ECG arrhythmia classification using CNN

Done by : Team 26
- Yuvarani V
- Hemanjali R
- Lohit ramaraju
- Praveen Kumar
Under the Guidance of : SmartBridge

1) INTRODUCTION
 1.1) Overview
 1.2) Purpose
2) LITRATURE SURVEY
 2.1) Existing Problem
 2.2) Proposed Problem
3) THEORITICAL ANALYSIS
 3.1) Block Diagram
 3.2) Hardware / Software designing
4) EXPERIMENTAL INVESTIGATIONS
5) FLOWCHART
6) RESULT
7) ADVANTAGES AND DISADVANTAGES
8) APPLICATIONS
9) CONCLUSION
10) FUTURE SCOPE
11) BIBILOGRAPHY
 APPENDIX
 A.) Source Code

1) INTRODUCTION

1.1) 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 single arrhythmia heartbeat may not have a serious impact on life, continuous arrhythmia beats can result in fatal circumstances. For example, prolonged premature ventricular contraction (PVCs) beats occasionally turn into a ventricular tachycardia (VT) or ventricular fibrillation (VF) beats which can immediately lead to heart failure. Thus, it is important to periodically monitor the heart rhythms to manage and prevent the CVDs. Electrocardiogram (ECG) is a non-invasive medical tool that displays the rhythm and status of the heart. Therefore, automatic detection of irregular heart rhythms from ECG signals is a significant task in the field of cardiology.

In this, we firstly propose an ECG arrhythmia classification method using deep two-dimensional CNN with grayscale ECG images. By transforming one-dimensional ECG signals into two-dimensional ECG images, noise filtering and feature extraction are no longer required. This is important since some of ECG beats are ignored in noise filtering and feature extraction. In addition, training data can be enlarged by augmenting the ECG images which result in higher classification accuracy. Data augmentation is hard to be applied in previous literature since the distortion of one-dimensional ECG signal could downgrade the performance of the classifier. However, augmenting two-dimensional ECG images with different cropping methods helps the CNN model to train with different viewpoints of the single ECG images. Using ECG image as an input data of the ECG arrhythmia classification also benefits in the sense of robustness. Current ECG arrhythmia detection methods are sensitive to the noise signal since every ECG one-dimensional signal value is treated to have an equal degree of the classification. However, when the ECG signal is converted 4 Tae Joon Jun et al to the two-dimensional image, proposed CNN model can automatically ignore the noise data while extracting the relevant feature map throughout the convolutional and pooling layer. Thus, proposed CNN model can be applied to the ECG signals from the various ECG devices with different sampling rates and amplitudes while previous literature requires a different model for the different ECG devices. Furthermore, detecting ECG arrhythmia with ECG images resembles how medical experts diagnose arrhythmia since they observe an ECG graph from the patients throughout the monitor, which shows a series of ECG images. In other words, the proposed scheme can be applied to the medical robot that can monitors the ECG signals and helps the experts to identify ECG arrhythmia more precisely.

Our classification method consists the following steps: data acquisition, ECG data pre-processing, and CNN classifier. ECG signal data treated in this paper is obtained from the MIT-BIH database which is generally used as an arrhythmia database in ECG arrhythmia classification research.

1.2) Purpose

Automatic detection of irregular heart rhythms from ECG signals is a significant task in the field of cardiology.To make this task easy and reduce the time taken for this task.

2) LITRATURE SURVEY

2.1) Existing Problem

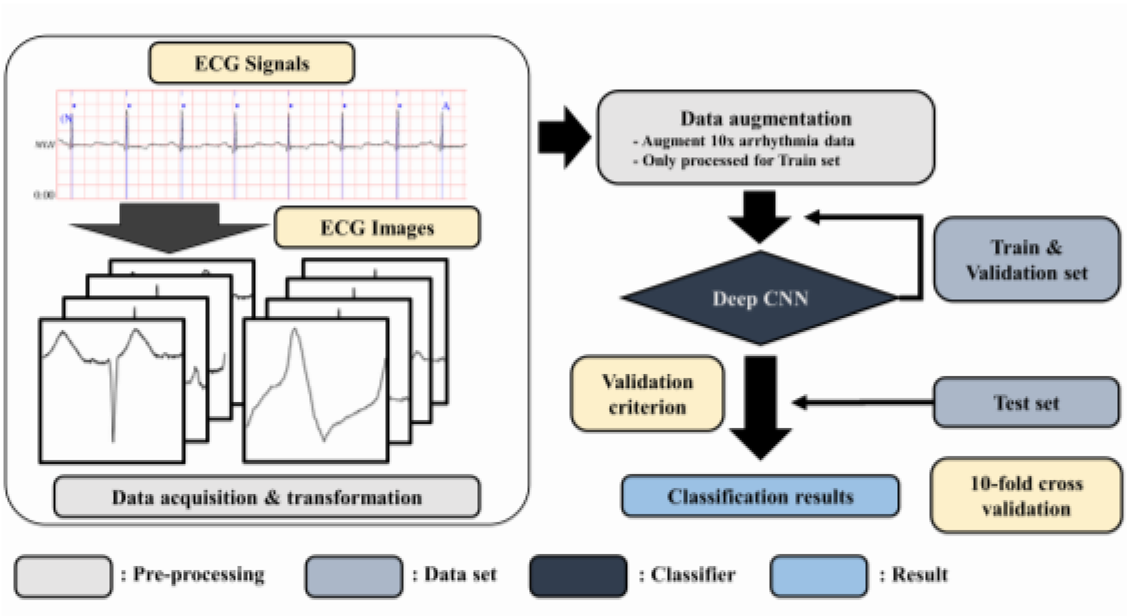
Although a number of the literature were proposed for the ECG arrhythmia classification, they have one or more of the following limitations: 1) good performance on carefully selected ECG recordings without cross-validation, 2) ECG beat loss in noise filtering and feature extraction schemes, 3) limited number of ECG arrhythmia types for the classification, 4) relatively low classification performance to adopt in practical.

2.2) Proposed Problem

In this project, we propose an effective electrocardiogram (ECG) arrhythmia classification method using a deep two-dimensional convolutional neural network (CNN), in which we classify ECG into 5 categories, one being normal and the other six being different types of arrhythmia using deep two-dimensional CNN . We are creating a web application where the user selects the image which is to be classified. The image is fed in to the model that is trained and predicted class will be displayed on web page.

3) THEORITICAL ANALYSIS

3.1) Block Diagram



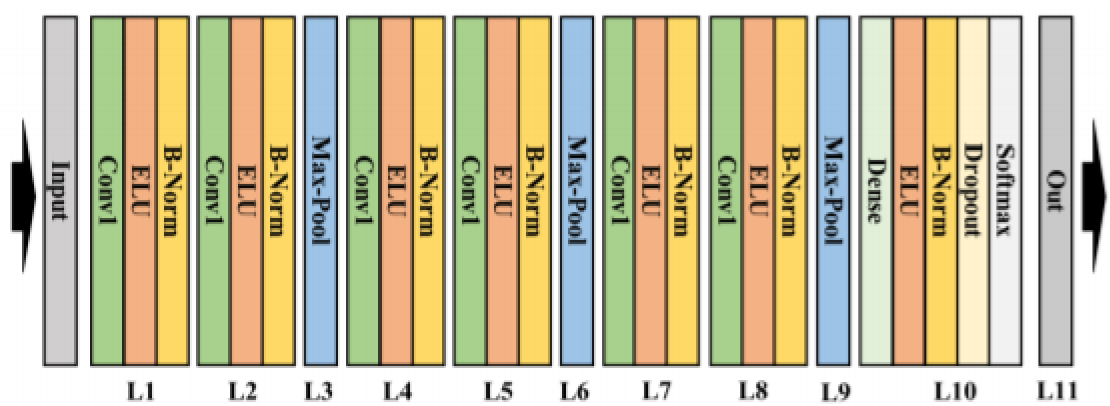
3.2) Hardware / Software designing

Proposed classifier and two other CNN models are deployed in Python language with TensorFlow which is an open source software library for deep learning launched by Google. Since CNN requires a lot of free parameters to train, GPGPU support is strongly recommended to reduce the learning time of the model. Thus, our experimental system is composed of two servers each contains two spyder (python 3.7), 64GB main memory. Versions of each software are TensorFlow r 1.14.0, Opencv2 4.2.0, and keras 2.2.4

4) EXPERIMENTAL INVESTIGATIONS

The performance of the proposed CNN model was compared with two well known CNN models, AlexNet and VGGNet, and results of existing ECG arrhythmia classification literature. MIT-BIH arrhythmia database is used for the evaluation of the experiments.We compared the performance of proposed CNN model with previous ECG arrhythmia classification works. Since these works have a different number of the test set and types of arrhythmia, it is unfair to directly compared with accuracy itself.

5) FLOWCHART



6) RESULT

The performance of the proposed CNN model was compared with two well known CNN models and its clear that this model has more accuracy than other models.

7) ADVANTAGES AND DISADVANTAGES

ADVANTAGES :

CNNs are very good **feature extractors.**

Convolutional Neural Networks take advantage of *local spatial coherence* in the input (often images), which allow them to have fewer weights as some parameters are shared. This process, taking the form of ***convolutions***, makes them especially well suited to extract relevant information at a low computational cost.

DISADVANTAGES

The usage of CNNs are motivated by the fact that they can capture / are able to learn relevant features from an image /video at different levels similar to a human brain. This is **feature learning !** Conventional neural networks cannot do this.

8) APPLICATIONS

It can be used by doctors to detect the type of arrthymia easily.

9) CONCLUSION

we proposed an effective ECG arrhythmia classification method using two-dimensional convolutional neural networks with ECG image as an input. 128 x 128 grayscale images are transformed from the MIT-BIH arrhythmia database ECG recording. Over 100,000 ECG beat images are obtained with eight types of ECG beats including normal beat and seven arrhythmia beats. Optimized CNN model is designed with considering important concepts such as data augmentation, regularization, and K-fold cross-validation. As a result, our proposed scheme achieved 0.989 AUC, 99.05% average accuracy, 99.57% specificity, 97.85% average sensitivity, and 98.55% average positive predictive value. Our ECG arrhythmia classification result indicates that detection of arrhythmia with ECG images and CNN model can be an effective approach to help the experts to diagnose cardiovascular diseases which can be seen from ECG signals

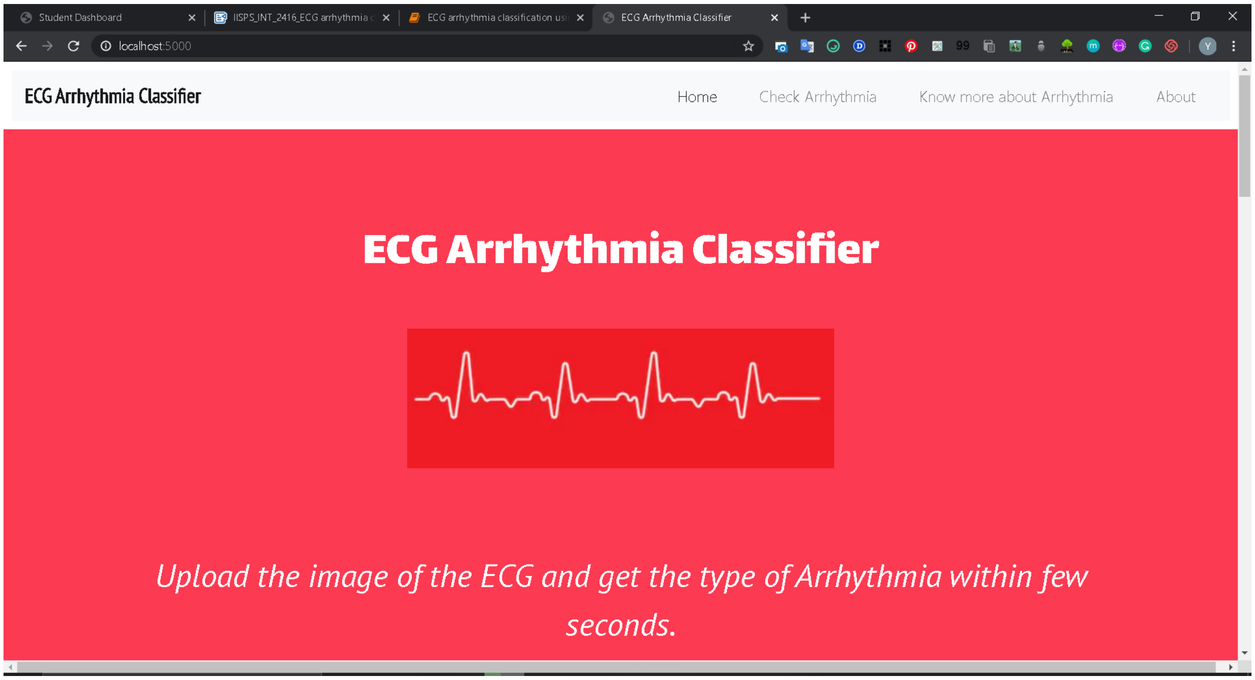
10) FUTURE SCOPE

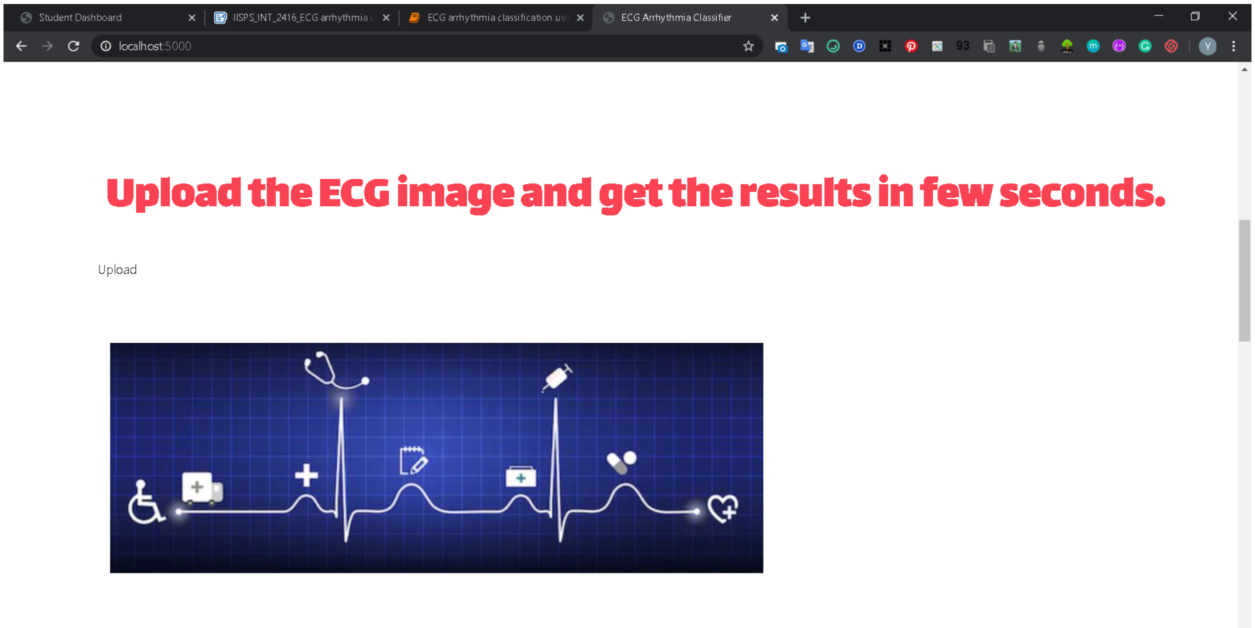
Our ECG arrhythmia classification result indicates that detection of arrhythmia with ECG images and CNN model can be an effective approach to help the experts to diagnose cardiovascular diseases which can be seen from ECG signals. Furthermore, proposed ECG arrhythmia classification method can be applied to the medical robot or the scanner that can monitors the ECG signals and helps the medical experts to identify ECG arrhythmia more precisely and easily. For the future work, we are building an integrated ECG arrhythmia classification system that scans the patient’s ECG monitor through the camera of the medical robot and diagnose the arrhythmia to inform the physician. Proposed ECG arrhythmia classification will be applied the ECG images obtained from the medical robot’s camera.

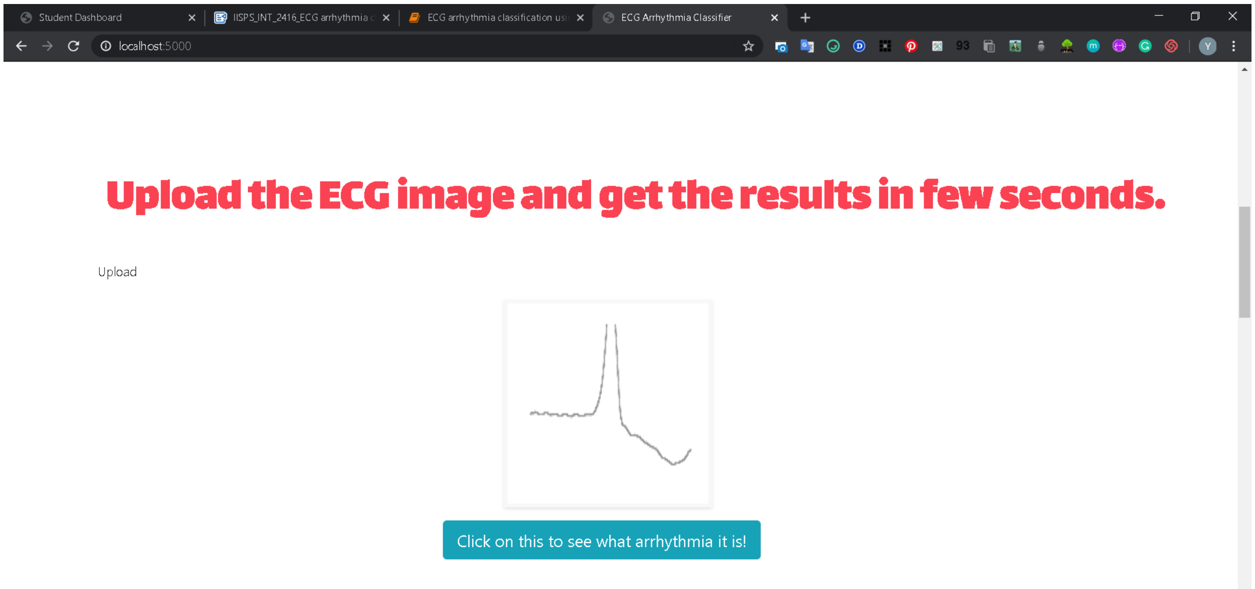
11) BIBILOGRAPHY

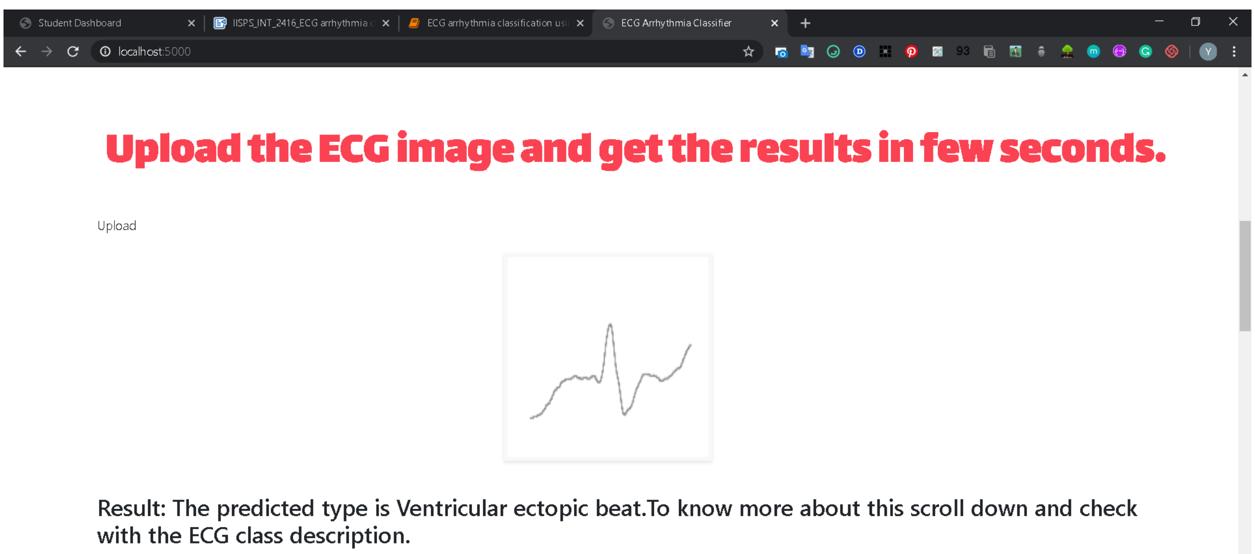
APPENDIX

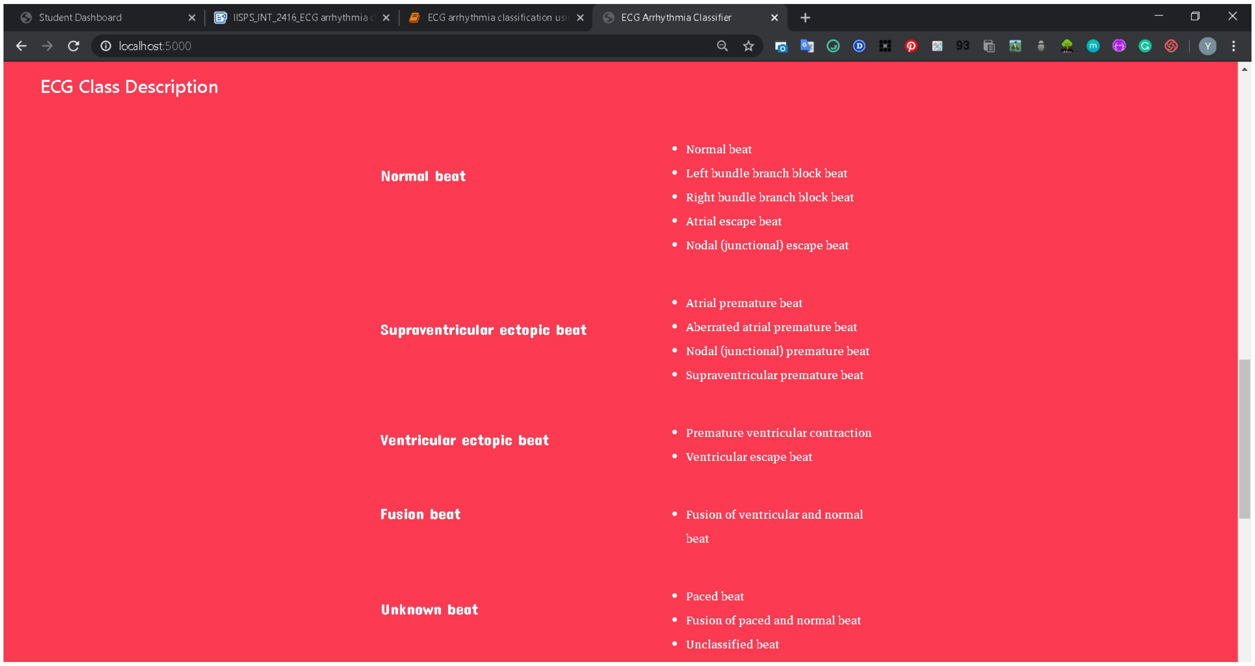
Screenshots :

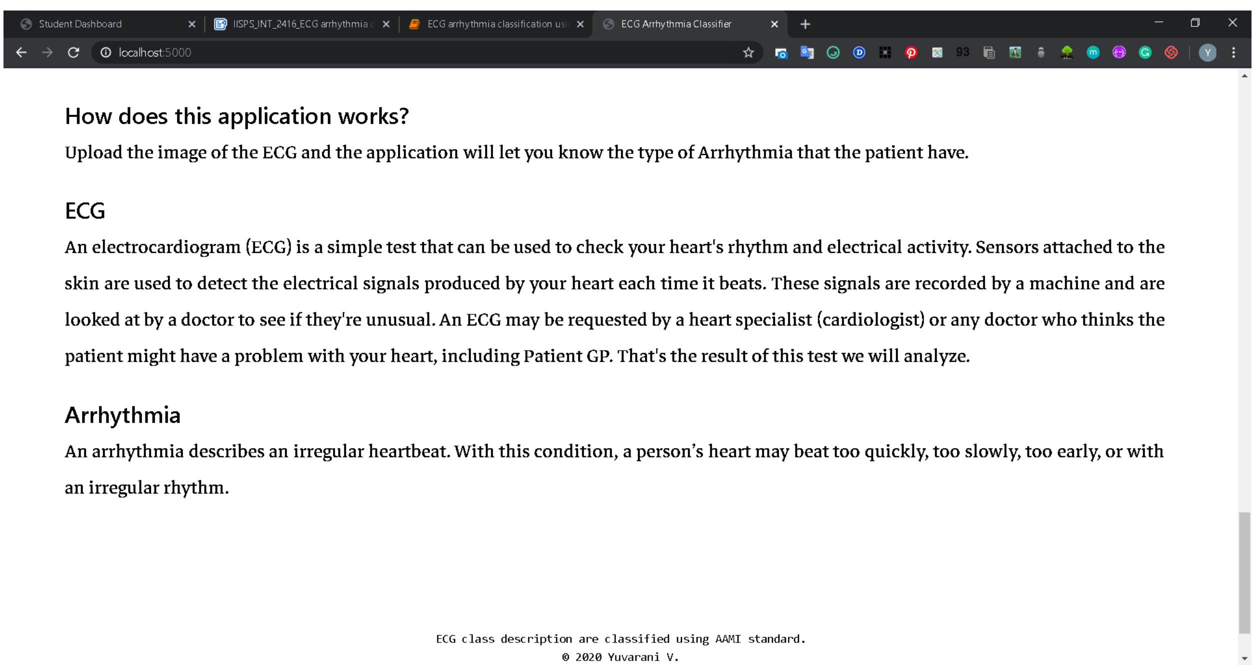





Student Dashboard
IISPS_INT_2416_ECG arrhythmia c
ECG arrhythmia classification usi
ECG Arrhythmia Classifier
localhost:5000
Upload the ECG image and get the results in few seconds.
Upload
Click on this to see what arrhythmia it is!


Student Dashboard
IISPS_INT_2416_ECG arrhythmia c
ECG arrhythmia classification usi
ECG Arrhythmia Classifier
localhost:5000
Upload the ECG image and get the results in few seconds.
Upload
Result: The predicted type is Ventricular ectopic beat.To know more about this scroll down and check with the ECG class description.


Student Dashboard
IISPS_INT_2416_ECG arrhythmia c
ECG arrhythmia classification usi
ECG Arrhythmia Classifier
localhost:5000
ECG Class Description
Normal beat
Normal beat
Left bundle branch block beat
Right bundle branch block beat
Atrial escape beat
Nodal (junctional) escape beat
Supraventricular ectopic beat
Atrial premature beat
Aberrated atrial premature beat
Nodal (junctional) premature beat
Supraventricular premature beat
Ventricular ectopic beat
Premature ventricular contraction
Ventricular escape beat
Fusion beat
Fusion of ventricular and normal beat
Unknown beat
Paced beat
Fusion of paced and normal beat
Unclassified beat


Student Dashboard
IISPS_INT_2416_ECG arrhythmia c
ECG arrhythmia classification usi
ECG Arrhythmia Classifier
localhost:5000
How does this application works?
Upload the image of the ECG and the application will let you know the type of Arrhythmia that the patient have.
ECG
An electrocardiogram (ECG) is a simple test that can be used to check your heart's rhythm and electrical activity. Sensors attached to the skin are used to detect the electrical signals produced by your heart each time it beats. These signals are recorded by a machine and are looked at by a doctor to see if they're unusual. An ECG may be requested by a heart specialist (cardiologist) or any doctor who thinks the patient might have a problem with your heart, including Patient GP. That's the result of this test we will analyze.
Arrhythmia
An arrhythmia describes an irregular heartbeat. With this condition, a person’s heart may beat too quickly, too slowly, too early, or with an irregular rhythm.
ECG class description are classified using AAMI standard.
© 2020 Yuvarani V.

A. Source Code

Flask file :

```
import numpy as np
import os
from keras.models import load_model
from keras.preprocessing import image
import tensorflow as tf
global graph
graph = tf.get_default_graph()
from flask import Flask , request, render_template
from werkzeug.utils import secure_filename
from gevent.pywsgi import WSGIServer

app = Flask(__name__)
model = load_model("ECG-1.h5")

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

@app.route('/predict',methods = ['GET','POST'])
def upload():
    if request.method == 'POST':
        f = request.files['image']
        print("current path")
        basepath = os.path.dirname(__file__)
        print("current path", basepath)
        filepath = os.path.join(basepath,'uploads',f.filename)
        print("upload folder is ", filepath)
        f.save(filepath)

        img = image.load_img(filepath,target_size = (64,64))
        x = image.img_to_array(img)
        x = np.expand_dims(x,axis =0)
```

```
        with graph.as_default():
            preds = model.predict_classes(x)
            4

            print("prediction",preds)

        index = ['Fusion Beat','Normal Beat','Unknown Beat','Supraventricular ectopic
Beat','Ventricular ectopic beat']


        text = "The predicted type is " + str(index[preds[0]]) +".To know more about this
scroll down and check with the ECG class description."

    return text
if __name__ == '__main__':
    app.run(debug = True)
```

html file :

```
<html lang="en">

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <meta http-equiv="X-UA-Compatible" content="ie=edge">
    <title>ECG Arrhythmia Classifier</title>
    <link href="https://cdn.bootcss.com/bootstrap/4.0.0/css/bootstrap.min.css"
rel="stylesheet">
    <script src="https://cdn.bootcss.com/popper.js/1.12.9/umd/popper.min.js"></script>
    <script src="https://cdn.bootcss.com/jquery/3.3.1/jquery.min.js"></script>
    <script src="https://cdn.bootcss.com/bootstrap/4.0.0/js/bootstrap.min.js"></script>
      <link href="{{ url_for('static', filename='css/main.css') }}" rel="stylesheet">
<link
href="https://fonts.googleapis.com/css2?family=Concert+One&family=Fredoka+One&f
amily=Lalezar&family=Markazi+Text&family=PT+Sans+Narrow&family=PT+Sans:ital@1
```

```
&family=Share+Tech+Mono&display=swap" rel="stylesheet">

</head>

<body>
<nav class="navbar navbar-expand-lg navbar-light bg-light">
  <button class="navbar-toggler" type="button" data-toggle="collapse"
data-target="#navbarTogglerDemo03" aria-controls="navbarTogglerDemo03"
aria-expanded="false" aria-label="Toggle navigation">
    <span class="navbar-toggler-icon"></span>
  </button>
  <a class="navbar-brand" href="#">ECG Arrhythmia Classifier</a>

  <div class="collapse navbar-collapse" id="navbarTogglerDemo03 options">
    <ul class="navbar-nav ml-auto">
      <li class="nav-item active">
        <a class="nav-link" href="#Home">Home<span
class="sr-only">(current)</span></a>
      </li>
      <li class="nav-item">
        <a class="nav-link" href="#CheckECG">Check Arrhythmia</a>
      </li>
       <li class="nav-item">
        <a class="nav-link" href="#predict">Know more about Arrhythmia</a>
      </li>
      <li class="nav-item">
       <a class="nav-link" href="#about">About</a>
     </li>
    </ul>
  </div>
</nav>
<!-- Home -->
<section id="Home">
<div class="top">
  <h1>ECG Arrhythmia Classifier</h1><br><br>

<img src="{{url_for('static', filename='img1.jpg')}}"/>
```

```
  <p>Upload the image of the ECG and get the type of Arrhythmia within few
seconds.</p>
</div>
</section>
<!-- check ECG -->
<section id="CheckECG">
<div class="checkecg">
  <div class="row">
        <div class="">
          <h1 class="big-heading">Upload the ECG image and get the results in few
seconds.</h1>
          <form action = "http://localhost:5000/predict" id="upload-file" method="post"
enctype="multipart/form-data">
            <br>
            <div class="lab">
              <label for="imageUpload" class="upload-label">
                                        Upload
                                </label>
                                <input type="file" name="image" id="imageUpload"
accept=".png, .jpg, .jpeg">
                        </form>
            </div>

      <div class="image-section " style="display:none;">
                                <div class="img-preview">
                                        <div id="imagePreview">
                                        </div>
                                </div>
                                <div>
                                        <button type="button" class="btn btn-info btn-lg "
id="btn-predict">Click on this to see what arrhythmia it is!</button>
                                </div>
                        </div>
<br>
                        <div class="loader" style="display:none;"></div>

                        <h3>
```

```
                                    <span id="result"> </span>
                              </h3>

                        </div>
                </div>
                <br>
                <div class="">
                  <img src="{{url_for('static', filename='img.jpg')}}"/>
                </div>
              </div>
            </div>
          </section>
        </div>
        </section>

        <!-- predict -->
        <section id="predict">
        <div class="">
          <h3>ECG Class Description</h3>
          <div class="container">
            <div class="row cla">
              <h4>Normal beat</h4>
            </div>
            <div class="Feature-box col-lg-4 col-md-12">
              <ul>
                <li>Normal beat</li>
                <li>Left bundle branch block beat</li>
                <li>Right bundle branch block beat</li>
                <li>Atrial escape beat</li>
                <li>Nodal (junctional) escape beat</li>
              </ul>
            </div>
          </div>
        <div class="container">
          <div class="row cla">
              <h4>Supraventricular ectopic beat</h4>
          </div>
```

```
    <div class="Feature-box col-lg-4 col-md-12">
      <ul>
        <li>Atrial premature beat</li>
        <li>Aberrated atrial premature beat </li>
        <li>Nodal (junctional) premature beat</li>
        <li>Supraventricular premature beat</li>
      </ul>
    </div>
</div>
<div class="container">
  <div class="row v">
    <h4>Ventricular ectopic beat</h4>
  </div>
<div class="Feature-box col-lg-4 col-md-12">
  <ul>
    <li>Premature ventricular contraction</li>
    <li>Ventricular escape beat</li>
  </ul>
</div>
</div>
  <div class="container">
    <div class="row f">
      <h4>Fusion beat </h4>
    </div>
    <div class="Feature-box col-lg-4 col-md-12">
      <ul>
        <li>Fusion of ventricular and normal beat</li>
      </ul>
    </div>
  </div>
  <div class="container">
    <div class="row u">
        <h4>Unknown beat</h4>
    </div>
  <div class="Feature-box col-lg-4 col-md-12">
    <ul>
      <li>Paced beat</li>
```

```
      <li>Fusion of paced and normal beat</li>
      <li>Unclassified beat</li>
    </ul>
  </div>
  </div>


</div>

</section>
<!-- About -->
<section id="about">
<div id="home" class="ab">
  <div class="">
    <h3>How does this application works?</h3>
    <p>Upload the image of the ECG and the application will let you know the type of
Arrhythmia that the patient have.</p>
  </div>
  <div class="">
  <h3>ECG</h3>
  <p>An electrocardiogram (ECG) is a simple test that can be used to check your heart's
rhythm and electrical activity.
    Sensors attached to the skin are used to detect the electrical signals produced by
your heart each time it beats.
    These signals are recorded by a machine and are looked at by a doctor to see if
they're unusual.
    An ECG may be requested by a heart specialist (cardiologist) or any doctor who thinks
the patient might have a problem with your heart, including Patient GP. That's the result
of this test we will analyze.
  </p>
</div>

<div class="">
  <h3>Arrhythmia</h3>
  <p>An arrhythmia describes an irregular heartbeat. With this condition, a person’s heart
may beat too quickly, too slowly, too early, or with an irregular rhythm.</p>
</div>
```

```html
</div>


<footer>
  <div id="footer" class="container-fluid white">
      <p class="foot1">ECG class description are classified using AAMI standard.</p>
      <p class="foot1">&copy; 2020 Yuvarani V.</p>

  </div>

    <script src="{{ url_for('static', filename='js/main.js') }}" type="text/javascript"></script>
</footer>
</body>
</html>
```

Css File :

```css
#options{
  text-align: right;
}
#about{
  padding: 70px;
  background-color:  white;
  color: black;
}
#predict{
padding: 3%;
  background-color: #fc3a52;
  color: white;
}
.top{
  text-align: center;
padding: 2%;
  background-color: #fc3a52;
  color: white;
}
.bg-light{
```

```
  background-color: #ffaaaa;
  margin: 10px;
}
.navbar-brand{
  font-family: 'PT Sans Narrow', sans-serif;
  font-size: 25px;
  font-weight: bold;
}
h1{
  font-family: 'Lalezar', cursive;
  font-size: 60px;
padding-top: 6%;
}
.top p{
font-family: 'PT Sans', sans-serif;
padding: 7%;
font-size: 40px;
}
#about h3,{
font-family: 'Concert One', cursive;
font-size: 50px;
}
.Feature-box{
  font-family: 'Markazi Text', serif;
  font-size: 25px;
}
#about p{
  font-family: 'Markazi Text', serif;
  font-size: 30px;
  padding-bottom: 15px;
  text-align: justify;
}
#hoe
.container{
  padding-bottom: 30px;
}
.container-fluid {
```

```
  padding: 3% 7%;
}
.Feature-box{
  position: relative;
    left: 600;
    top: 20;
}
.cla{
  position: relative;
top: 100;
left: 200;
font-family: 'Concert One', cursive;
}

.v{
  position: relative;
top: 70;
left: 200;
font-family: 'Concert One', cursive;
}
.f{
  position: relative;
top: 58;
left:200;
font-family: 'Concert One', cursive;
}
.u{
  position: relative;
top: 79;
left:200;
font-family: 'Concert One', cursive;
}
.big-heading {
  font-weight: 900;
  font-size: 3.5rem;
  line-height: 1.5;
  color: #fa4252;
```

```
  text-align: left;
  margin: 10px;
}


.checkecg{
  margin: 40px;
}

.img-preview {
width: 256px;
height: 256px;
    position: relative;
    left:500;
    border: 5px solid #F8F8F8;
    box-shadow: 0px 2px 4px 0px rgba(0, 0, 0, 0.1);
    margin-top: 1em;
    margin-bottom: 1em;
    margin-right: 5em;
}



.img-preview>div {
    width: 100%;
    height: 100%;
    background-size: 256px 256px;
    background-repeat: no-repeat;
    background-position: center;
}
input[type="file"] {
    display: none;
}

}
#imageUpload{
    display: inline-block;
```

```
    padding: 12px 30px;
    font-size: 1em;
    transition: all .4s;
    cursor: pointer;

}
#CheckECG{
  margin: 2%;
  padding:0 1% 2% 4%;
}

.upload-label{
font-weight:200;
}

.upload-label:hover{
    background: #34495E;
    color: #39D2B4;
    font-weight: 900;
}

.loader {
  border: 8px solid black;
    border-top: 8px solid #3498db;
    border-radius: 50%;
    width: 50px;
    height: 50px;
    animation: spin 1s linear infinite;

}
#btn-predict{
position: relative;
    left: 425;
}
.ab{
  padding:  0 20px 80px 5px;
}
```

```
.nav-item {
  padding: 0 18px;
}

.nav-link {
  font-size: 1.2rem;
  font-weight: lighter;
}
#footer p.foot1{
    font-family:monospace;
    font-size: 15px;
    text-align: center;
    margin: 0;
    padding: 0;
    position: relative;
  left: 0;
  right: 0;
  bottom: 0;
}
```

Js file :

```
$(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!');
        },
    });
```

```
    });

});
```

Data PreProcessing :

```
from keras.models import Sequential
from keras.layers import Dense
from keras.layers import Convolution2D
from keras.layers import MaxPooling2D
from keras.layers import Flatten
from keras.preprocessing.image import ImageDataGenerator
train_datagen=ImageDataGenerator(rescale=1./255,shear_range=0.2,zoom_range=0.2,horizontal_flip=True)
test_datagen=ImageDataGenerator(rescale=1./255)
x_train =
train_datagen.flow_from_directory(r"G:\data\train",target_size=(64,64),batch_size=32,class_mode="categorical")
x_test =
test_datagen.flow_from_directory(r"G:\data\test",target_size=(64,64),batch_size=32,class_mode="categorical")
print(x_train.class_indices)
```

Model Building :

```
model=Sequential()
model.add(Convolution2D(32,(3,3),input_shape=(64,64,3),activation="relu"))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Flatten())
model.add(Dense(init='uniform',activation='relu',units=300))
model.add(Dense(init='uniform',activation='relu',units=100))
model.add(Dense(init='uniform',activation='relu',units=60))
model.compile(loss="categorical_crossentropy",optimizer="adam",metrics=["accuracy"])
model.fit_generator(x_train,steps_per_epoch=2736,
epochs=1,validation_data=x_test,validation_steps=684)
model.save("ECG_arrhythmia.h5")
```