OTP)
FR-3	User Login	Login via registered User Id and password
FR-4	Contact Details	Contact details of nearby healthcare specialized is shown
FR-5	Information about CVDs	The information regarding the CVDs and its types is shown
FR-6	Input	The input box must allow images in pc to be uploaded to the site effectively
FR-7	Output	Accurate prediction of the given situation and suitable information must be shown
FR-8	Devices	The website must be smoothly working on all types of devices without any problem
FR-9	Image processing accuracy	The website must show the accuracy of the prediction
FR-10	Training	The website should increase the accuracy of prediction by constantly training the model with new datasets
FR-11	Precautions	The precautions suggested by a professional for a particular type of CVD must bee shown

4.2 NON-FUNCTION REQUIREMENT



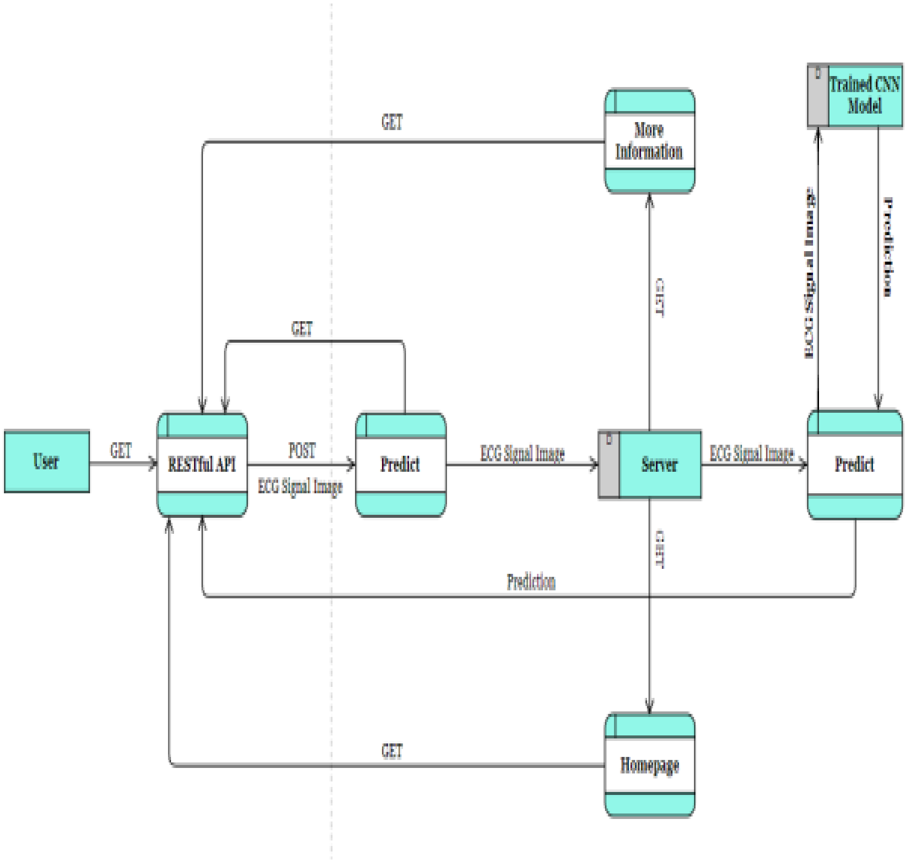
FR No.	Non-Functional Requirement	Description
NFR-1	**Usability**	The website must be composed of simple English vocabulary so that the users can understand The input box should mention the type of image and maximum size permitted to upload



		The users can clear doubt about the particular CVD from the healthcare professional contact that is suggested below the output
NFR-2	**Security**	Only data administrator has permission to access the system and train the model with datasets User Id, password, OTP altogether needed to access the database
NFR-3	**Reliability**	All the datasets are securely stored in cloud for backup The database update process must roll back all related updates when any update fails.
NFR-4	**Performance**	Each page load time must be no more than 2 seconds for users with stable internet connection. The output or prediction should be displayed within 4 seconds from the time of giving input
NFR-5	**Availability**	New module deployment mustn't impact website pages availability and mustn't take longer than one hour to be live. The pages that may experience problems must display a notification with a timer showing when the system is going to be up again.
NFR-6	**Scalability**	The website traffic limit must be scalable enough to support 10,000 users at a time. The size of the database can be increased to accommodate more datasets without affecting the performance

5.PROJECT DESIGN

5.1 DATA FLOW DIAGRAMS

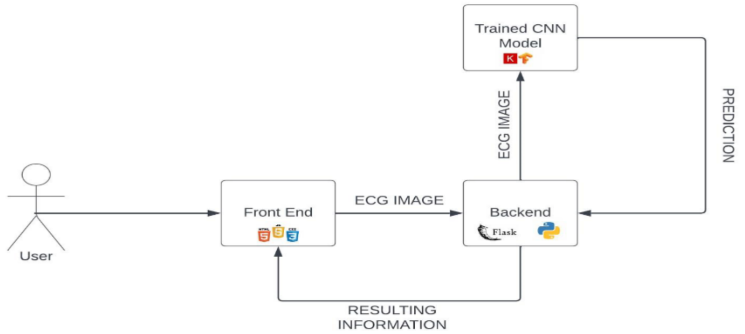


5.2 SOLUTION AND TECHNICAL ARCHITECTURE

Project Design Phase-I
Solution Architecture

Date	06 October 2022
Team ID	PNT2022TMID21442
Project Name	Classification of Arrhythmia by Using Deep Learning with 2-D ECG Spectral Image Representation
Maximum Marks	4 Marks

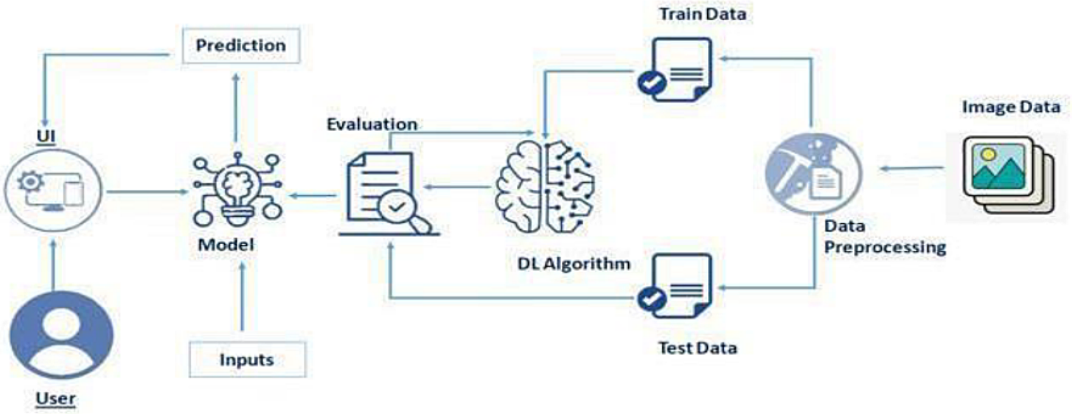
Solution Architecture Diagram:



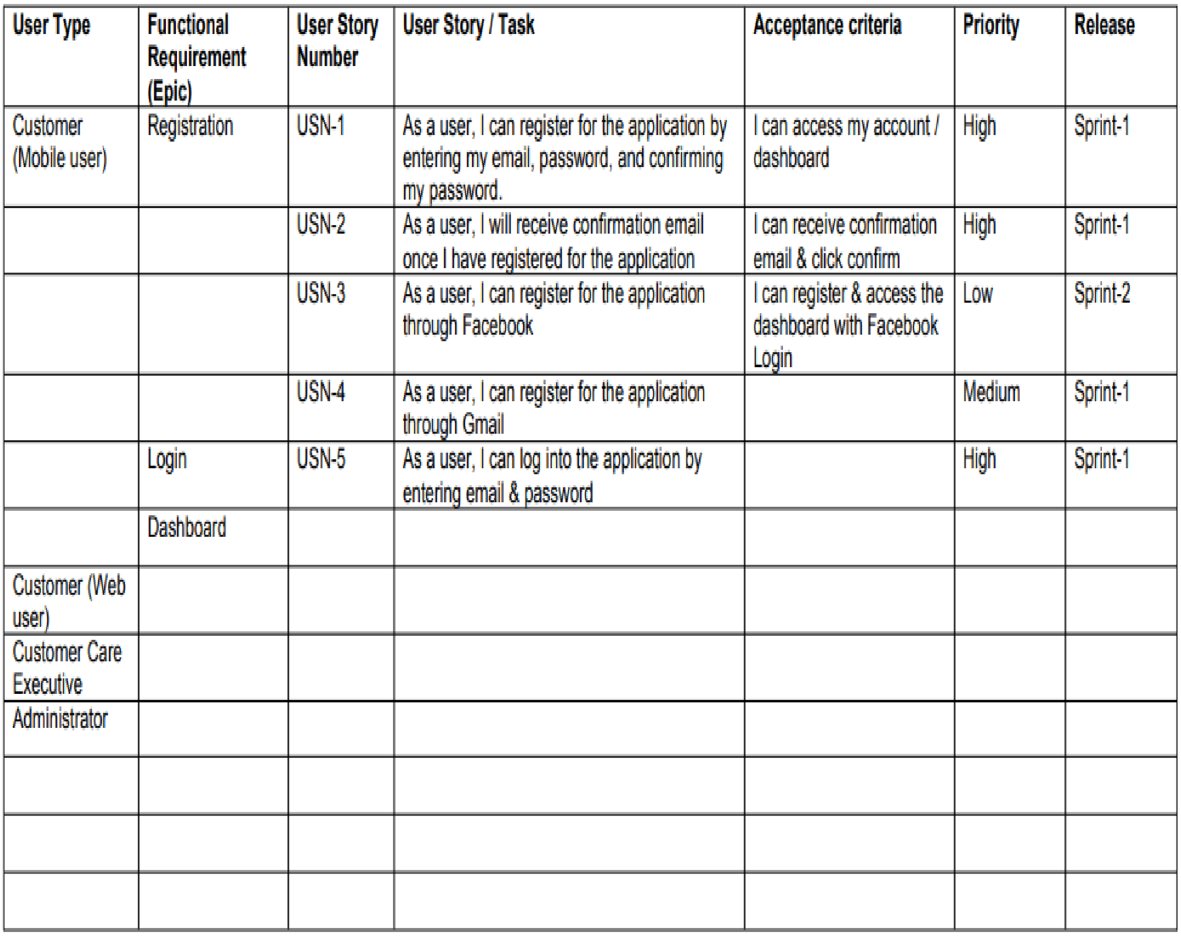
Technologies needed for Minimum Viable Product deployment

Upon research, it was found that we need to require the following software technologies for the systematic development and deployment of the project:

- HTML/CSS/JavaScript/Bootstrap – Front End Development
- Python
- Tensor Flow
- Image Processing Basics
- Flask – Backend Development
- Git & GitHub – Project Management
- IBM Cloud – Hosting
- IBM Watson – Training the Deep Learning Model



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	Sprint-1
		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	Sprint-1
		USN-3	As a user, I can register for the application through Facebook	I can register & access the dashboard with Facebook Login	Low	Sprint-2
		USN-4	As a user, I can register for the application through Gmail		Medium	Sprint-1
	Login	USN-5	As a user, I can log into the application by entering email & password		High	Sprint-1
	Dashboard					
Customer (Web user)						
Customer Care Executive						
Administrator						

6.PROJECT PLANNING & SCHEDULING

‘Project Planning and Scheduling’, though separate, are two sides of the same coin in project management. Fundamentally, ‘Project planning’ is all about choosing and designing effective policies and methodologies to attain project objectives. While ‘Project scheduling’ is a procedure of assigning tasks to get them completed by allocating appropriate resources within an estimated budget and time-frame. The basis of project planning is the entire project. Unlikely, project scheduling focuses only on the project-related tasks, the project start/end dates and project dependencies. Thus, a ‘project plan’ is a comprehensive document that contains the project aims, scope, costing, risks, and schedule. And a project schedule includes the estimated dates and sequential project tasks to be executed. Project Planning The project planning phase refers to: ● Developing a project to make it ready for investment ● Determines the jobs/tasks required to attain project objectives

6.1 SPRINT PLANNING AND ESTIMATION

What is sprint planning?

- Sprint planning is an event in scrum that kicks off the sprint.

- The purpose of sprint planning is to define what can be delivered in the sprint and how that work will be achieved.
- Sprint planning is done in collaboration with the whole scrum team.
- In scrum, the sprint is a set period of time where all the work is done.
- However, before you can leap into action you have to set up the sprint.
- You need to decide on how long the time box is going to be, the sprint goal, and where you're going to start.
- The sprint planning session kicks off the sprint by setting the agenda and focus. If done correctly, it also creates an environment where the team is motivated, challenged, and can be successful.
- Bad sprint plans can derail the team by setting unrealistic expectations

6.2 SPRINT DELIVERY SCHEDULE

Sprint-1

- Download The Dataset
- Import ImageDataGenerator Library
- Configure ImageDataGenerator class
- Import Libraries
- IInitialize the Model

Sprint-2

- Register IBM Cloud
- Apply ImageDataGeneratorfu functionality to Trainset and Dataset
- Test the mode

Sprint-3

- Train the model on IBM
- Create Html files 2022
- Train the Model

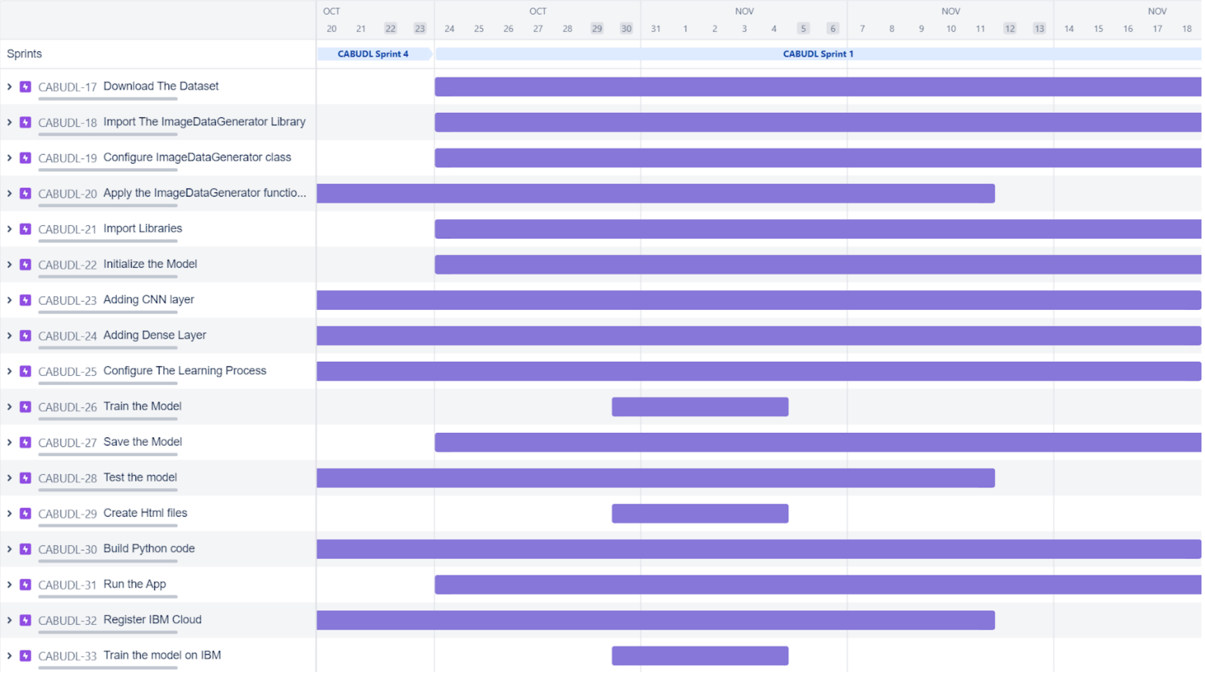
Sprint–4

- Configure The Learning Process
- Build Python code
- Adding Dense Layer
- Adding CNN laye

6.3 REPORTS FROM JIRA

Jira's value proposition heavily relies on its reporting capabilities. By delivering critical insights in real time, reporting elevates the value of your Jira deployment and empowers your team to take informed decisions that improve output and performance. It is crucial to evaluate each project's status in order to accomplish objectives and control workloads. Jira reports can aid teams in quickly identifying and resolving performance, bandwidth, and workflow obstacles, enabling them to stay on top of both short-term and long-term projects.

It's crucial to note that Jira provides a variety of tools and reports to assist you in getting a clear picture of your team's progress, each with unique advantages, restrictions, and applications. Each team or organisation should utilise the reports and resources that



Sprints

CABUDL-17 Download The Dataset

CABUDL-18 Import The ImageDataGenerator Library

CABUDL-19 Configure ImageDataGenerator class

CABUDL-20 Apply the ImageDataGenerator functio...

CABUDL-21 Import Libraries

CABUDL-22 Initialize the Model

CABUDL-23 Adding CNN layer

CABUDL-24 Adding Dense Layer

CABUDL-25 Configure The Learning Process

CABUDL-26 Train the Model

CABUDL-27 Save the Model

CABUDL-28 Test the model

CABUDL-29 Create Html files

CABUDL-30 Build Python code

CABUDL-31 Run the App

CABUDL-32 Register IBM Cloud

CABUDL-33 Train the model on IBM

7.CODING & SOLUTION

College graduates with prior programming expertise or technical degrees are recruited and transitioned into professional positions with Alabama firms and organisations through the highly competitive CodingSolutions job accelerator and talent refinement programme at no cost to the graduates. We provide a pool of varied, well-trained, techs-savvy individuals that wants to launch and advance their career in Alabama.

The mission of veteran- and woman-owned CodingSolutions is to mobilise the next generation of IT talent and provide them the tools and resources they require to make your business successful. Innovative talent is necessary for innovative technologies. We wish to provide CodingSolutions prospects to assist you expand your Alabama team.

Our applicants are swiftly hired at the top of the list by growing businesses for lucrative, long-term positions.

7.1 FEATURES

- User interacts with User interface to upload image
- Uploaded image is analyzed by the model which is integrated
- Once model analyses the uploaded image, the prediction is showcased on the UI

8. TESTING

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

8.1 TYPES OF TESTS

8.1.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

8.1.2 Integration testing

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

8.1.3 Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals. Functional testing is centered on the following items: Valid Input : identified classes of valid input must be accepted. Invalid Input : identified classes of invalid input must be rejected. Functions : identified functions must be exercised. Output : identified classes of application outputs must be exercised. Systems/Procedures: interfacing systems or procedures must be invoked. Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

8.1.4 System Test

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

8.1.5 White Box Testing

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

8.1.6 Black Box Testing

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

8.2 Unit Testing:

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

8.2.1 Test strategy and approach

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

8.2.2 Test objectives

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

8.2.3 Features to be tested

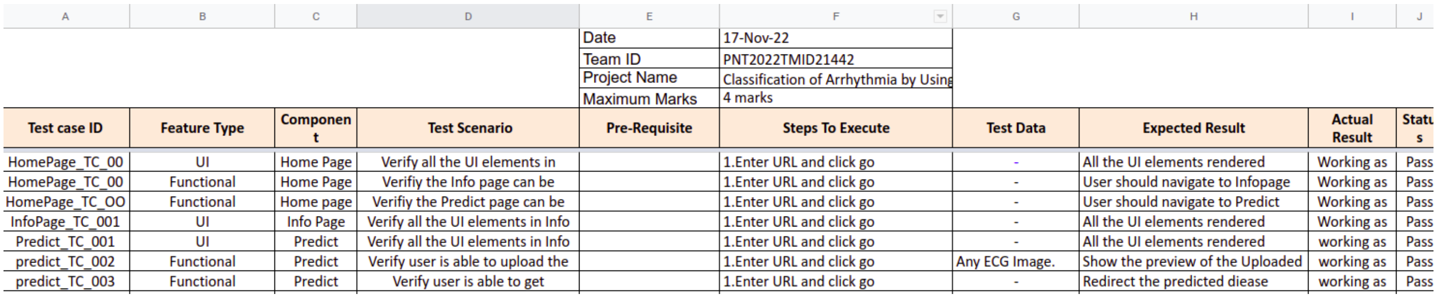
- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

8.3 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects. The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error. Test Results: All the test cases mentioned above passed successfully. No defects encountered.

8.4 Acceptance Testing

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



Date	17-Nov-22
Team ID	PNT2022TMID21442
Project Name	Classification of Arrhythmia by Using
Maximum Marks	4 marks

Test case ID	Feature Type	Component	Test Scenario	Pre-Requisite	Steps To Execute	Test Data	Expected Result	Actual Result	Status
HomePage_TC_00	UI	Home Page	Verify all the UI elements in		1.Enter URL and click go	-	All the UI elements rendered	Working as	Pass
HomePage_TC_00	Functional	Home Page	Verifiy the Info page can be		1.Enter URL and click go	-	User should navigate to Infopage	Working as	Pass
HomePage_TC_OO	Functional	Home page	Verifiy the Predict page can be		1.Enter URL and click go	-	User should navigate to Predict	Working as	Pass
InfoPage_TC_001	UI	Info Page	Verify all the UI elements in Info		1.Enter URL and click go	-	All the UI elements rendered	Working as	Pass
Predict_TC_001	UI	Predict	Verify all the UI elements in Info		1.Enter URL and click go	-	All the UI elements rendered	working as	Pass
predict_TC_002	Functional	Predict	Verify user is able to upload the		1.Enter URL and click go	Any ECG Image.	Show the preview of the Uploaded	working as	Pass
predict_TC_003	Functional	Predict	Verify user is able to get		1.Enter URL and click go	-	Redirect the predicted diease	working as	Pass

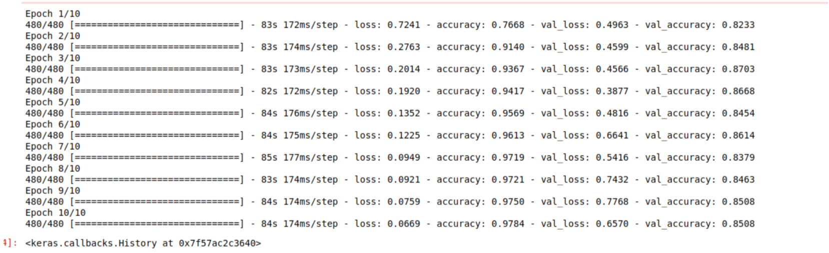
	Test Scenarios
1	Verify user is able to see Home page
2	Verify user is able to navigate Info Page
3	Verify user is able to navigate Predict Page
4	Verify user is able to see Info page
5	Verify user is able to see Predict page
6	Verify user is able to upload ECG image

7	Verify user is able to redirect to predicted result page

9.RESULTS

9.1 PERFORMANCE METRICS

Training Accuracy - 97.84
Validation Accuracy - 85.08



```
    Epoch 1/10
    480/480 [==============================] - 83s 172ms/step - loss: 0.7241 - accuracy: 0.7668 - val_loss: 0.4963 - val_accuracy: 0.8233
    Epoch 2/10
    480/480 [==============================] - 83s 174ms/step - loss: 0.2763 - accuracy: 0.9140 - val_loss: 0.4599 - val_accuracy: 0.8481
    Epoch 3/10
    480/480 [==============================] - 83s 173ms/step - loss: 0.2014 - accuracy: 0.9367 - val_loss: 0.4566 - val_accuracy: 0.8703
    Epoch 4/10
    480/480 [==============================] - 82s 172ms/step - loss: 0.1920 - accuracy: 0.9417 - val_loss: 0.3877 - val_accuracy: 0.8668
    Epoch 5/10
    480/480 [==============================] - 84s 176ms/step - loss: 0.1352 - accuracy: 0.9569 - val_loss: 0.4816 - val_accuracy: 0.8454
    Epoch 6/10
    480/480 [==============================] - 84s 175ms/step - loss: 0.1225 - accuracy: 0.9613 - val_loss: 0.6641 - val_accuracy: 0.8614
    Epoch 7/10
    480/480 [==============================] - 85s 177ms/step - loss: 0.0949 - accuracy: 0.9719 - val_loss: 0.5416 - val_accuracy: 0.8379
    Epoch 8/10
    480/480 [==============================] - 83s 174ms/step - loss: 0.0921 - accuracy: 0.9721 - val_loss: 0.7432 - val_accuracy: 0.8463
    Epoch 9/10
    480/480 [==============================] - 84s 174ms/step - loss: 0.0759 - accuracy: 0.9750 - val_loss: 0.7768 - val_accuracy: 0.8508
    Epoch 10/10
    480/480 [==============================] - 84s 174ms/step - loss: 0.0669 - accuracy: 0.9784 - val_loss: 0.6570 - val_accuracy: 0.8508
4]: <keras.callbacks.History at 0x7f57ac2c3640>
```

10. ADVANTAGES & DISADVANTAGES

1. High accuracy
2. High sensitivity
3. High reliability
4. Reduced loss
5. Do not monitor motor response
6. Do not identify different stages in CVD

11. CONCLUSION

In this study, we proposed a 2-D CNN-based classification model for automatic classificationof cardiac arrhythmias using ECG signals. An accurate taxonomy of ECG signals is extremelyhelpful in the prevention and diagnosis of CVDs. Deep CNN has proven useful in enhancing theaccuracy of diagnosis algorithms in the fusion of medicine and modern machine learning technologies.The proposed CNN-based classification algorithm, using 2-D images, can classify eight kinds ofarrhythmia, namely, NOR, VFW, PVC, VEB, RBB, LBB, PAB, and APC, and it achieved 97.91% averagesensitivity, 99.61% specificity, 99.11% average accuracy, and 98.59% positive predictive value (precision).These results indicate that the prediction and classification of arrhythmia with 2-D ECG representationas spectrograms and the CNN model is a reliable operative technique in the diagnosis of CVDs.The proposed scheme can help experts diagnose CVDs by referring to the automated classification of

Remote Sens. 2020,12, 1685 12 of 14ECG signals. The present research uses only a single-lead ECG signal. The effect of multiple lead ECGdata to further improve experimental cases will be studied in future work

12. FUTURE SCOPE

Automatic heartbeat classification is essential for real-time applications in detection of cardiac arrhythmias. Programmed heartbeat order is basic for continuous applications in the location of cardiovascular arrhythmias. The acquired consequences of this proposal recommend that there is a potential development of future in programmed ECG order frameworks. The frameworks must incorporate four conclusive advances: pre-handling, QRS complex discovery, highlights extraction and order of pulses. The further exertion of this work should move towards proposing new component extraction and arrangement strategies.

The future is using this detection of cardiac arrhythmia tools in wearable devices so that they could continuously monitor the health of the person and send alerts when there is an abnormality. We additionally recommend the utilization of new patterns to catch the ECG signal, for example, off-the-individual methodologies, for the elaboration of new databases. In any case, we accept that the making of such databases would be an extraordinary test in light of the fact that, other than the money related costs included, they would need to be consolidated into gauges, for example, AAMI measures to contact the ideal crowd**.**

13. APPENDIX

SOURCE CODE

MODEL TRAINING:

1. Data Collection

In [ ]:

```
pwd
```

In [ ]:

```
!pip install keras
!pip install tensorflow
```

In [ ]:

```
import keras
keras.__version__
```

In [ ]:

```
import tensorflow
```

```
tensorflow.__version__
```

Loading our data

In [ ]:

```
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='dlfSXsHrHx9xly751DrA5r13Jmo2sWxk76JXeII4pi2N',
    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 = 'classificationofarrhythmiabyusing-donotdelete-pr-7j5j6vzxaehedv'
object_key = 'data.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 the possibilities to load the data.
# ibm_boto3 documentation: https://ibm.github.io/ibm-cos-sdk-python/
# pandas documentation: http://pandas.pydata.org/
```

In [ ]:

```
from io import BytesIO
import zipfile
unzip = zipfile.ZipFile(BytesIO(streaming_body_2.read()),'r')
file_paths=unzip.namelist()
print(file_paths)
for path in file_paths:
    unzip.extract(path)
```

In [ ]:

pwd

2. Image Preprocessing

Import The Image data Generator

In [ ]:

```
from keras.preprocessing.image import ImageDataGenerator

#setting parameter for Image bata agumentation to the traing data
train_datagen= ImageDataGenerator(rescale=1./255, shear_range=0.2,
zoom_range=0.2, horizontal_flip=True)

#Image Data agumentation to the testing data
test_datagen = ImageDataGenerator(rescale=1./255)
```

Apply Image Data Generator Functionality To Trainset and Testset

In [ ]:

```
#performing data agumentation to train data
x_train = train_datagen.flow_from_directory(directory =
"/home/wsuser/work/data/train", target_size = (64, 64), batch_size = 32,
class_mode='categorical')

#performing data agumentation to test data
x_test =  test_datagen.flow_from_directory(directory =
"/home/wsuser/work/data/test" , target_size = (64, 64), batch_size = 32,
class_mode="categorical")
```

Confiqure Image Data Generator Class

In [ ]:

```
x_train.class_indices
```

Model Buliding

Adding Layers:

In [ ]:

```
#Import req. Lib.
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Convolution2D, MaxPooling2D, Flatten,
Dense
# Build a CNN Block:
model = Sequential() #intializing sequential model
model.add(Convolution2D(32,(3,3),activation='relu',
input_shape=(64,64,3))) #convolution laye
model.add(MaxPooling2D(pool_size=(2, 2))) #Maxpooling layer
model.add(Flatten()) #Flatten layer
model.add(Dense(400,activation='relu')) #Hidden Layer 1
model.add(Dense(200,activation='relu')) #Hidden Layer 2
model.add(Dense(6,activation='softmax')) #Output Layer
```

In [ ]:

```
model.summary()#summary of our model
```

Compiling :

In [ ]:

```
# Compiling The Model...
model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['a
ccuracy'])
```

Fit / Train The Model :

In [ ]:

```
#Train Model:
model.fit_generator(x_train,
steps_per_epoch=len(x_train),
epochs=10,
validation_data=x_test,
validation_steps=len(x_test))
```

Saving The Model :

In [ ]:

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

In [ ]:

```
!tar -zcvf ECG-Image-based-heartbeat-classification-model_new.tgz ECG.h5
```

In [ ]:

```
ls -1
```

In [ ]:

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

In [ ]:

```
# Replace the credentials that you got from watson machine learning
service
from ibm_watson_machine_learning import APIClient
wml_credentials = {
                    "url": "https://us-south.ml.cloud.ibm.com",
                    "apikey":"TkHXR1FH4Bk9bPi9BRyxTUiUQrPT9AAffkQ-
ND1CUuSS"
                  }
client = APIClient(wml_credentials)
```

In [ ]:

```
client = APIClient(wml_credentials)
```

In [ ]:

```
def guid_from_space_name(client, space_name):
    space = client.spaces.get_details()
    #print(space)
    return(next(item for item in space['resources'] if
item['entity']['name'] == space_name)['metadata']['id'])
```

In [ ]:

```
space_uid = guid_from_space_name(client, 'image_classification')
print("Space UID = "+ space_uid)
```

In [ ]:

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

In [ ]:

```
client.software_specifications.list(limit=100)
```

In [ ]:

```
import tensorflow
tensorflow.__version__
```

In [ ]:

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

In [ ]:

```
model_details = client.repository.store_model(model='ECG-Image-based-
heartbeat-classification-model_new.tgz',meta_props={
client.repository.ModelMetaNames.NAME:"Image_Classification",
client.repository.ModelMetaNames.TYPE:"tensorflow_2.7",
client.repository.ModelMetaNames.SOFTWARE_SPEC_UID: software_spec_uid})

model_id = client.repository.get_model_uid(model_details)
```

In [ ]:

```
client.repository.download(model_id, 'my_model.tar.gz')
```

In [ ]:

```
!tar xvzf my_model.tar.gz
```

In [ ]:

```
!ls -1
```

In [ ]:

```
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
```

In [ ]:

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

FLASK WEBSITE

app.py :-

```
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("about.html")#rendering html page

@app.route("/about") #default route
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':
        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

        preds=model.predict(x)#predicting classes
        pred=np.argmax(preds,axis=1)#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#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=8000)
```

index.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 %}
```

main.css

```
.img-preview {
    width: 256px;
    height: 256px;
    position: relative;
    border: 5px solid #F8F8F8;
    box-shadow: 0px 2px 4px 0px rgba(0, 0, 0, 0.1);
    margin-top: 1em;
    margin-bottom: 1em;
}
```

```css
.img-preview>div {
    width: 100%;
    height: 100%;
    background-size: 256px 256px;
    background-repeat: no-repeat;
    background-position: center;
}

input[type="file"] {
    display: none;
}

.upload-label{
    display: inline-block;
    padding: 12px 30px;
    background: #39D2B4;
    color: #fff;
    font-size: 1em;
    transition: all .4s;
    cursor: pointer;
}

.upload-label:hover{
    background: #34495E;
    color: #39D2B4;
}

.loader {
    border: 8px solid #f3f3f3; /* Light grey */
    border-top: 8px solid #3498db; /* Blue */
    border-radius: 50%;
    width: 50px;
    height: 50px;
    animation: spin 1s linear infinite;
}

@keyframes spin {
    0% { transform: rotate(0deg); }
    100% { transform: rotate(360deg); }
}
```

main.js:

```
$(document).ready(function () {
    // Init
    $('.image-section').hide();
    $('.loader').hide();
    $('#result').hide();

    // Upload Preview
    function readURL(input) {
        if (input.files && input.files[0]) {
            var reader = new FileReader();
            reader.onload = function (e) {
                $('#imagePreview').css('background-image',
'url(' + e.target.result + ')');
                $('#imagePreview').hide();
                $('#imagePreview').fadeIn(650);
            }
            reader.readAsDataURL(input.files[0]);
        }
    }
```

```
$("#imageUpload").change(function () {
    $('.image-section').show();
    $('#btn-predict').show();
    $('#result').text('');
    $('#result').hide();
    readURL(this);
});

// Predict
$('#btn-predict').click(function () {
    var form_data = new FormData($('#upload-file')[0]);

    // Show loading animation
    $(this).hide();
    $('.loader').show();

    // Make prediction by calling api /predict
    $.ajax({
        type: 'POST',
        url: '/predict',
        data: form_data,
        contentType: false,
        cache: false,
        processData: false,
        async: true,
        success: function (data) {
            // Get and display the result
            $('.loader').hide();
            $('#result').fadeIn(600);
            $('#result').text(' Result:  ' + data);
            console.log('Success!');
        },
            });
        });

    });
```

GITHUB & PROJECT DEMO LINK

Github: https://github.com/IBM-EPBL/IBM-Project-26636-1660031573
Demo :