Minor Project (2EC703)
REVIEW III

Department of Electronics and Communication Engineering
Nirma University
2, December 2023

IOT Based ECG Data Collection and Classification

Submitted By:
Dishank Garg(20BEC030)
Jainam Jain(20BEC043)

Project Guide :
Dr. M. A. Upadhyay

Outline

01 About Project

02 Approach

03 Sensor and circuit

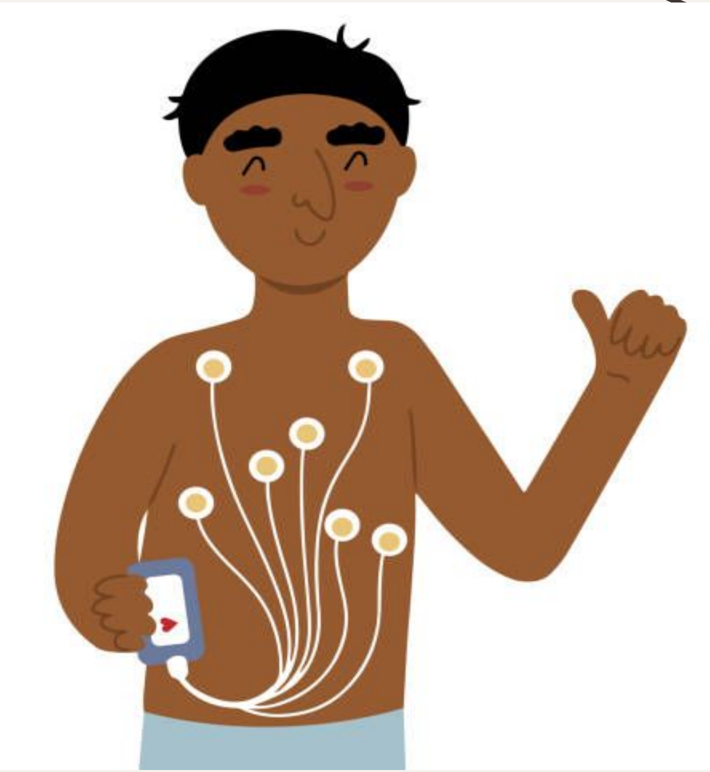
04 Dataset and Model

05 Results

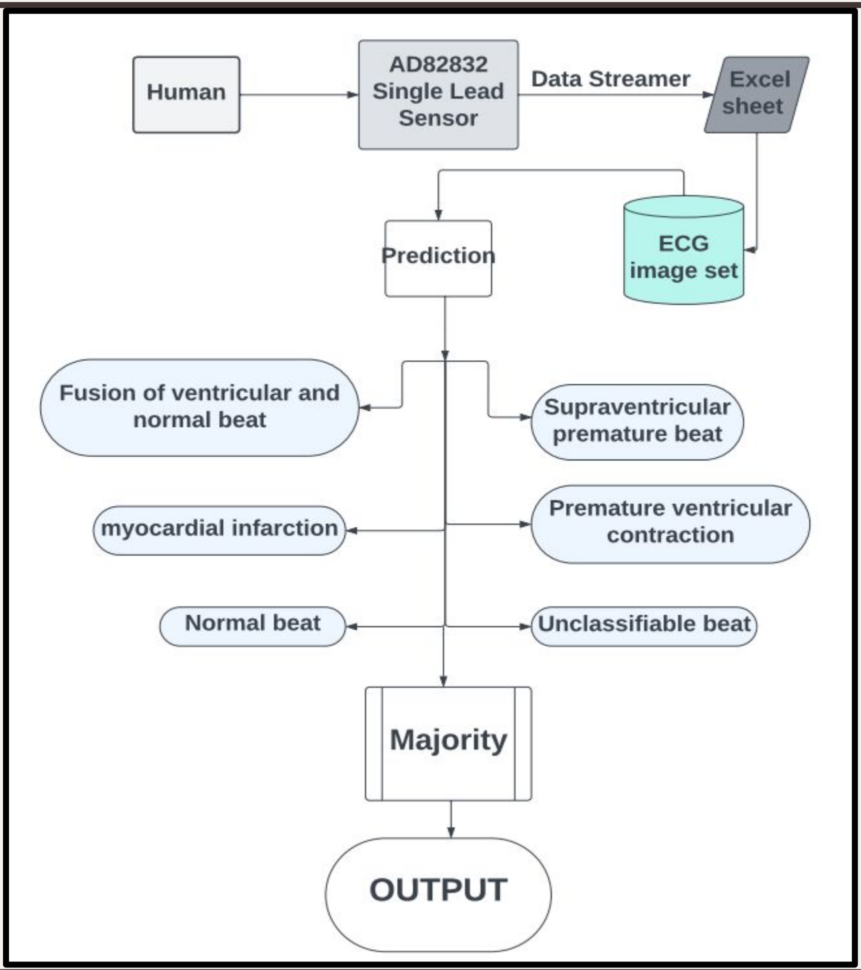
06 Future Scope and References

About Project

- This project enables individuals to monitor their ECG and assess their heart condition, especially in situations where immediate access to professional medical facilities is unavailable.

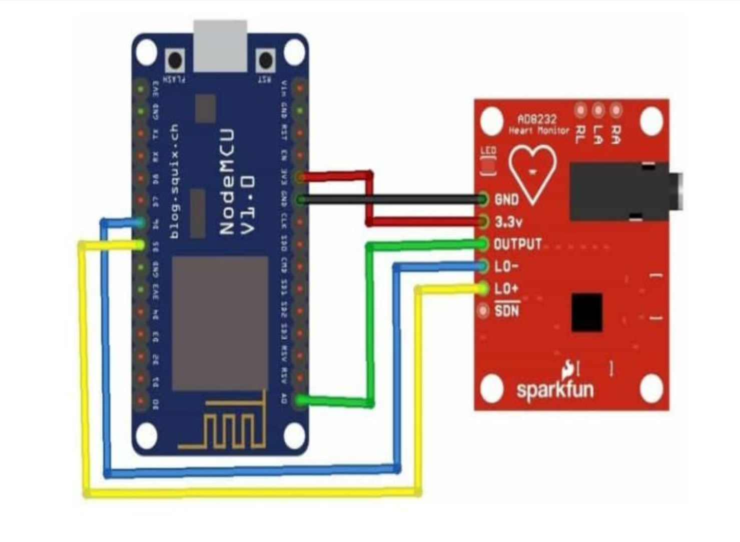


Approach

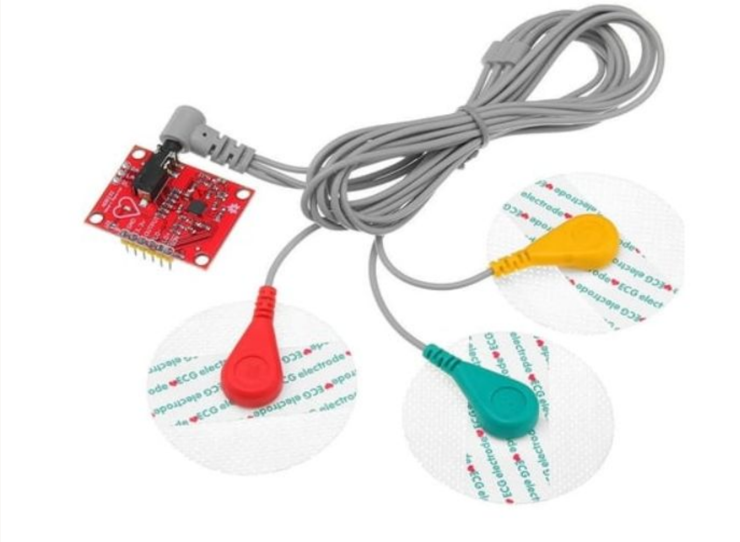


(Flow chart)

Circuit Diagram and Sensor

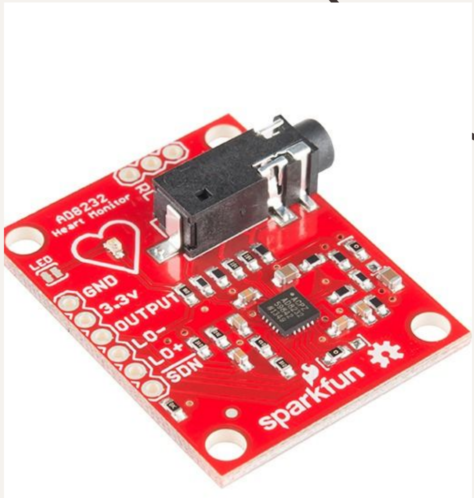


NodeMCU connected with AD8232



AD8232 ECG sensor, electrodes with 3.5mm jack cable

- GND - This pin is connected to the ground or 0V reference.
- 3.3V - This pin is used to provide a 3.3V power supply to the sensor.
- Output - This pin provides the processed ECG waveform.
- Lo+ ->(Lead Off Positive) monitors the positive electrode's contact
- Lo- -> (Lead Off Negative) monitors the negative electrode's contact
- Both detect lead-off conditions by changes in impedance when electrodes lose skin contact.
- SDN ->Shut Down Pin , when LOW keep the circuit steady

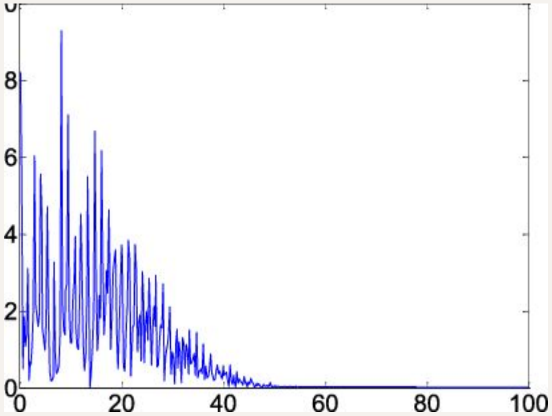


1. RR Peak Detection:

```
if ((value > threshold) && (!flag)){
        count++;
        flag = 1;
        interval = micros() - instance1;
        instance1 = micros();
    }
```

2. Leads Off Detection:

```
if ((digitalRead(8) == 1) || (digitalRead(9) == 1))
{
     Serial.println("leads off!");
     digitalWrite(shutdown_pin, LOW);
   }
```



3. Time-Based Heart Rate Calculation

Calculation:

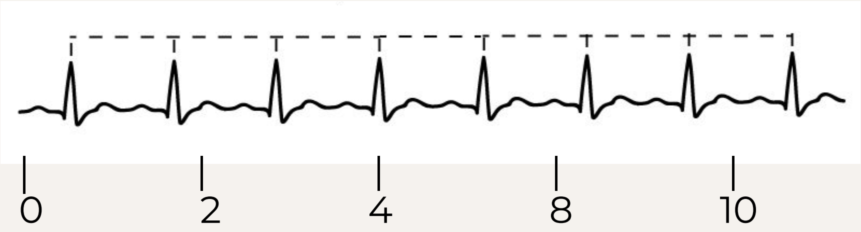
Heart Beat = 10*(Number of R peaks in 6 seconds)

Or

6*(Number of R peaks in 10 seconds)

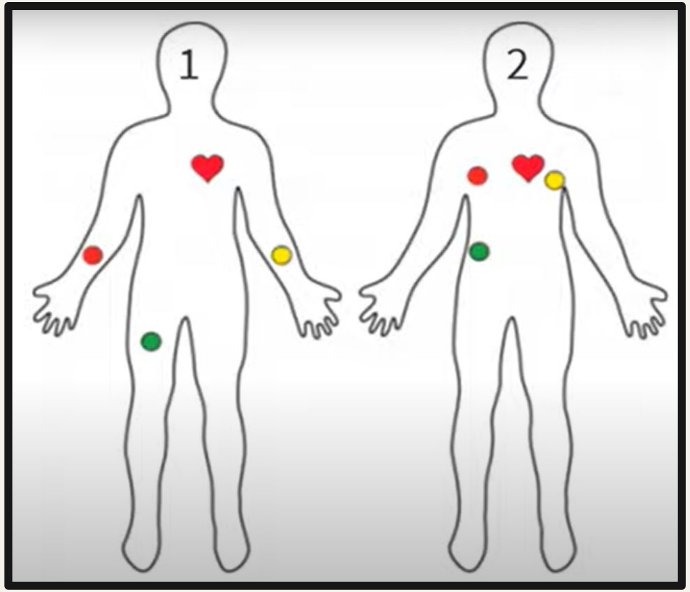
In the figure, number of peaks = 7

HV = 6*7 = 70 beats per minute

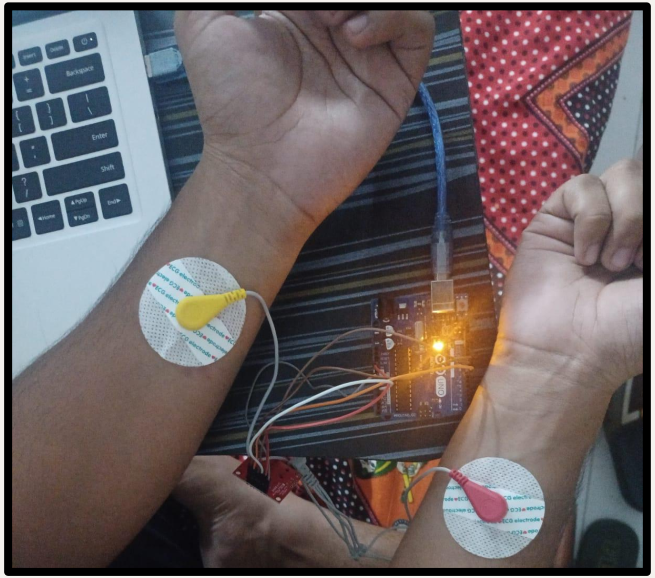



```
if ((millis() - timer) > 10000)
    {
      // If more than 10 seconds have elapsed, do the following:
      hr = count * 6;  // Calculate heart rate based on the count of RR peaks.
      timer = millis();  // Reset the timer.
      count = 0;  // Reset the count.
    }
```

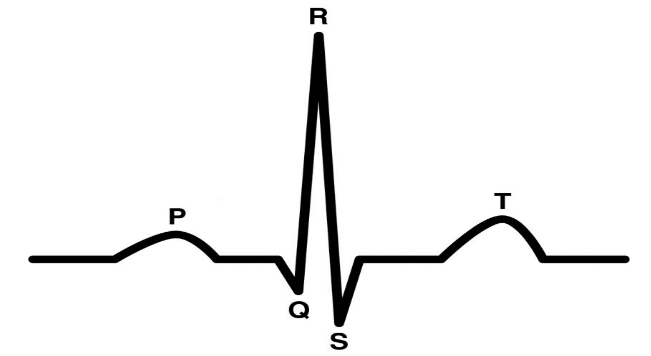
Configuration Used:



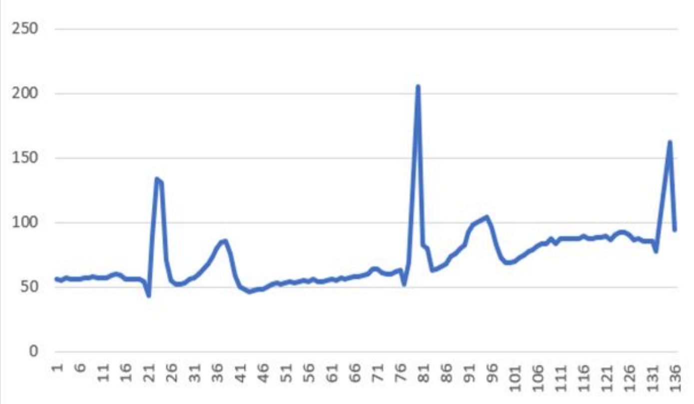
One Lead, ECG

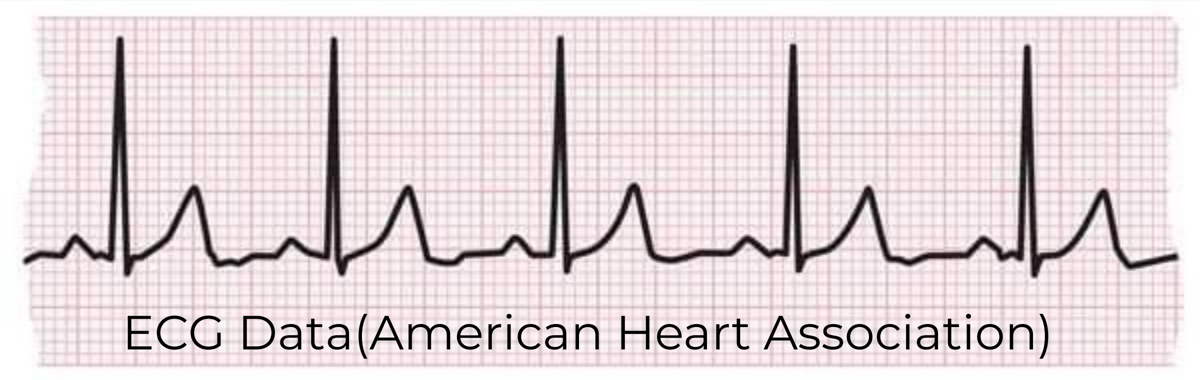


Normal ECG



Fetched Data:





ECG Data(American Heart Association)

Dataset Details

- This dataset is composed of collections of heartbeat signals derived from famous datasets in heartbeat classification, the MIT-BIH Arrhythmia Dataset.

- The Datapoints are sampled at sampling frequency of 125 Hz.

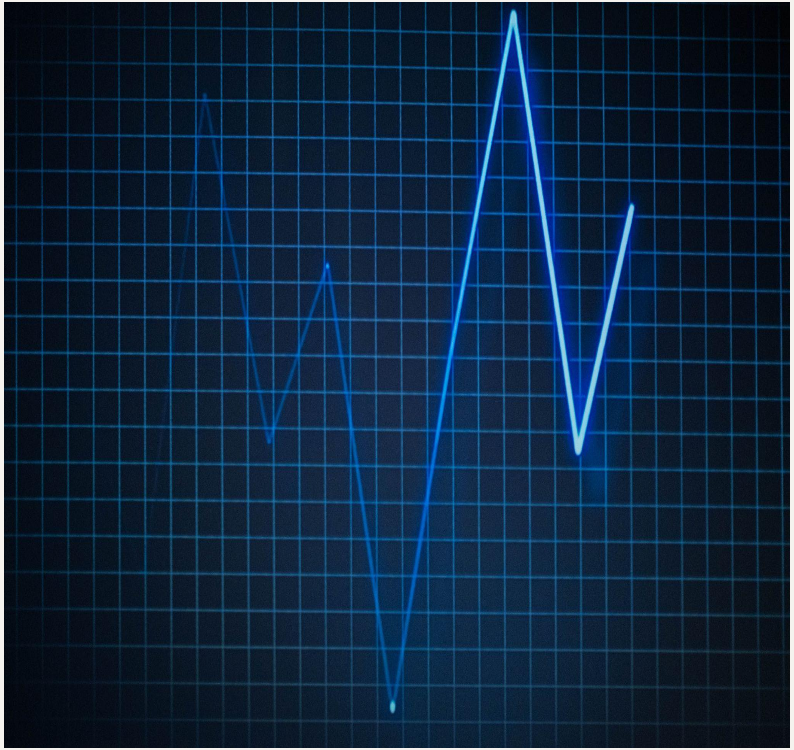
- This dataset is created by saving the each ECG arrhythmia interval into the image form.

- The output classes are as follows:
  N: Normal beat
  S: Supraventricular premature beat
  V: Premature ventricular contraction
  F: Fusion of ventricular and normal beat
  Q: Unclassifiable beat
  M: myocardial infarction

Conversion of Data Points to Image Form

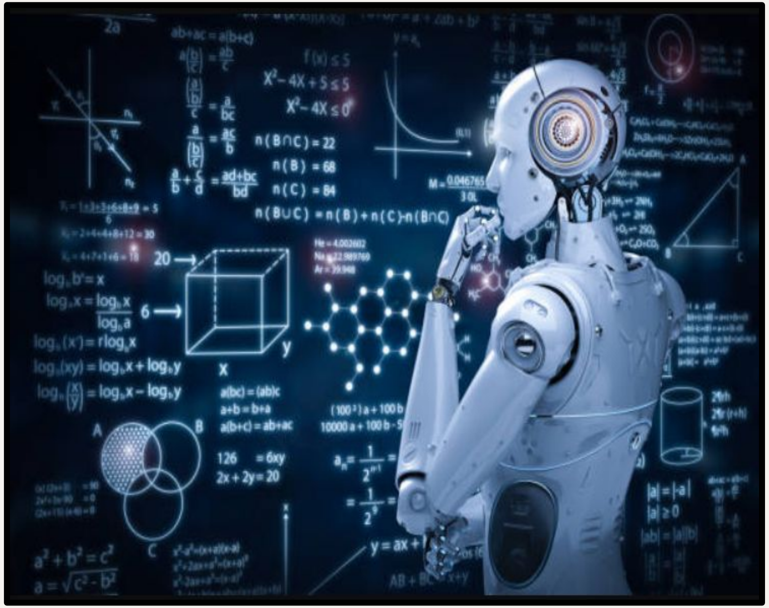
- To overcome the limitations of number of leads we have converted the ECG datapoints into the form of image.

- The images are made at a predetermined intervals of excel.



Model Details

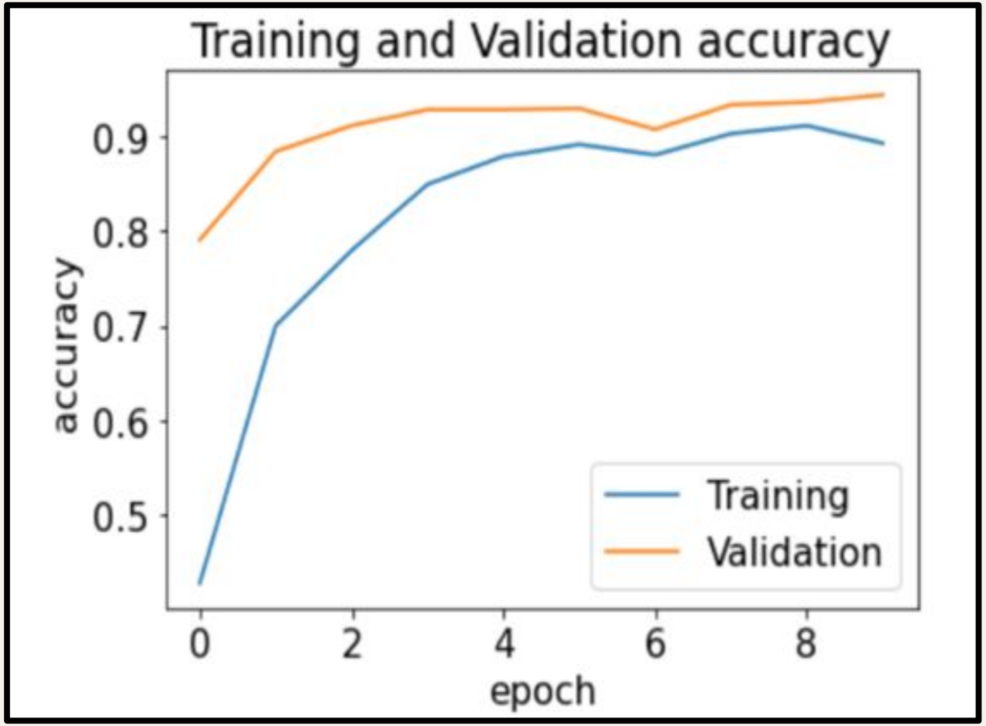
- ResNet-50 is a convolutional neural network that is 50 layers deep.
- ResNet-50 consists of multiple blocks, where each block contains several convolutional layers along with ReLU activation functions.



Training and validating the model

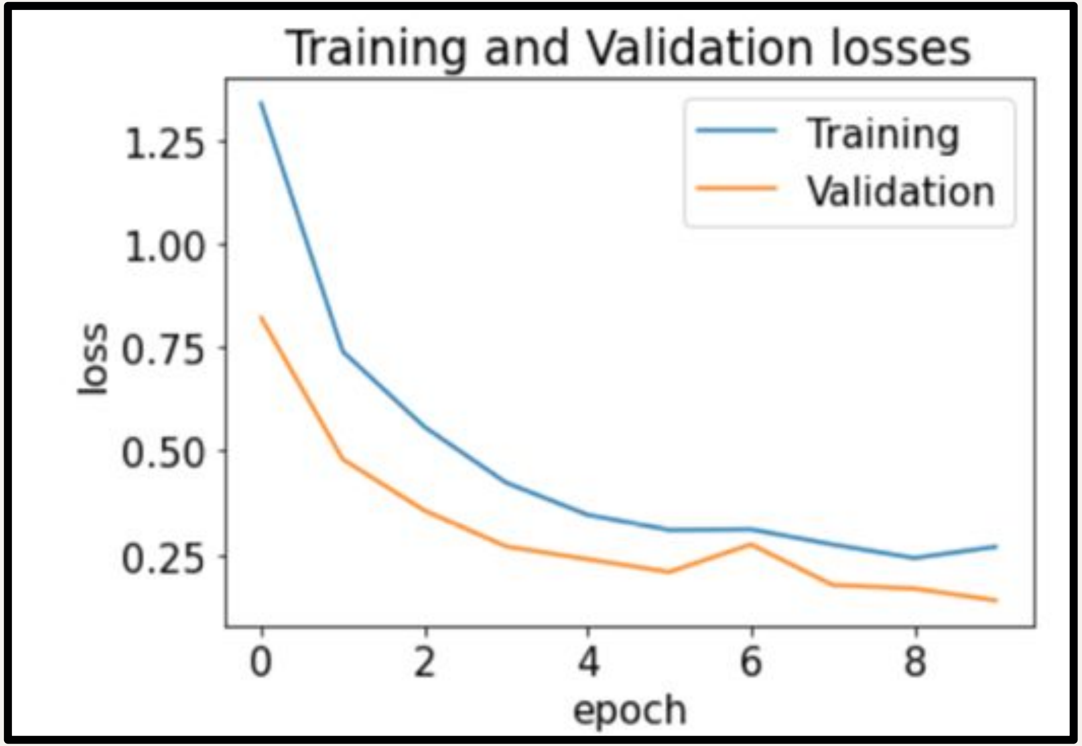
- In the image dataset all the images were labeled as per their ecg classification.
- As we have 6 classes for the output classification, 640 images from each class were taken to train the model.
- Therefore in total 3840 images were used to train the model.
- For testing, from each class 160 images were taken and the model was validated on this 960 images.

Accuracy



Training and Validation accuracy
accuracy
0.9
0.8
0.7
0.6
0.5
0
2
4
6
8
epoch
Training
Validation

Loss


Training and Validation losses
Training
Validation
1.25
1.00
0.75
0.50
0.25
loss
0
2
4
6
8
epoch

Model Accuracy:

- Test Accuracy: 94.6875%
- Test Precision: 94.6764%
- Test Recall: 94.4792%

Predicting data by feeding the values

Function to predict the image class

```
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
```

Script of Predicting the disease

1. It creates the folder to store the plot images. (Once)
2. It deletes all existing images if any, so that the prediction can be accurate.
3. We need to give path of the csv or excel file used to store the data.
4. Then the plot will be created inside the folder.
5. After running the prediction algorithm, whatever is the majority class of state will be come as output of prediction

Predictions Results

Normal beat

```
Normal beat

[19] print(lab[0],'\t', predictions[0])
     print(lab[1],'\t\t\t', predictions[1])
     print(lab[2],'\t\t\t\t', predictions[2])
     print(lab[3],'\t\t\t', predictions[3])
     print(lab[4],'\t', predictions[4])
     print(lab[5],'\t', predictions[5])

Fusion of ventricular and normal beat    0
myocardial infarction                    0
Normal beat                              153
Unclassifiable beat                      33
Supraventricular premature beat          0
Premature ventricular contraction        0
```

Supraventricular premature beat

```
Supraventricular premature beat

[21] print(lab[0],'\t', predictions[0])
     print(lab[1],'\t\t\t', predictions[1])
     print(lab[2],'\t\t\t\t', predictions[2])
     print(lab[3],'\t\t\t', predictions[3])
     print(lab[4],'\t', predictions[4])
     print(lab[5],'\t', predictions[5])

Fusion of ventricular and normal beat    0
myocardial infarction                    0
Normal beat                              0
Unclassifiable beat                      0
Supraventricular premature beat          190
Premature ventricular contraction        20
```

Future Scope

Integration with Wearable Technology:

- Advanced Sensor Integration: Further integration of ECG monitoring capabilities into wearable devices, enhancing their accuracy, battery life, and usability.
- User-Friendly Designs: Development of more user-friendly and aesthetically appealing wearable ECG devices to encourage continuous usage among consumers.

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/..