# “Vital Data Collection and prediction using IOT and Machine Learning”

**Minor Project Report**

*Submitted in Partial Fulfillment of the*

*Requirements for the Degree of*

**BACHELOR OF TECHNOLOGY**

**IN**

**ELECTRONICS AND COMMUNICATION ENGINEERING**

By

## Dishank Garg (20BEC030)

## Jainam Jain (20BEC043)



**Department of Electronics and Communication Engineering,**

**Institute of Technology,**

**Nirma University,**

**Ahmedabad 382 481**

**November 2023**

**CERTIFICATE**

This is to certify that the Minor Project Report entitled “**Vital Data Collection and Prediction using IOT and Machine Learning”** submitted by Mr Jainam Jain (20BEC043) towards the partial fulfillment of the requirements for the award of degree in Bachelor of Technology in the field of Electronics & Communication Engineering of Nirma University is the record of work carried out by him under our supervision and guidance. The work submitted has in our opinion reached a level required for being accepted for examination. The results embodied in this minor project work to the best of our knowledge have not been submitted to any other University or Institution for award of any degree or diploma.

| **Dr. Manisha A. Upadhyay**  **(Guide)**  Department of Electronics & Communication Engineering,  Institute of Technology,  Nirma University,  Ahmedabad 382481 |  | **Dr. Usha S. Mehta**  **Head of Department**  Department of Electronics & Communication Engineering,  Institute of Technology,  Nirma University,  Ahmedabad 382 481 |
| --- | --- | --- |

**Date: 02-12-2023**

**CERTIFICATE**

This is to certify that the Minor Project Report entitled “**Vital Data Collection and Prediction using IOT and Machine Learning”** submitted by Mr Dishank Garg (20BEC030) towards the partial fulfillment of the requirements for the award of degree in Bachelor of Technology in the field of Electronics & Communication Engineering of Nirma University is the record of work carried out by him under our supervision and guidance. The work submitted has in our opinion reached a level required for being accepted for examination. The results embodied in this minor project work to the best of our knowledge have not been submitted to any other University or Institution for award of any degree or diploma.

| **Dr. Manisha A. Upadhyay**  **(Guide)**  Department of Electronics & Communication Engineering,  Institute of Technology,  Nirma University,  Ahmedabad 382481 |  | **Dr. Usha S. Mehta**  **Head of Department**  Department of Electronics & Communication Engineering,  Institute of Technology,  Nirma University,  Ahmedabad 382 481 |
| --- | --- | --- |

**Date: 02-12-2023**

### Undertaking for Originality of the Work

We, Dishank Garg and Jainam Jain, 20BEC030 and 20BEC043, give undertaking that the Minor Project entitled “Vital Data Collection and Prediction using IOT and Machine Learning” submitted by us, towards the partial fulfillment of the requirements for the degree of Bachelor of Technology inElectronics and Communication of Nirma University, Ahmedabad 382 481, is the original work carried out by us and We give assurance that no attempt of plagiarism has been made. We understand that in the event of any similarity found subsequently with any other published work or any project report elsewhere; it will result in severe disciplinary action.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature of the Student

Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Place: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Endorsed by:

(Signature of the Guide)

### Acknowledgement

A journey is more enriching when shared with others. Interdependence, rather than independence, holds the key to meaningful progress. This Project, a culmination of my efforts, bears the indelible imprint of individuals who have enriched my journey with their unwavering support and guidance.

My deepest gratitude extends to Dr. Manisha A Upadhyay, my esteemed supervisor, whose exceptional guidance, invaluable insights, and unwavering encouragement have been the bedrock of my research endeavors. Her unwavering belief in my abilities has fueled my determination, and her mentorship has shaped my growth as a researcher and a professional. I am immensely fortunate to have had her guidance throughout this transformative journey.

I would also like to express my sincere appreciation to the entire laboratory staff for their invaluable contributions. Their expertise, patience, and support have played a crucial role in the success of my research endeavors.

To my cherished class fellows, I extend my heartfelt gratitude for their unwavering camaraderie, infectious smiles, and friendship. Their presence has enriched my academic experience, making my time at Nirma University an unforgettable and enjoyable one.

Finally, I would like to express my profound gratitude to the Almighty, the Prime Mover, for showering me with His blessings and guiding me throughout this journey. His grace has been my constant companion, providing me with strength and resilience in the face of challenges.

Dishank Garg (20BEC030)

Jainam Jain (20BEC043)

### Abstract

The rising prevalence of heart disease, coupled with limited access to healthcare facilities, particularly in remote areas, poses a significant challenge to the timely diagnosis and treatment of cardiovascular conditions. Electrocardiogram (ECG) is a crucial diagnostic tool for heart ailments, but its availability and affordability remain limited. To address these challenges, the integration of Internet of Things (IoT) and machine learning (ML) technologies offers a promising approach for real-time vital data monitoring and prediction of heart health risks.

IoT-enabled wearable devices can continuously collect vital signs data, including ECG signals, from patients remotely. This data can then be transmitted to a cloud-based platform for analysis using ML algorithms. By analyzing patterns in the collected data, ML models can predict potential heart health risks, enabling early intervention and preventive measures.

This approach has the potential to significantly improve healthcare outcomes for heart patients, particularly in underserved areas. By providing real-time insights into patient health status, IoT and ML can facilitate timely diagnosis, personalized treatment plans, and improved patient outcomes. Additionally, this approach can reduce the cost of ECG checkups, making healthcare more accessible and affordable.

In conclusion, the integration of IoT and ML technologies offers a powerful solution for addressing the challenges of heart disease diagnosis and management. By enabling real-time vital data monitoring and predictive analytics, this approach can revolutionize cardiovascular healthcare, leading to improved patient outcomes and reduced healthcare costs.

INDEX

| **Chapter No.** | Title | | **Page No.** |
| --- | --- | --- | --- |
|  | Acknowledgement | | **i** |
|  | Abstract | | **ii** |
|  | Index | | **iii** |
|  | List of Figures | | **iv** |
|  | List of Tables | | **vi** |
|  | Nomenclature | | **viii** |
| **1** | Introduction | | 1 |
|  | 1.1 | Introduction | 1 |
|  | 1.2 | Motivation | 3 |
|  | 1.3 | Objective | 3 |
|  | 1.4 | Problem Statement | 4 |
|  | 1.5 | Approach | 4 |
|  | 1.6 | Scope of the Project | 5 |
|  | 1.7 | Gantt Chart (Timeline of your work) | 5 |
| **2** | Literature Review | | 6 |
|  | 2.1 | Previous Approaches to Solve the Problem (include details of research/review papers, articles, products, components, similar projects that you have studied related to the work) | 6 |
|  | 2.2 | Details of Tools and Technologies used |  |
| **3** | **Hardware Design** | | 7 |
| **4** | **Software Design** | | 8 |
| **5** | **Experimental/Simulation Results and Discussion** | | 9 |
| **6** | **Conclusions and Future Scope** | | 10 |
| **7** | **References** | | 11 |

**LIST OF FIGURES**

| **Figure**  **No.** | **Title** | **Page No.** |
| --- | --- | --- |
| 3.1 | Hardware Design | 7 |
| 4.1 | Software Flowchart | 15 |
|  |  |  |

**LIST OF TABLES**

| **Table No.** | **Title** | **Page No.** |
| --- | --- | --- |
| 3.1 | Power Consumption of Hardware Components | 2 |
| 4.1 | Simulation Parameters | 24 |

## NOMENCLATURE

| A | Energy level indicator |
| --- | --- |
| B | Bottom product rate, kmol/hr |
|  |
| **Greek** |
| θ | Root of Underwood equation |
| α | Relative volatility |
| λ | Latent heat of vaporization, kcal/kmol |
|  | Difference |
| ε | Energy change indicator |
|  |
| **Subscripts** |
| min | Minimum |
| i | Any component |
|  |  |
| **Abbreviations** |
| CGCC | Column Grand Composite Curves |
| IRS | Invariant Rectifying Stripping |
| CMO | Constant Molar Overflow |

**Chapter 1**

Introduction

1.1 Prologue

Heart disease remains a leading cause of death globally, with an estimated 17.9 million deaths attributed to cardiovascular diseases in 2016 alone. This alarming trend, coupled with the increasing prevalence of risk factors such as hypertension, obesity, and diabetes, underscores the urgent need for effective and accessible heart health monitoring strategies. Traditional methods of heart monitoring, such as electrocardiograms (ECGs), require hospital visits or specialized equipment, limiting their reach and effectiveness, particularly in resource-constrained settings. Moreover, the high cost of ECG checkups further hinders access to essential heart health monitoring services.

ML algorithms, trained on vast amounts of heart-related data, can analyze these IoT-generated signals to identify patterns and predict potential cardiac events. This predictive capability enables proactive interventions, such as lifestyle modifications, medication adjustments, or timely medical consultations, potentially preventing the occurrence of serious cardiovascular events and improving overall patient outcomes.

The integration of IoT and ML holds immense potential to transform heart health monitoring, particularly in underserved areas with limited access to healthcare facilities.

1.2 Motivation

To address these challenges, the integration of Internet of Things (IoT) and machine learning (ML) offers a promising solution for revolutionizing heart health monitoring. IoT devices, equipped with sensors and embedded systems, can continuously collect vital data, including heart rate, blood pressure, and ECG readings, enabling real-time monitoring of an individual's heart health. This continuous data stream provides a comprehensive picture of an individual's cardiovascular health, allowing for early detection of potential heart-related issues

1.3 Objective

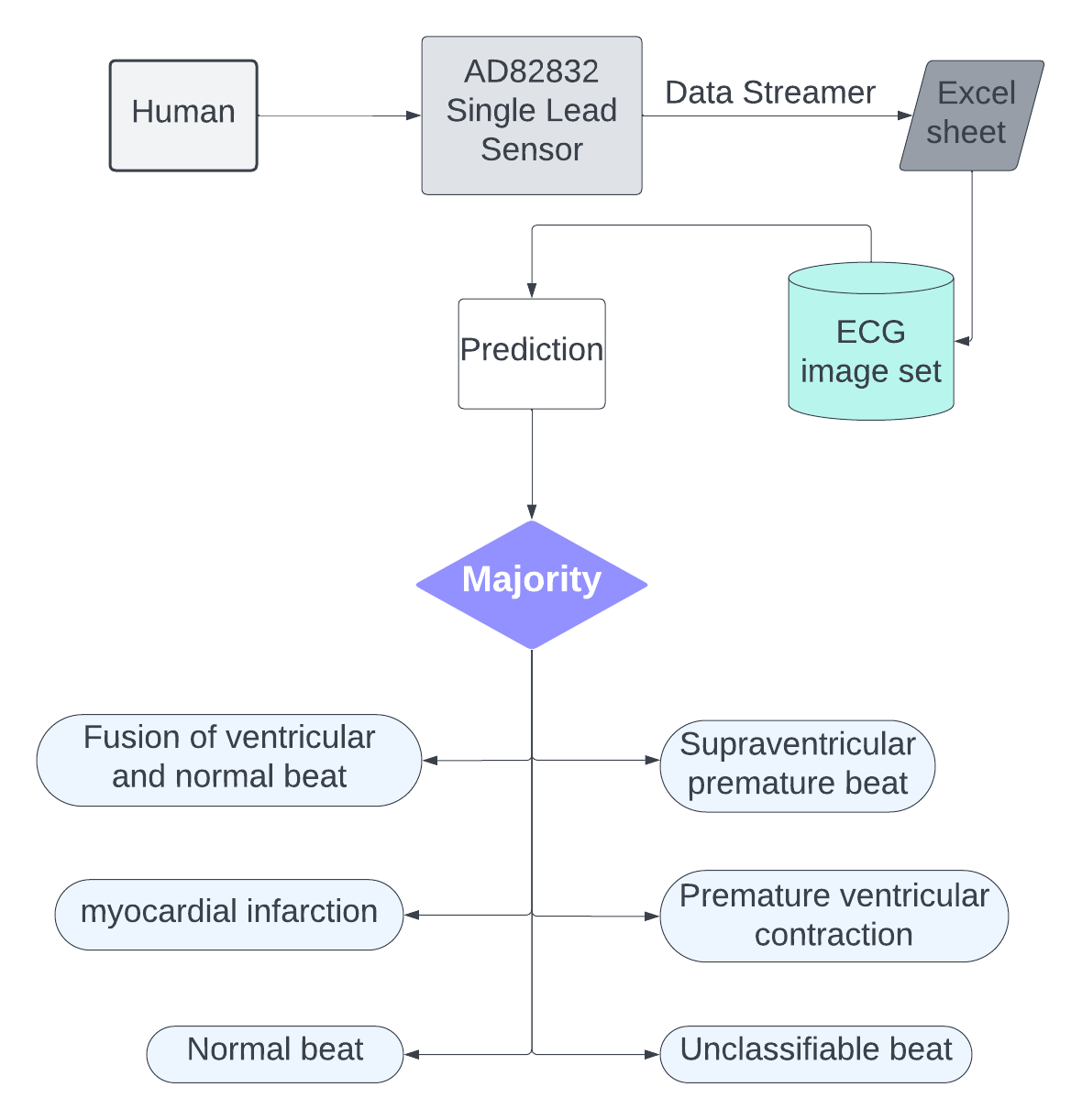
The main objective of the project is to make the Ecg Cost effective and fast

as compared to conventional ECG.

1.4 Problem Statement

**“** IOT based ECG and Disease Prediction using Machine Learning ”

**1.5 Approach**

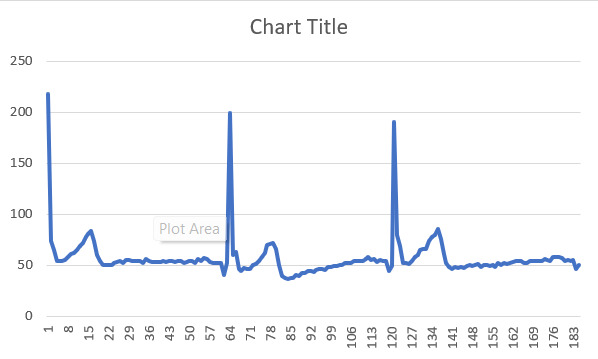
****

1. **Vital data collection using IOT:**

Collecting ECG data of humans using AD8232 single lead ECG sensor. It is a specialized integrated circuit designed for biomedical applications, particularly for measuring and processing the electrical activity of the heart. It is specifically developed to capture and amplify the signal from a single electrocardiogram lead.

Then we make changes in the arduino code to detect the RR peaks and detect when the leads off to increase the accuracy of the ecg data of the human. Also calculate the time-based heart rate in order to predict the disease more accurately.

the output of the data collected through sensor is:



The dataset contains multiple class of

N: Normal beat

S: Supraventricular premature beat

V: Premature ventricular contraction

F: Fusion of ventricular and normal beat

Q: Unclassifiable beat

M: myocardial infarction

1. **Prediction using machine learning:**

We have chosen a machine learning model which was based on neural network. The dataset was splitted into training and testing subsets which consisted of 30000 data points. The machine learning model was trained and tested on this dataset. The output of the machine learning model was either healthy or unhealthy patients. Then the data points were converted in the form of images which eliminated the limitation of the number of leads. Then to use the images for ECG classification we used another machine learning model called resnet50, which was trained on a dataset of images. The images were made from the MIT BIH dataset. The data points of this dataset were sampled at 125 Hz. The entire dataset of images was splitted into two parts test and train. the model was trained from training subset images. Model was validated from test subset of images.

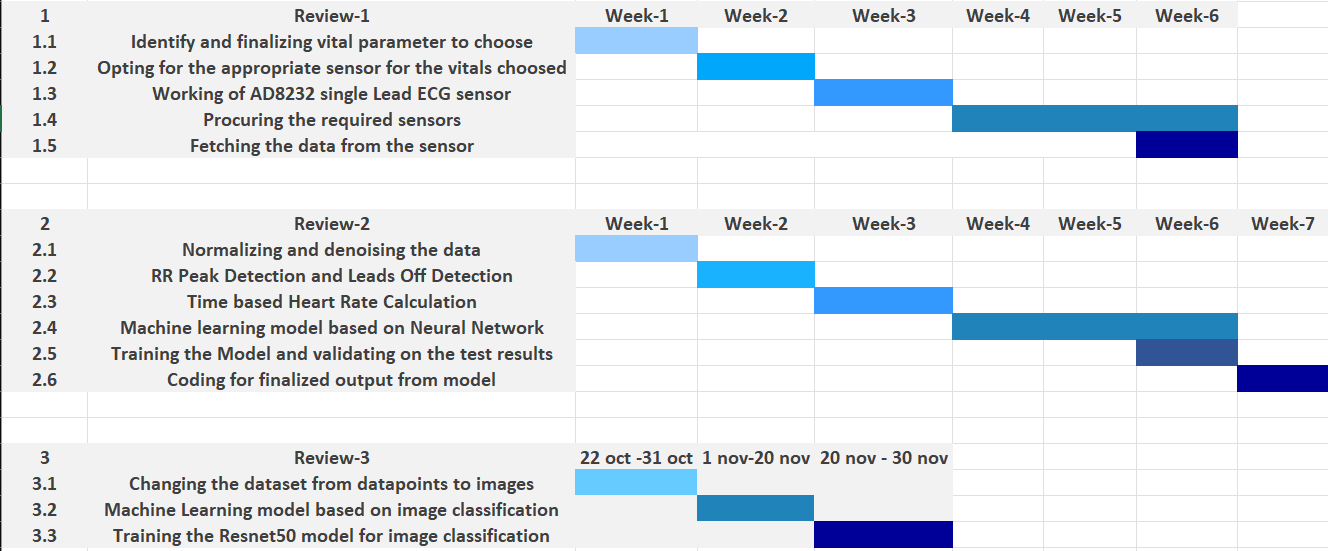
We fetched the data from the AD8232 sensor and then on the interval of 100 we made the images and fed into the model for the classification. Finally for every image the classification was counted. Classification having maximum count was given as final output.

**1.6 Scope of the project**

Scope of the project is to classify the ECG heart beat from image data and

the classification having 6 possible outcomes

**1.7 Gantt Chart**

****

**2 Literature review**

**Method of diagnosing heart disease based on deep learning ECG sign**

The paper proposes a novel method to predict heart diseases from ECG signals using cardiology knowledge, signal processing methods, and deep learning models. The authors highlight the importance of feature extraction and wavelet transformation in analyzing the time-frequency characteristics of ECG signals. They also mention the advantages of using deep convolutional neural networks for classification tasks. The paper references several related studies on automated detection of arrhythmias, ECG-based atrial fibrillation detection, ECG analysis, and wavelet distance measure for person identification using electrocardiograms. The authors emphasize the simplicity, speed, and state-of-the-art performance of their proposed system.

**Parameters for R wave extraction**

The R wave extraction process involves several parameters. First, a band-pass filter is used to reduce noise interference, with a passband of approximately 5-15 Hz. The filter is designed using a specialized technique and has a transfer function. Next, a derivative approximation filter is applied, followed by an amplitude squaring process. The signal then passes through a moving-window integrator, with the width of the integration window set to N=30 samples. Finally, adaptive thresholds are used to discriminate the locations of the R waves, taking advantage of the improved signal-to-noise ratio provided by the band-pass filter.

**Parameters for feature wave interception**

To extract feature waves from ECG signals, a fixed number of R waves within a certain range is considered desirable. If the total number of R waves is more or less than this range, the signal is considered noise. To reduce computational complexity and improve classification accuracy, feature waves containing four cardiac cycles are intercepted from the middle of the ECG signals.

**Parameters for wavelet transformation**

Wavelet transformation is used to obtain the time-frequency characteristics of signals. The wavelet transform formula involves scale factor (a) and displacement factor (b). In this method, the db4 wavelet basis function is used. The integral function is discretized and summed to achieve integration in programming.

**Parameters for classification via ResNet**

For classification purposes, the ResNet-34 architecture is selected as the classification network. ResNet is a deep convolutional neural network that can extract deep features from images. It is chosen for its capability to achieve good classification results based on the obvious differences in time-frequency diagrams of different classes.

**Experimental results**

The proposed method for predicting heart diseases from ECG signals shows promising results. The system achieves high accuracy in detecting arrhythmias, with precision and recall rates above 90% for most classes. The system also outperforms other participants in terms of F1 score, demonstrating its robustness and state-of-the-art performance.

**Accuracy and Other Parameters**

The document provides information on the accuracy and other parameters of the ECG classification system. According to Table 2, the precision and recall for each class are as follows:

* Normal: Precision - 99.32%, Recall - 96.67%
* Atrial fibrillation (AF): Precision - 97.83%, Recall - 90.00%
* Other rhythm: Precision - 98.44%, Recall - 90.00%
* Noise: Precision - 68.18%, Recall - 100%

When noise signals are properly removed from the ECG signals, the classification accuracy can reach more than 98.6% on average. However, it does not provide the overall accuracy of the system

**Automatic ECG Diagnosis Using Convolutional Neural Network**

In this study, the authors proposed an automated heart disease recognition technique based on recent and innovative CNN networks. The proposed technique achieved high accuracy and had low complexity of implementation. The method utilized deep learning to capture the typical characteristics of heart disease in the ECG signal domain.

The model used in the study is a 5-layer Convolutional Neural Network (CNN). This model was implemented for classifying ECG signals. The study compared the performance of this model with other methods and reported an accuracy of 98.33%. The model structure was designed to be robust against the "vanishing gradients"

Training and Validation Accuracy: According to the given document, the training and validation accuracy stabilized at 100% after 100 epochs. This indicates a good percentage of accuracy in the classification of the three classes described in the document.

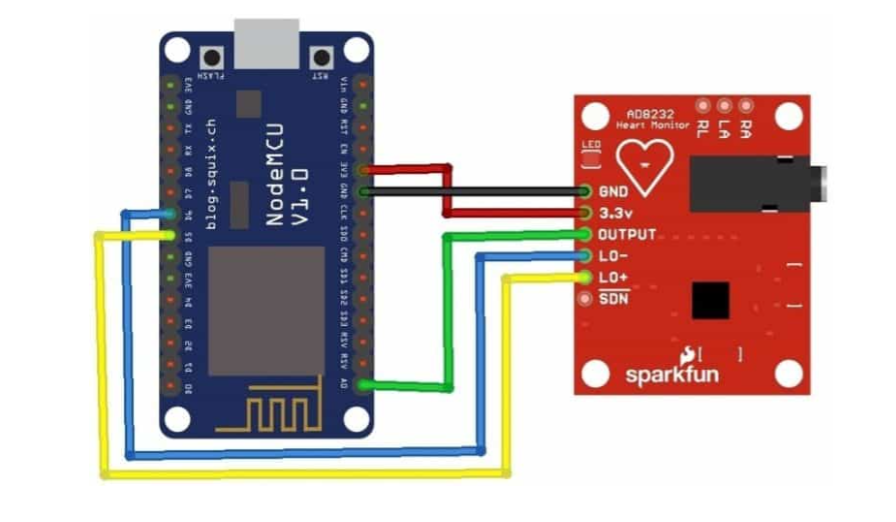
Testing Set Accuracy: The accuracy obtained with the "testing set" was assessed and the relative confusion matrix showed an average classification accuracy level of 98.33%. This indicates that the CNN network performed well on ECG sequences external to the training dataset.

Statistical Parameters: The statistical parameters used to evaluate the proposed method include sensitivity (TPR), specificity (TNR), Fall-Out (FPR), and F1 score. These parameters were calculated based on the results obtained from the confusion matrix. The TPR, TNR, FPR, FDR, and F1 score values for each class are reported in Table 3 of the document.

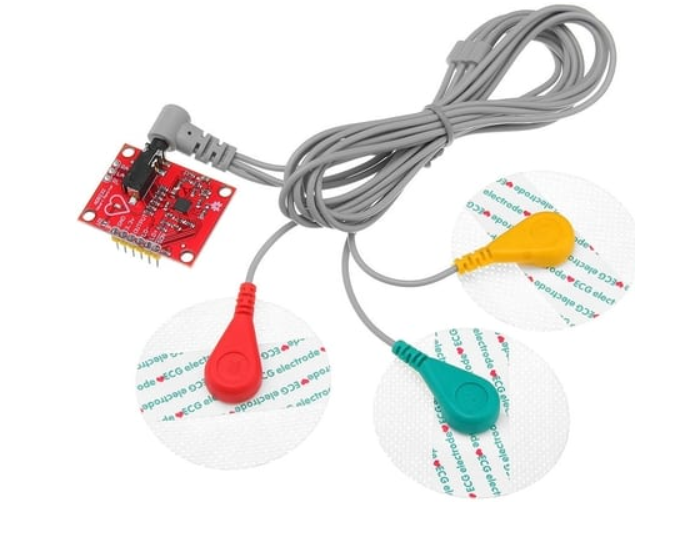
Cross-Validation Analysis: The document mentions the use of K-fold cross-validation to obtain reliable estimates of the generalization error of the model. The training dataset was divided into ten parts, and during the ten iterations, nine parts were used for training, and one part was used as a test set for model evaluation. The average accuracy and standard deviation for the model used in this study were 96.8 ± 1.2%.

Comparison with Other Methods: In terms of accuracy and classification performance, the method proposed in the document outperformed some other methods mentioned in the document. The authors claimed an average accuracy of 96.56% and 87.66% for different methods, while the average accuracy of the proposed method was 98.1%. However, there were other methods with comparable performances, but they used more hidden layers

**3 Hardware Design**

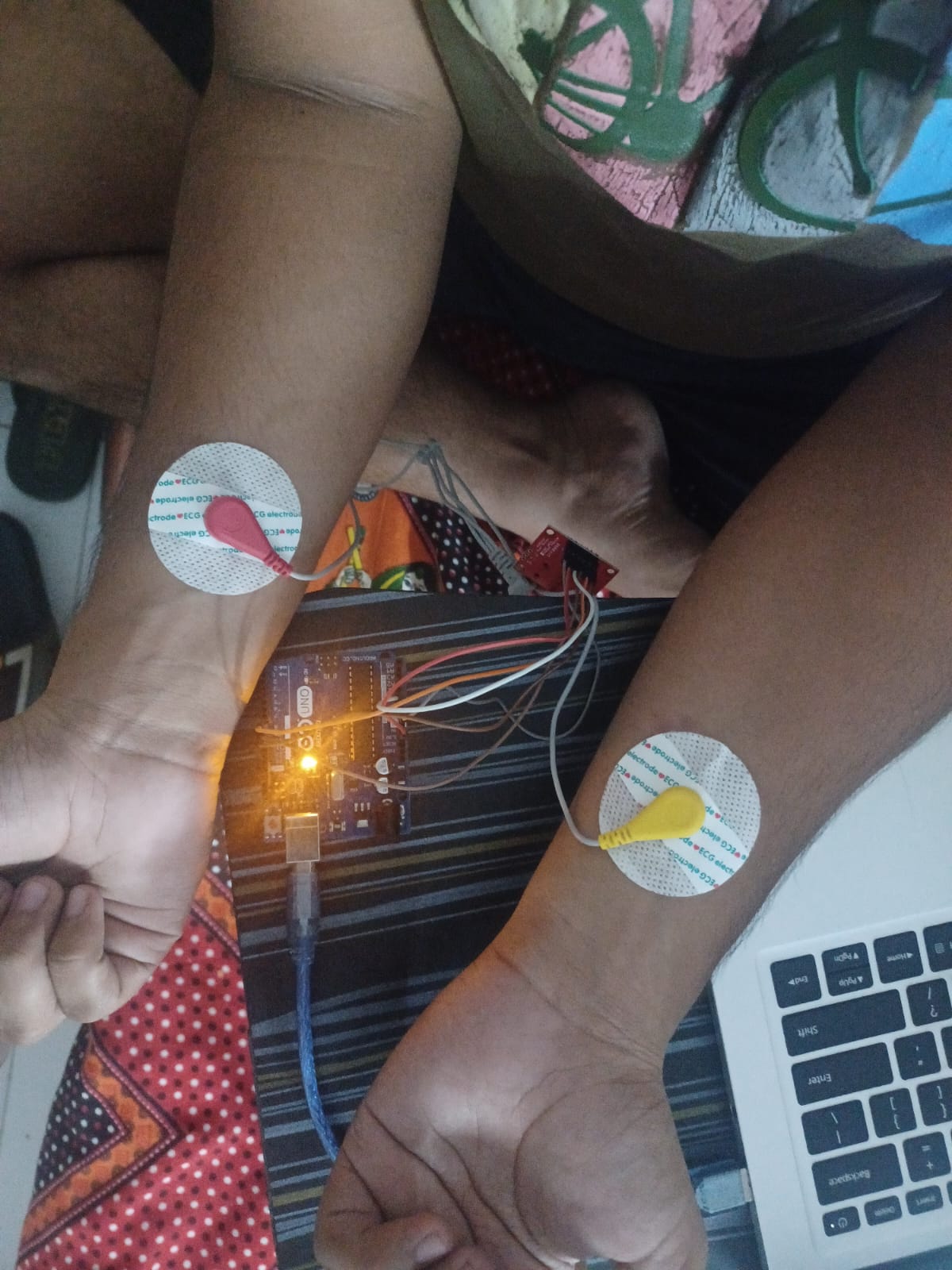
****

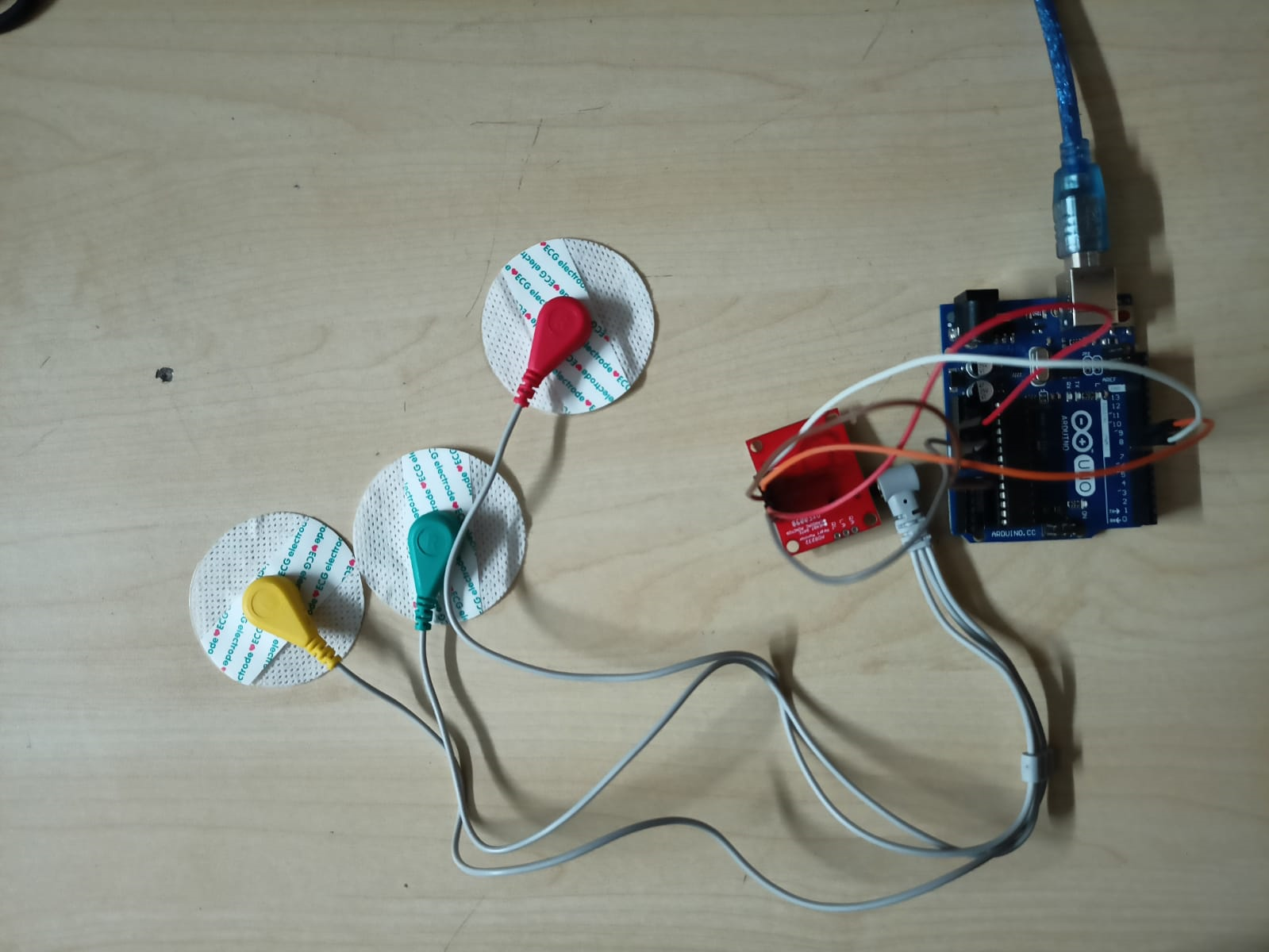
**Circuit Diagram**

****

**AD8232 Heartbeat Sensor**

**Implementation**

****

****

AD8232 sensor was used for fetching the data of patient, this sensor consisted of one lead and 3 electrodes but in hospitals the machines have 12 leads for taking the ECG. So this sensor takes analog signal and passess it on to microcontroller, then through analog pin signal is taken by microcontroller which converts it into digital form. The data points of voltage with respect to time are captured in excel file. this excel file is used for further processing.

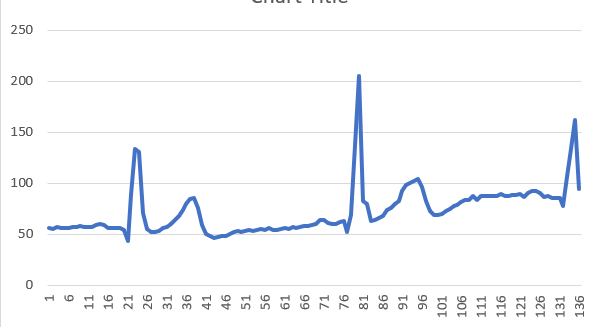
**4 Software Design**

All the processing after fetching the data was part of software design process. The fetched data was collected from arduino ide with a specified baud rate. Then the data points were cleaned and denoised using python. Python and Arduino C was mainly used for the processing part. Machine learning was done in python only. Many libraries of python were used for the implementation of Machine learning model and for executing various tasks.

* Arduino C code was dumped into the microcontroller for fetching the data into excel sheet. The code implements a basic algorithm to process ECG data from a sensor, detect RR peaks in the ECG signal, calculate heart rate, and estimate heart rate variability using an Arduino. And with the help of a data streamer, the data was stored into the excel file using the port address.
* Model 1 was based on neural network, this model was trained and validated in python. The model was named the Sequential model. It was feeded with a total of 30,000 datapoints. The model training details are given Optimizer= Adam, Loss= Cross Entropy Loss, Number of Epochs= 150, Regularization parameter = 0.001, Drop out parameter = 0.3. This model was first trained on without dropout and regularization and then with dropout and regularization. With increasing epochs, accuracy also increased. In total 150 epochs were used for training the model. Finally the prediction code was written based on which model gave the output whether patient is healthy or unhealthy.
* Model 2 was based on image classification. To eliminate the limitation of number of leads we have used image classification model. Model named Resnet 50 was used for this purpose. The model was trained on the images. This images were made from MIT BIH dataset, this dataset was sampled at 125 hz and then many images were formed on the specified interval. In our case the interval is 100, so that in one image we can inculcate 1 waveform. The model output had 5 classifications N,S,V,F,Q,M. Normal beat, Supraventricular premature beat, Premature ventricular contraction, Fusion of ventricular and normal beat, Unclassifiable beat, myocardial infarction. The model was trained on this images based on this 5 classification. With increasing number of epochs accuracy of the model also increased and loss of the loss function decreased. For the prediction of the heart disease we took a interval of datapoints and in that interval, its graph was plotted. Those graphs were converted into images and used for the purpose of classification. Each and every image classification was counted and the maximum time a disease is classified was given as output to the patient

**5 Experimental Results**

ECG data plot using AD8232:

****

**Model Accuracy:**

Test Loss: 0.1637

Test Accuracy: 94.6875%

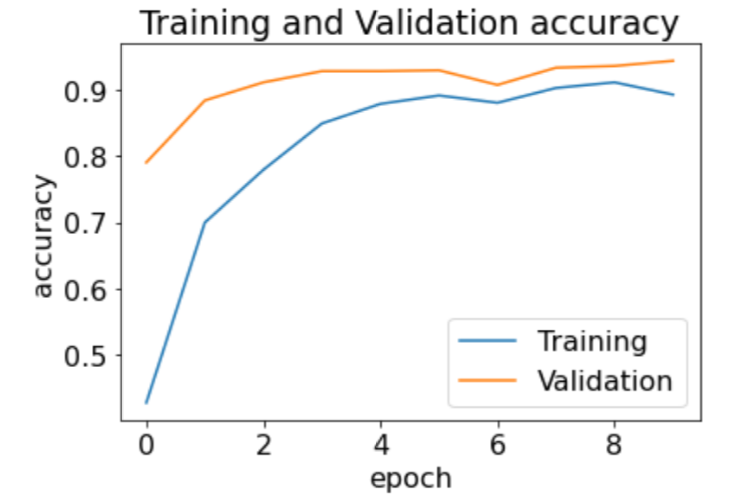
Test Precision: 94.6764%

Test Recall: 94.4792%

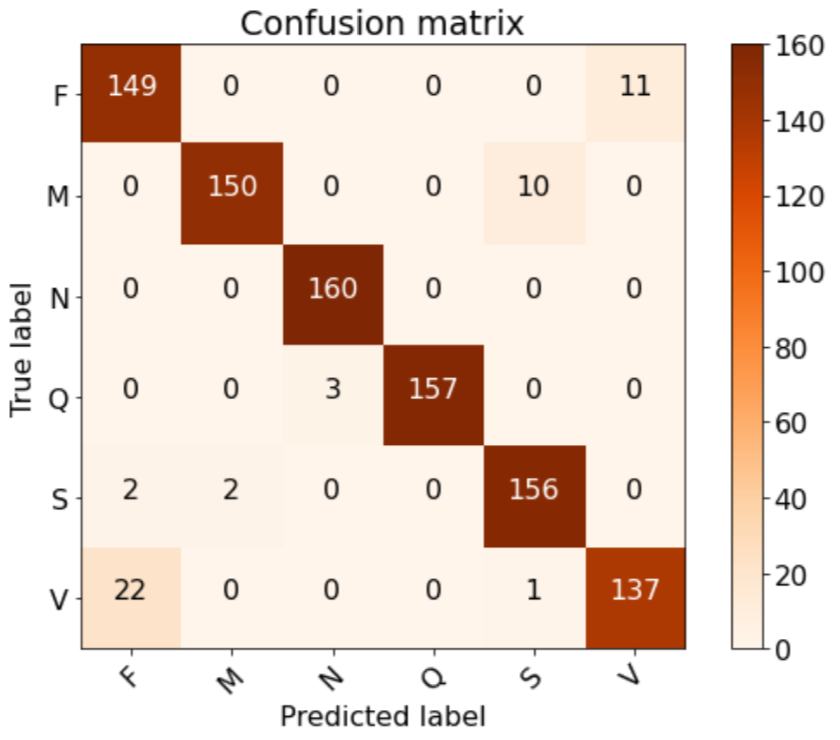
Test AUC: 0.9970

****

Training and validation losses

****

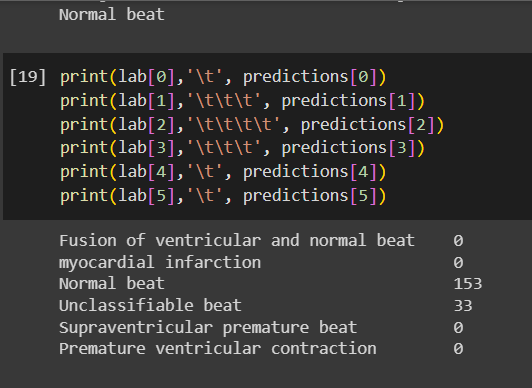
Training and validation accuracy

****

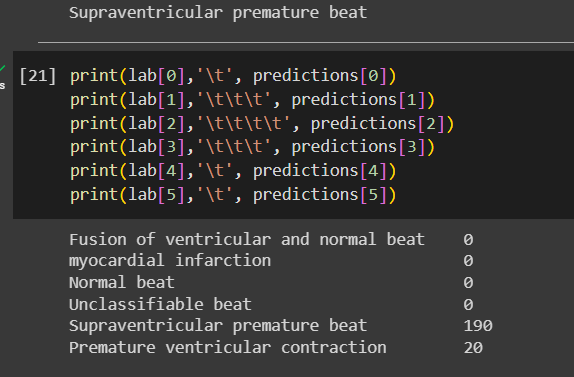
Confusion matrix without normalization

Prediction results:

1. for normal beat:



1. Supraventricular premature beat



**6 Conclusion and Future Scope**

The increasing heart patients is a global cause and early detection becomes more important for humanity therefore this project has shown the potential of early detection of heart disease by classifying the ECG data into 6 classifications namely P,S,Q,V,M,F. The output of the model can be trusted with accuracy of 94 percent.

Looking ahead, the fusion of IoT and machine learning in the realm of healthcare presents an expansive future scope, especially concerning the escalating prevalence of heart-related ailments worldwide. The pressing need for early detection and intervention in heart diseases underscores the significance of innovative approaches like the one demonstrated in this project. By classifying ECG data into precise categories such as P, S, Q, V, M, F with a commendable accuracy of 94 percent, this initiative showcases the potential of technology to revolutionize cardiac care.

Future advancements in this field could lead to the development of wearable IoT devices capable of continuously monitoring and analyzing cardiac signals in real-time. These devices could seamlessly integrate into daily life, offering individuals the ability to monitor their heart health proactively. Machine learning algorithms embedded within these devices could swiftly detect anomalies or patterns indicative of cardiovascular issues, alerting both patients and healthcare providers for timely intervention.

Moreover, the amalgamation of robust data analytics, enhanced algorithms, and the growing pool of medical data will further refine predictive models. This refinement will not only improve accuracy but also facilitate personalized medicine by tailoring treatments based on an individual's unique cardiac profile, ultimately enhancing patient outcomes and reducing healthcare burdens.

As this technology progresses, collaborations between tech innovators, healthcare professionals, and regulatory bodies will be crucial to ensure ethical implementation, data privacy, and widespread accessibility. The future holds immense promise, where IoT-driven predictive models for early heart disease detection can significantly contribute to global healthcare strategies, potentially mitigating the adverse impact of cardiac ailments on humanity.

**References**

1. Avanzato, Roberta, and Francesco Beritelli. “Automatic ECG Diagnosis Using Convolutional Neural Network.” *Electronics*, vol. 9, no. 6, 8 June 2020, p. 951, https://doi.org/10.3390/electronics9060951.
2. Zhang, Jie, et al. *Method of Diagnosing Heart Disease Based on Deep Learning ECG Sign*.
3. Zaheer, Usman, et al. *ECG BASED RECORDING to SHOW REAL TIME CARDIAC ACTIVITY Contents*.
4. GitHub. “GitHub.” *GitHub*, 2023, github.com/.
5. Kaggle. “Kaggle: Your Home for Data Science.” *Kaggle.com*, 2022, www.kaggle.com/.
6. “How to Electronics | Engineering Projects & Tutorials.” *How to Electronics*, how2electronics.com/.
7. “Hindawi.” *Hindawi.com*, 2019, www.hindawi.com/.
8. “Circuit Digest.” *Circuitdigest.com*, 2019, circuitdigest.com/..

**Appendix**

Code for Prediciton:

from keras.models import load\_model

from keras.preprocessing.image import load\_img,img\_to\_array

import matplotlib.pyplot as plt

import matplotlib.image as mpimg

import numpy as np

import pandas as pd

import os

model1 = load\_model('./Resnet.h5',compile=False)

lab = {0:'Fusion of ventricular and normal beat',1:'myocardial infarction',2:'Normal beat',3:'Unclassifiable beat',4:'Supraventricular premature beat',5:'Premature ventricular contraction'}

def output(location):

img=load\_img(location,target\_size=(224,224,3))

img=img\_to\_array(img)

img=img/255

img=np.expand\_dims(img,[0])

answer=model1.predict(img)

y\_class = answer.argmax(axis=-1)

y = " ".join(str(x) for x in y\_class)

y = int(y)

return y

# Create a folder to save the plots

folder\_name = 'interval\_plots'

os.makedirs(folder\_name, exist\_ok=True) # Create the folder if it doesn't exist

folder\_path = 'interval\_plots' # Replace this with the path to your folder

# List all files in the folder

files = os.listdir(folder\_path)

# Iterate through each file and delete them

for file\_name in files:

file\_path = os.path.join(folder\_path, file\_name)

if os.path.isfile(file\_path): # Check if it's a file (not a directory)

os.remove(file\_path)

print(f"Deleted: {file\_path}")

print("All files inside the folder have been deleted.")

# Load the CSV file

file\_path = './record3.csv' # Replace 'your\_file.csv' with your file's path

data = pd.read\_csv(file\_path)

# Assuming the CSV file has a column named 'values' that you want to plot

column\_name = 'Data’'

# Plotting in intervals of 100

interval = 40

num\_rows = len(data)

for i in range(0,2000, interval):

subset = data.iloc[i:i + interval] # Extract a subset of data for the interval

plt.plot(subset[column\_name], label=f'Interval {i+1}-{i+interval if i+interval<num\_rows else num\_rows}')

fig = plt.gcf()

fig.set\_size\_inches(224 / fig.dpi, 224 / fig.dpi)

# Save the plot in the folder

plt.savefig(f'{folder\_name}/plot\_interval\_{i+1}\_{i+interval if i+interval<num\_rows else num\_rows}.png')

plt.close() # Close the plot to avoid overlapping

print("Plots saved in the folder:", folder\_name)

# Path to the folder containing images

folder\_path = 'interval\_plots/'

# Get a list of all files in the folder

file\_list = os.listdir(folder\_path)

# Filter out only the image files (assuming they have .png extension)

image\_files = [file for file in file\_list if file.endswith('.png')]

# List to store the predictions

predictions = {0:1,1:0,2:0,3:0,4:0,5:0}

# Loop through each image file in the folder

for img\_file in image\_files:

# Path to each image file

img\_path = os.path.join(folder\_path, img\_file)

# Assuming you have an output function that processes the image

result = output(img\_path)

# Ensure the result is within the range of 0 to 5

result = max(0, min(result, 5))

# Append the result to the predictions list

predictions[result]+=1;

maj = -1;

count = 0;

for i in range(6):

if(predictions[i]>count):

count = predictions[i];

maj = i;

print(lab[maj])

predictions

Arduino C code:

/\*

\* VARIABLES

\* count: variable to hold count of rr peaks detected in 10 seconds

\* flag: variable that prevents multiple rr peak detections in a single heartbeat

\* hr: HeartRate (initialized to 72)

\* hrv: Heart Rate variability (takes 10-15 seconds to stabilize)

\* instance1: instance when the heartbeat is first detected

\* interval: interval between the second heartbeat and the first heartbeat

\* timer: variable to hold the time after which hr is calculated

\* value: raw sensor value of the output pin

\*/

long instance1 = 0, timer; // Initialize two long integer variables instance1 and timer.

double hrv = 0, hr = 72, interval = 0; // Initialize double variables hrv, hr, and interval.

int value = 0, count = 0; // Initialize two integer variables value and count.

bool flag = 0; // Initialize a boolean variable flag.

// Define constants for pin numbers and threshold values.

#define shutdown\_pin 10 // Digital pin 10 is used for shutdown control.

#define threshold 100 // Threshold value to identify R peaks.

#define timer\_value 10000 // Timer value in milliseconds for calculating heart rate.

void setup() {

Serial.begin(9600); // Initialize serial communication with a baud rate of 9600.

pinMode(8, INPUT); // Set digital pin 8 as an input for leads off detection LO+.

pinMode(9, INPUT); // Set digital pin 9 as an input for leads off detection LO-.

}

void loop()

{

if ((digitalRead(8) == 1) || (digitalRead(9) == 1))

{

// If either of the leads is off (digital pin 8 or 9 is high), perform the following:

Serial.println("leads off!"); // Print "leads off!" to the serial monitor.

digitalWrite(shutdown\_pin, LOW); // Set the shutdown\_pin to LOW (standby mode).

instance1 = micros(); // Record the current time in microseconds as instance1.

timer = millis(); // Record the current time in milliseconds as timer.

}

else

{

// If both leads are on, perform the following:

digitalWrite(shutdown\_pin, HIGH); // Set the shutdown\_pin to HIGH (normal mode).

value = analogRead(A0); // Read an analog value from pin A0 and store it in the value variable.

value = map(value, 250, 400, 0, 100); // Map the value from a range of 250-400 to 0-100.

if ((value > threshold) && (!flag))

{

// If the mapped value is greater than the threshold and the flag is not set, do the following:

count++; // Increment the count variable.

Serial.println("in"); // Print "in" to the serial monitor.

flag = 1; // Set the flag to 1 to prevent multiple detections in a single heartbeat.

interval = micros() - instance1; // Calculate the RR interval.

instance1 = micros(); // Update the instance1 to the current time.

}

else if ((value < threshold))

{

// If the value is less than the threshold, reset the flag to 0.

flag = 0;

}

if ((millis() - timer) > 10000)

{

// If more than 10 seconds have elapsed, do the following:

hr = count \* 6; // Calculate heart rate based on the count of RR peaks.

timer = millis(); // Reset the timer.

count = 0; // Reset the count.

}

hrv = hr / 60 - interval / 1000000; // Calculate heart rate variability.

Serial.print(hr); // Print heart rate to the serial monitor.

Serial.print(",");

Serial.print(hrv); // Print heart rate variability to the serial monitor.

Serial.print(",");

Serial.println(value); // Print the mapped sensor value to the serial monitor.

delay(1); // Delay for 1 millisecond.

}

}

//This code appears to be reading data from a heart rate sensor (ECG values), detecting RR peaks in the ECG signal, and calculating heart rate and heart rate variability. It uses an Arduino or similar microcontroller to perform these tasks.

* Use font Times New Roman and Size 14 for chapter title and subtitles.
* Use Content font size 12.
* Page numbers are to be given to all the pages
* You can add following in the Appendix
  + Bill of hardware materials used in the project
  + Data Sheet of essential hardware components you have used in the project
  + Snippets of most important part of your program/s