SMARTINTERNZ

GUIDED PROJECT

Classification of Arrhythmia by using deep learning with 2-D ECG Spectral Image Representation

Team ID:
LTVIP2023TMID04419

Team Members:

Chamantula Pradeepkumar
Bandi Saikumar
Ketharajupalli Praveenkumar
Pilli Dileep Reddy
Muppidi Dhanush

Project Description

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. The image is fed into the model that is trained and the cited class will be displayed on the webpage.

Project Objectives

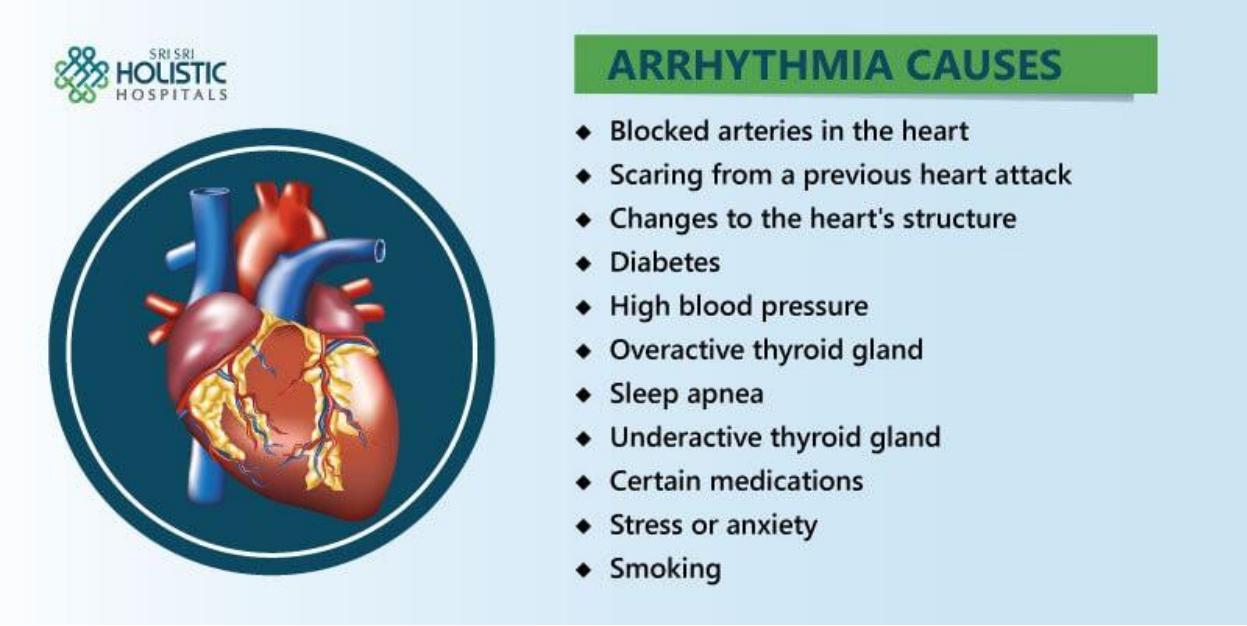
By the end of this project I

- knew fundamental concepts and techniques of the Artificial Neural Network and Convolution Neural Networks
- Gained a broad understanding of image data.
- Worked with Sequential type of modeling
- Worked with Keras capabilities
- Worked with image processing techniques
- knew how to build a web application using the Flask framework.

Literature Review

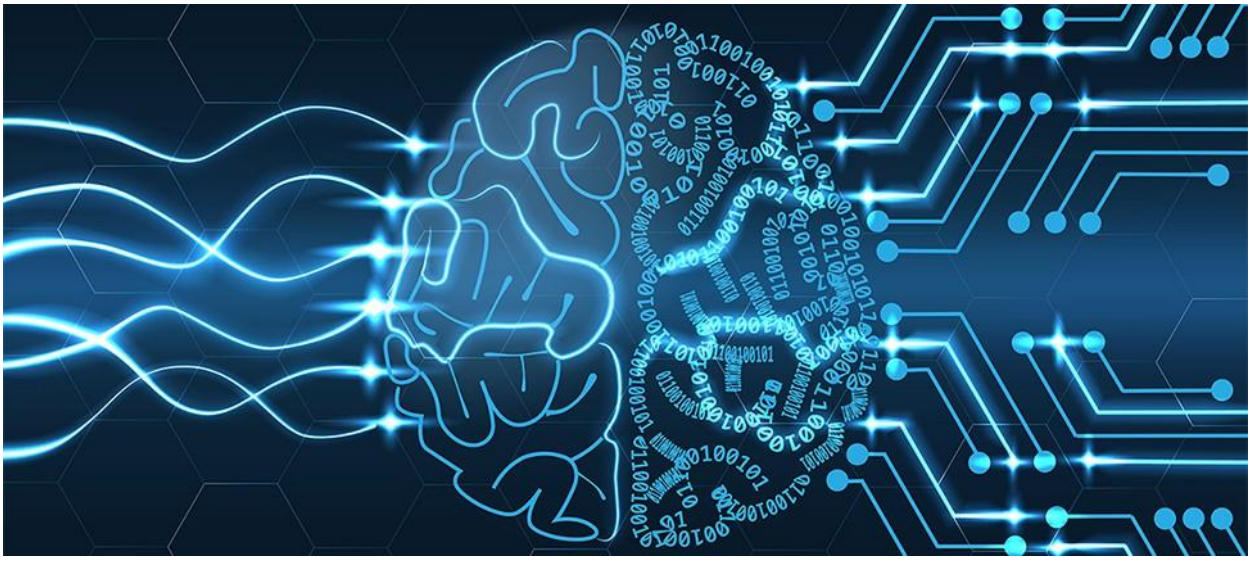
Problem

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 arrhythmia including atrial fibrillation, premature contraction, ventricular fibrillation, and tachycardia.



Solution

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



Role of Deep Learning

Deep learning is a branch of machine learning which is completely based on artificial neural networks, as neural network is going to mimic the human brain so deep learning is also a kind of mimic of human brain. In deep learning, we don't need to explicitly program everything. The concept of deep learning is not new. It has been around for a couple of years now. It's on hype nowadays because earlier we did not have that much processing power and a lot of data. As in the last 20 years, the processing power increases exponentially, deep learning and machine learning came in the picture.

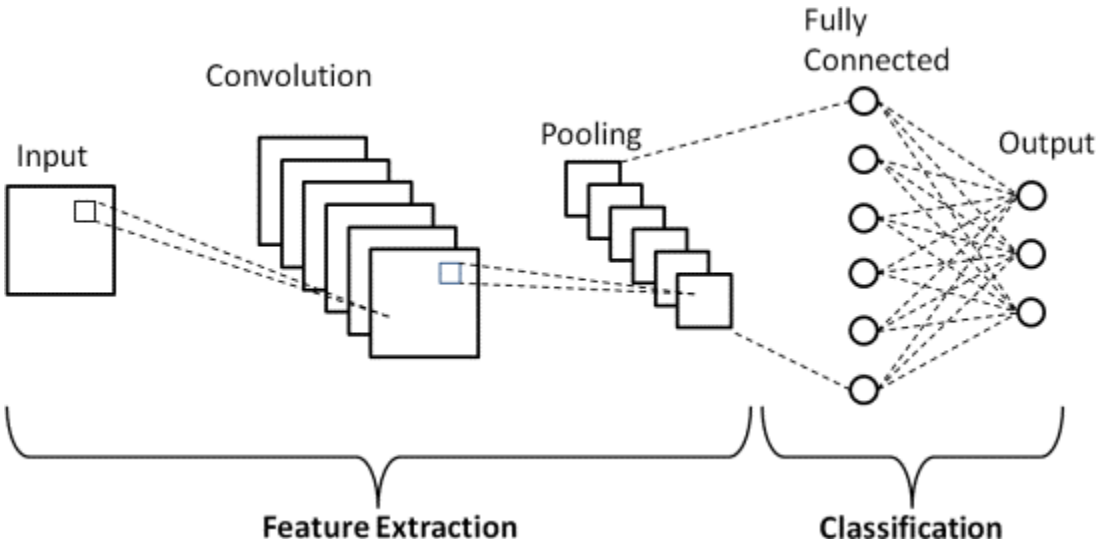
Deep learning is a particular kind of machine learning that achieves great power and flexibility by learning to represent the world as a nested hierarchy of concepts, with each concept defined in relation to simpler concepts, and more abstract representations computed in terms of less abstract ones.

CNN

In the field of deep learning, convolutional neural network (CNN) is among the class of deep neural networks, which was being mostly deployed in the field of analyzing/image recognition.Convolutional Neural uses a very special kind of method which is being known as Convolution.

The Convolutional neural networks(CNN) consists of various layers of artificial neurons. Artificial neurons, similar to that neuron cells that are being used by the human brain for passing various sensory input signals and other responses, are mathematical functions

that are being used for calculating the sum of various inputs and giving output in the form of an activation value.



The behaviour of each CNN neuron is being defined by the value of its weights. When being fed with the values (of the pixel), the artificial neurons of a CNN recognizes various visual features and specifications.

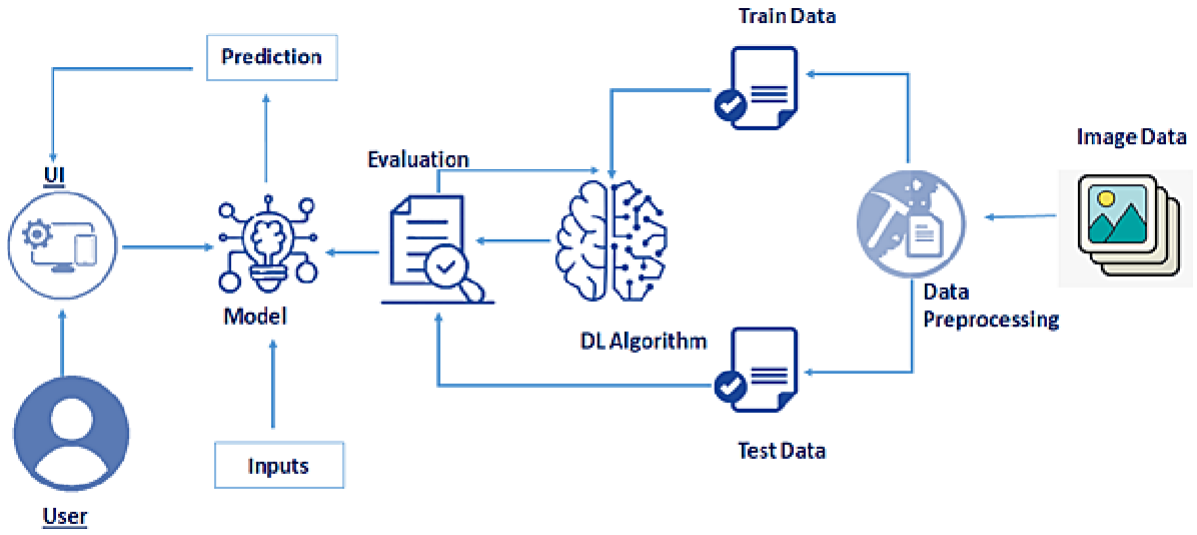
When we give an input image into a CNN, each of its inner layers generates various activation maps. Activation maps point out the relevant features of the given input image. Each of the CNN neurons generally takes input in the form of a group/patch of the pixel, multiplies their values(colours) by the value of its weights, adds them up, and input them through the respective activation function.

The first (or maybe the bottom) layer of the CNN usually recognizes the various features of the input image such as edges horizontally, vertically, and diagonally.

The output of the first layer is being fed as an input of the next layer, which in turn will extract other complex features of the input image like corners and combinations of edges.

The deeper one moves into the convolutional neural network, the more the layers start detecting various higher-level features such as objects, faces, etc...

Theoretical Experience



We will prepare the project by following the below steps:

- We will be working with Sequential type of modeling, Keras capabilities, image processing techniques
- We will build a web application using the Flask framework.
- Afterwards we will be training our dataset in the IBM cloud and building another model from IBM and we will also test it.

HARDWARE & SOFTWARE Desgining

Hardware Components used

Since we are using the IBM cloud as a platform to execute this project we don't need any hardware components other than our system.

Software Components Used

We will be using Anaconda Navigator which is installed in our system and Watson studio from the IBM cloud to complete the project.

- ***Anaconda Navigator***

  Anaconda Navigator is a free and open-source distribution of the Python and R

programming languages for data science and machine learning related applications. It can be installed on Windows, Linux, and macOS.Conda is an open-source, cross-platform, package management system. Anaconda comes with so very nice tools like JupyterLab, Jupyter Notebook, QtConsole, Spyder, Glueviz, Orange, Rstudio, Visual Studio Code. For this project, we will be using Jupiter notebook and spyder.

- ***Watson Studio***

  Watson Studio is one of the core services in Cloud Pak for Data as a Service. Watson Studio provides you with the environment and tools to solve your business problems by collaboratively working with data. You can choose the tools you need to analyze and visualize data, to cleanse and shape data, or to build machine learning models. This illustration shows how the architecture of Watson Studio is centered around the project. A project is a workspace where you organize your resources and work with data.

  Watson Studio projects fully integrate with the catalogs and deployment spaces:

  ● Deployment spaces are provided by the Watson Machine Learning service

  You can easily move assets between projects and deployment spaces

Experimental Investigations

In this project, we have deployed our training model using CNN on IBM Watson studio and in our local machine. We are deploying 4 types of CNN layers in a sequential manner , starting from

- Convolutional layer 2D:A 2-D convolutional layer applies sliding convolutional filters to 2-D input. The layer convolves the input by moving the filters along the input vertically and horizontally and computing the dot product of the weights and the input, and thenadding a bias term.
- Pooling Layer :Pooling layers are used to reduce the dimensions of the feature maps. Thus, it reduces the number of parameters to learn and the amount of computation performed in the network. The pooling layer summarises the features present in a region of the feature map generated by a convolution layer.
- Fully-Connected layer :After extracting features from multiple convolution layers and pooling layers, the fully-connected layer is used to expand the connection of all features. Finally, the SoftMax layer makes a logistic regression classification. Fully-connected layer transfers the weighted sum of the output of the previous layer to the activation function.
- Dropout Layer :There is usually a dropout layer before the fully- connected layer.

The dropout layer will temporarily disconnect some neurons from the network according to the certain probability during the training of the convolution neural network, which reduces the joint adaptability between neuron nodes, reduces overfitting, and enhances the generalization ability.

Project Flow

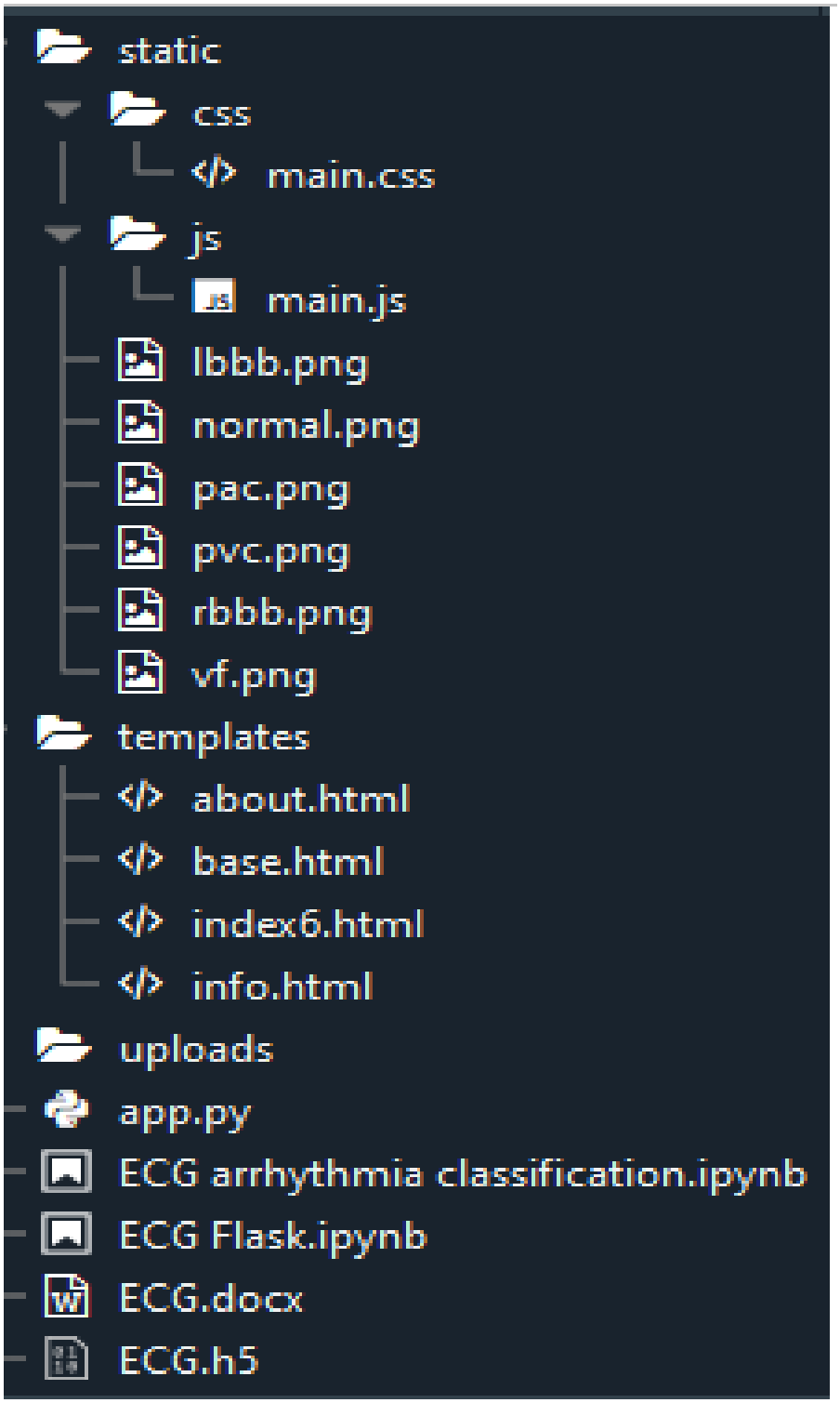
- User interacts with User interface to upload image
- Uploaded image is analyzed by the model which is integrated
- Once model analyses the uploaded image, the prediction is showcased on the UI

To accomplish this, we have to complete all the activities and tasks listed below.

- Data Collection.
    - Collect the dataset or Create the dataset
- Data Preprocessing.
    - Import the ImageDataGenerator library
    - Configure ImageDataGenerator class
    - Apply ImageDataGenerator functionality to Train dataset and Test dataset
- Model Building
    - Import the model building Libraries
    - Initializing the model
    - Adding Input Layer
    - Adding Hidden Layer
    - Adding Output Layer
    - Configure the Learning Process
    - Training and testing the model
    - Optimize the Model
    - Save the Model
- Application Building
    - Create an HTML file
    - Build Python Code
- Training model on IBM Cloud.

Project Structure

Create a Project folder which contains files as shown below



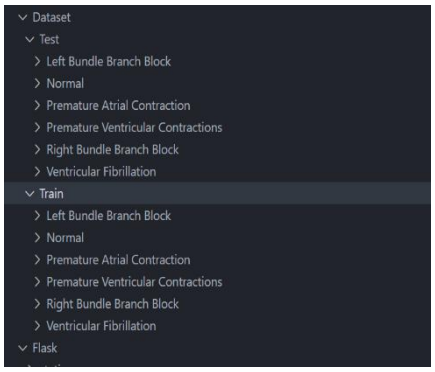
- We are building a Flask Application that needs HTML pages stored in the templates folderand a python script app.py for serverside scripting
- we need the model which is saved and the saved model in this content is ECG.h5
- The static folder will contain js and CSS files.
- Whenever we upload an image to predict, that images are saved in the uploads folder.

WORKING ON PROJECT

1. Dataset Collection

The dataset contains six classes:

1. Left Bundle Branch Block
2. Normal
3. Premature Atrial Contraction
4. Premature Ventricular Contractions
5. Right Bundle Branch Block
6. Ventricular Fibrillation



2. Image Processing

The dataset of our project includes images of different classes (6 classes) which need "Image Processing" step before feeding the input to ANN. This Image processing include

- Augmenting the image feature, image data generator library.
- Load dataset.
- Apply augmented feature to train set and test set.

3. Model Building

As the Image Processing step is done, now we start to build the CNN model for prediction. This step includes the following steps:

- Import libs
- Initialize the model
- Add CNN layers
- Configure your learning
- Fit the data
- Save the model

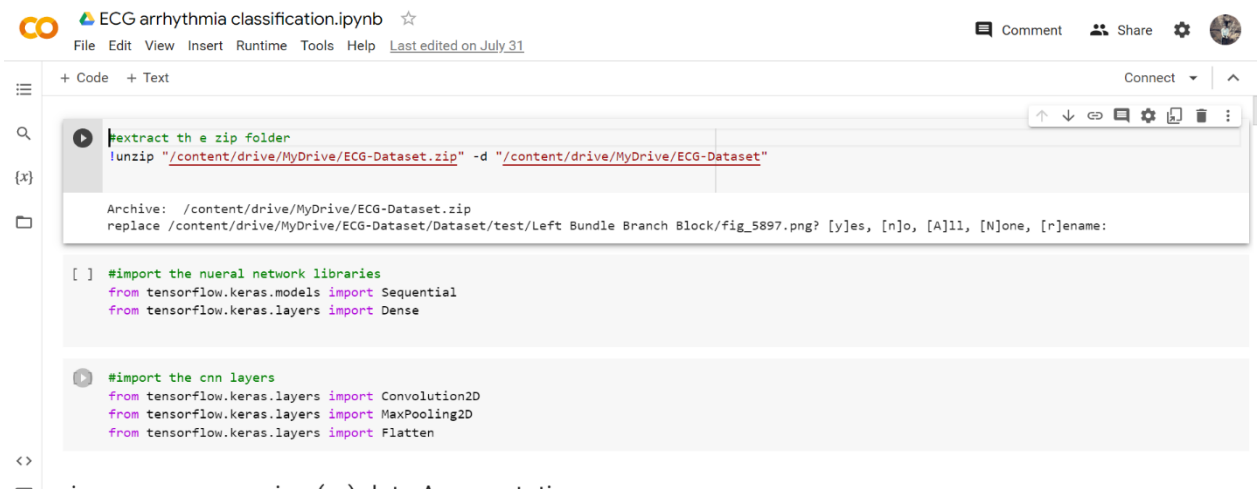
4. Application Building

In this project, we not only build a CNN model, but we also build an Application, a proper platform to use this trained model, using a micro web framework called **FLASK**.
For this we create HTML pages (with associated CSS and JS pages) which shares some information about Arrhythmia and classifications, and in the 'predict' page, we predict the type of Arrhythmia given as Image Input.
These HTML pages are used to make a proper interface (a website) to showcase our project output.

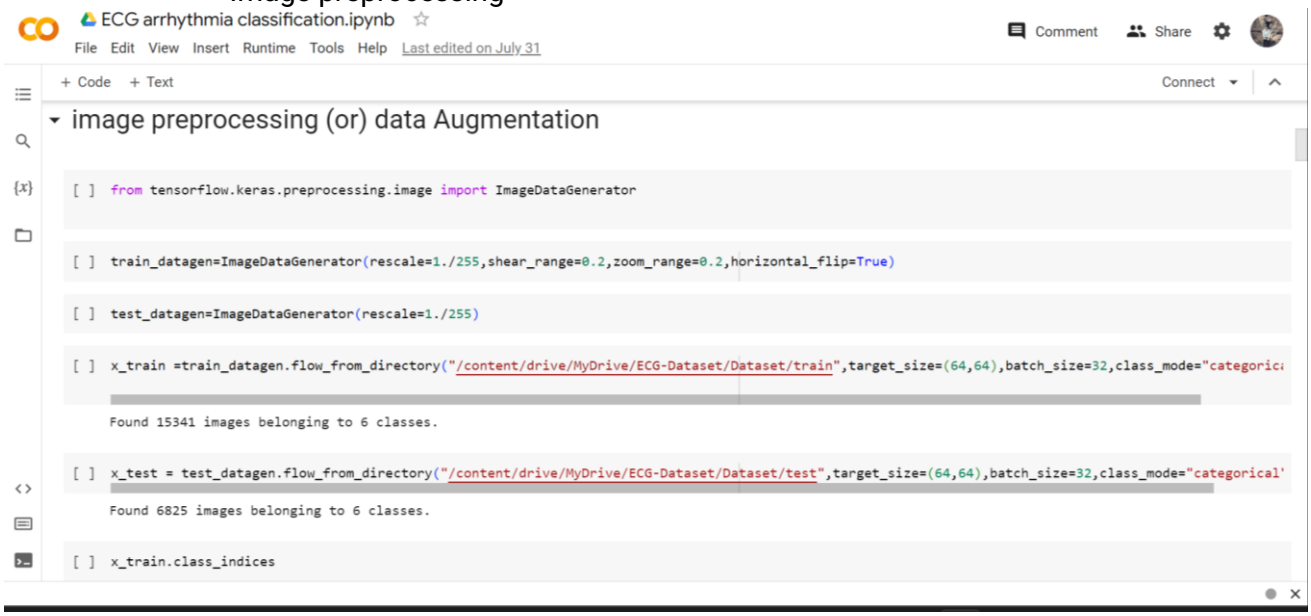
Flask folder includes
templates (to store the .html files needed for our website)
static (to store .css .js and some images included in .html files)
uploads (to store the images we uploaded while using the app)
app.py (flask application development python file)
<modelname>.h5 file (CNN trained model that is given to flask application as input)

5. Train the model on Google Colob or Jupyter Notebook

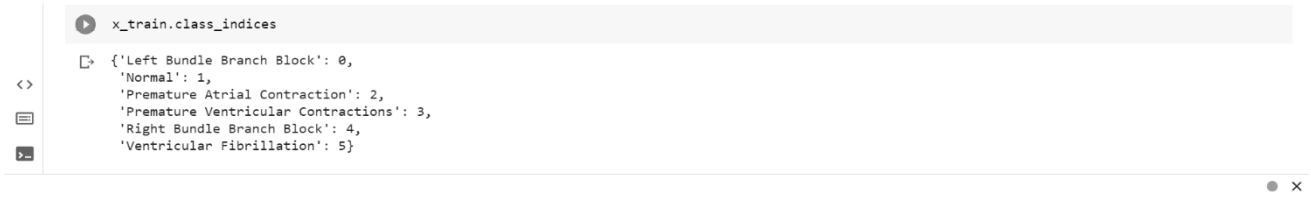
- unzip the zip folder in google colob from drive



- Image preprocessing



- Train class Indices



- Model Building

hidden layers

```
[ ] model.add(Dense(units=200,activation="relu",kernel_initializer="random_uniform"))
```

```
[ ] model.add(Dense(units=300,activation="relu",kernel_initializer="random_uniform"))
```

output layer

```
[ ] model.add(Dense(units=6,activation="softmax",kernel_initializer="random_uniform"))
```

compile model

```
[ ] model.compile(optimizer="adam",loss="categorical_crossentropy",metrics=["accuracy"])
```

ECG arrhythmia classification.ipynb

File Edit View Insert Runtime Tools Help Last edited on July 31

Comment Share

+ Code + Text Connect

train your model

```
tr=model.fit_generator(x_train,steps_per_epoch=480,epochs=25,validation_data=x_test,validation_steps=10)
#steps_per_epoch =>total trainging images/batch size
#validation_steps=>total testing images/batch size
```

```
Epoch 1/25
<ipython-input-29-85df04309dce>:1: UserWarning: `Model.fit_generator` is deprecated and will be removed in a future version. Please use `Model.fit`, wh
  tr=model.fit_generator(x_train,steps_per_epoch=480,epochs=25,validation_data=x_test,validation_steps=10)
480/480 [==============================] - 139s 290ms/step - loss: 0.1375 - accuracy: 0.9570 - val_loss: 0.4658 - val_accuracy: 0.8344
Epoch 2/25
480/480 [==============================] - 135s 280ms/step - loss: 0.1019 - accuracy: 0.9696 - val_loss: 0.5208 - val_accuracy: 0.8656
Epoch 3/25
480/480 [==============================] - 138s 288ms/step - loss: 0.0817 - accuracy: 0.9740 - val_loss: 0.4992 - val_accuracy: 0.9031
Epoch 4/25
480/480 [==============================] - 137s 285ms/step - loss: 0.0653 - accuracy: 0.9776 - val_loss: 0.6651 - val_accuracy: 0.8719
Epoch 5/25
480/480 [==============================] - 138s 286ms/step - loss: 0.0542 - accuracy: 0.9817 - val_loss: 1.0073 - val_accuracy: 0.8375
Epoch 6/25
480/480 [==============================] - 139s 289ms/step - loss: 0.0553 - accuracy: 0.9817 - val_loss: 0.9780 - val_accuracy: 0.8344
Epoch 7/25
480/480 [==============================] - 136s 284ms/step - loss: 0.0415 - accuracy: 0.9860 - val_loss: 0.9159 - val_accuracy: 0.8375
Epoch 8/25
```

ECG arrhythmia classification.ipynb

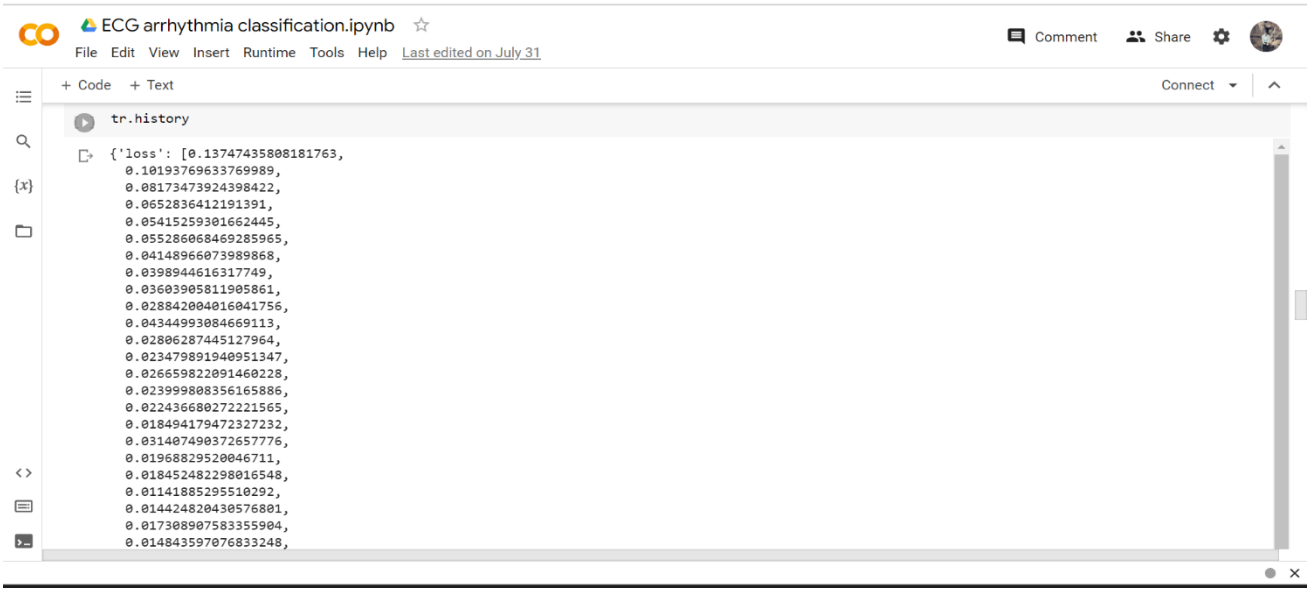
File Edit View Insert Runtime Tools Help Last edited on July 31

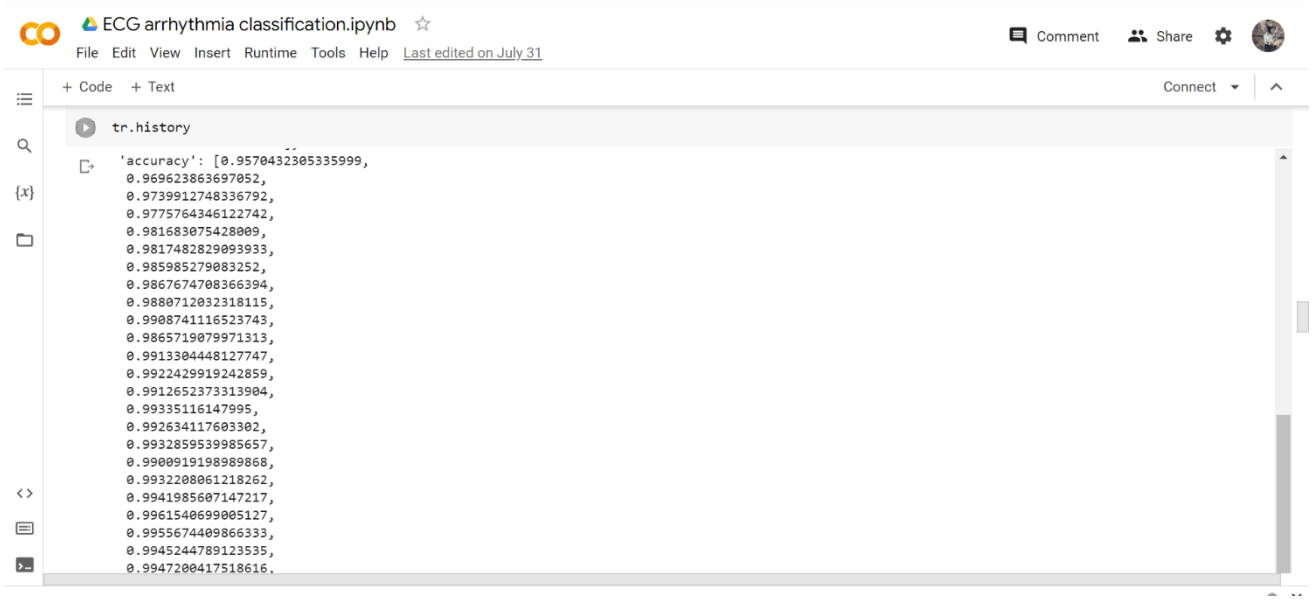
Comment Share

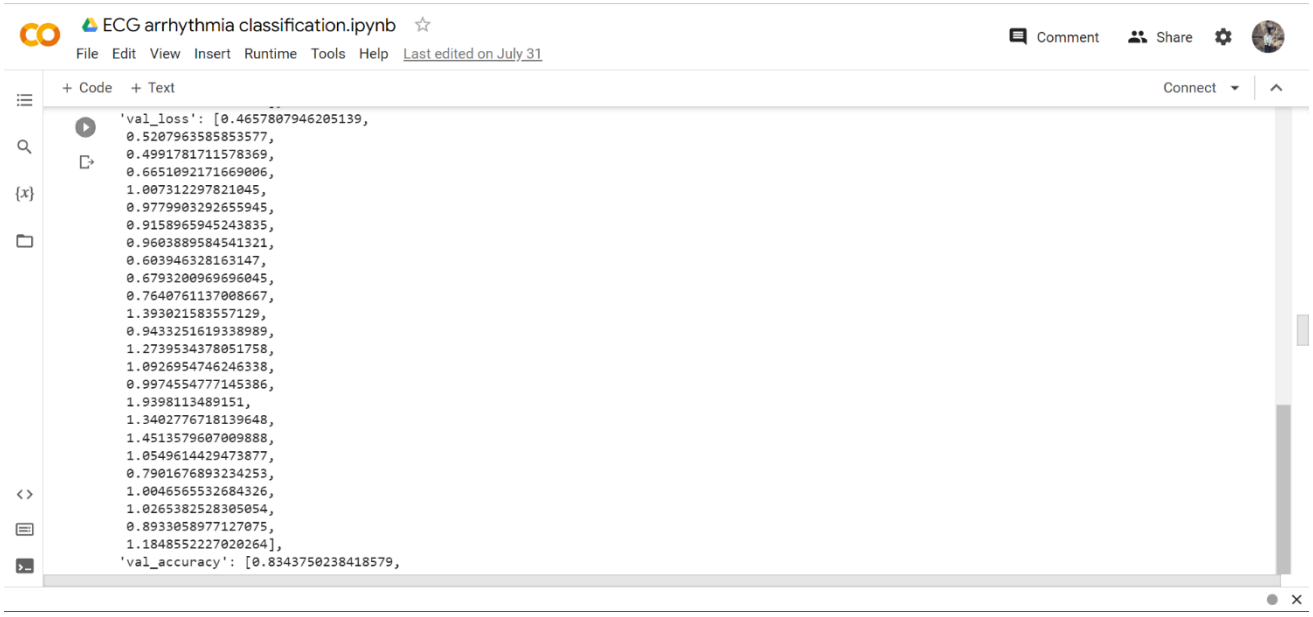
+ Code + Text Connect

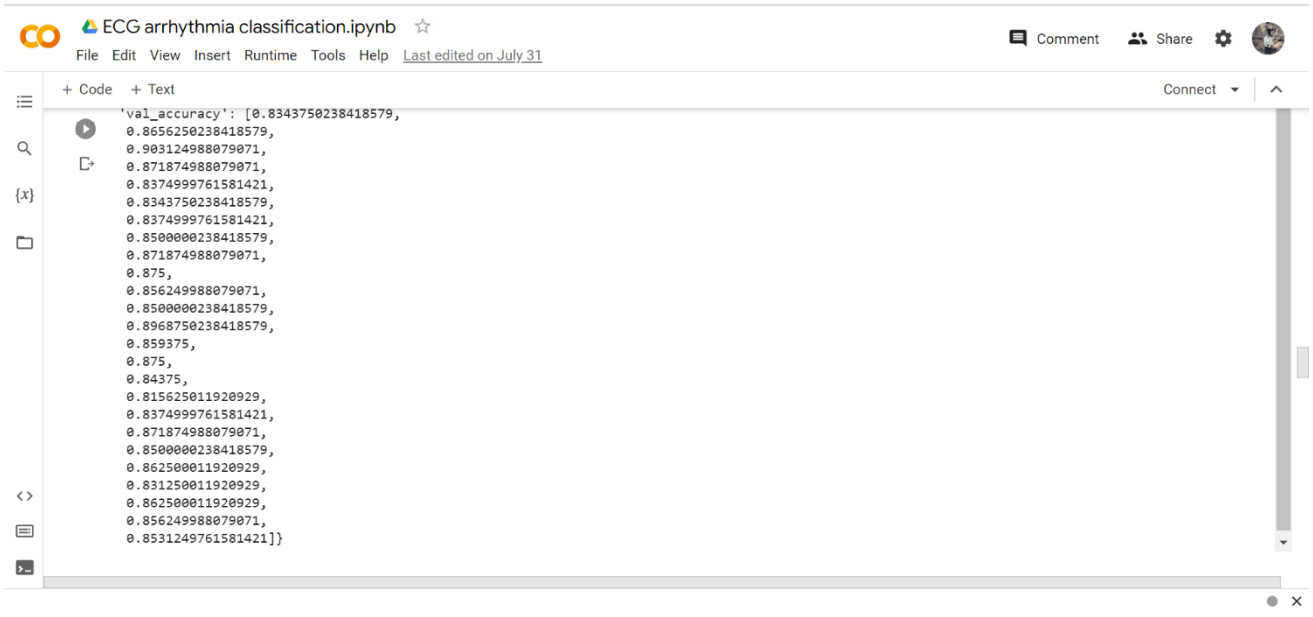
```
Epoch 13/25
480/480 [==============================] - 133s 277ms/step - loss: 0.0235 - accuracy: 0.9922 - val_loss: 0.9433 - val_accuracy: 0.8969
Epoch 14/25
480/480 [==============================] - 133s 276ms/step - loss: 0.0267 - accuracy: 0.9913 - val_loss: 1.2740 - val_accuracy: 0.8594
Epoch 15/25
480/480 [==============================] - 134s 279ms/step - loss: 0.0240 - accuracy: 0.9934 - val_loss: 1.0927 - val_accuracy: 0.8750
Epoch 16/25
480/480 [==============================] - 136s 283ms/step - loss: 0.0224 - accuracy: 0.9926 - val_loss: 0.9975 - val_accuracy: 0.8438
Epoch 17/25
480/480 [==============================] - 136s 282ms/step - loss: 0.0185 - accuracy: 0.9933 - val_loss: 1.9398 - val_accuracy: 0.8156
Epoch 18/25
480/480 [==============================] - 134s 280ms/step - loss: 0.0314 - accuracy: 0.9901 - val_loss: 1.3403 - val_accuracy: 0.8375
Epoch 19/25
480/480 [==============================] - 136s 282ms/step - loss: 0.0197 - accuracy: 0.9932 - val_loss: 1.4514 - val_accuracy: 0.8719
Epoch 20/25
480/480 [==============================] - 135s 280ms/step - loss: 0.0185 - accuracy: 0.9942 - val_loss: 1.0550 - val_accuracy: 0.8500
Epoch 21/25
480/480 [==============================] - 135s 281ms/step - loss: 0.0114 - accuracy: 0.9962 - val_loss: 0.7902 - val_accuracy: 0.8625
Epoch 22/25
480/480 [==============================] - 134s 279ms/step - loss: 0.0144 - accuracy: 0.9956 - val_loss: 1.0047 - val_accuracy: 0.8313
Epoch 23/25
480/480 [==============================] - 136s 283ms/step - loss: 0.0173 - accuracy: 0.9945 - val_loss: 1.0265 - val_accuracy: 0.8625
Epoch 24/25
480/480 [==============================] - 133s 278ms/step - loss: 0.0148 - accuracy: 0.9947 - val_loss: 0.8933 - val_accuracy: 0.8562
Epoch 25/25
480/480 [==============================] - 134s 279ms/step - loss: 0.0135 - accuracy: 0.9950 - val_loss: 1.1849 - val_accuracy: 0.8531
```

- You can view the train history about loss,accuracy,val_loss,val_accuracy






ECG arrhythmia classification.ipynb
File Edit View Insert Runtime Tools Help Last edited on July 31
Comment
Share
+ Code + Text
Connect
'val_loss': [0.4657807946205139,
0.5207963585853577,
0.4991781711578369,
0.6651092171669006,
1.007312297821045,
0.9779903292655945,
0.9158965945243835,
0.9603889584541321,
0.603946328163147,
0.6793200969696045,
0.7640761137008667,
1.393021583557129,
0.9433251619338989,
1.2739534378051758,
1.0926954746246338,
0.9974554777145386,
1.9398113489151,
1.3402776718139648,
1.4513579607009888,
1.0549614429473877,
0.7901676893234253,
1.0046565532684326,
1.0265382528305054,
0.8933058977127075,
1.1848552227020264],
'val_accuracy': [0.8343750238418579,


ECG arrhythmia classification.ipynb
File Edit View Insert Runtime Tools Help Last edited on July 31
Comment
Share
+ Code + Text
Connect
'val_accuracy': [0.8343750238418579,
0.8656250238418579,
0.903124988079071,
0.871874988079071,
0.8374999761581421,
0.8343750238418579,
0.8374999761581421,
0.8500000238418579,
0.871874988079071,
0.875,
0.856249988079071,
0.8500000238418579,
0.8968750238418579,
0.859375,
0.875,
0.84375,
0.815625011920929,
0.8374999761581421,
0.871874988079071,
0.8500000238418579,
0.862500011920929,
0.831250011920929,
0.862500011920929,
0.856249988079071,
0.8531249761581421]}

- Saving the model

Create html files and flask application

- About.html

```
<!DOCTYPE html>
<html>
<head>
<title>Home</title>
<style>
body
{
    background-image: url("https://thumbs.gfycat.com/ChiefHeftyBasil-small.gif");
    background-size: cover;
}
.pd{
padding-bottom:100%;}
.navbar
{
margin: 0px;
padding:20px;
background-color:white;
opacity:0.6;
color:black;
font-family:'Roboto',sans-serif;
font-style: italic;
border-radius:20px;
font-size:25px;
}
a
{
color:grey;
float:right;
text-decoration:none;
font-style:normal;
padding-right:20px;
}
a:hover{
background-color:black;
color:white;
border-radius:15px;0
font-size:30px;
padding-left:10px;
}
p
{
color:white;
font-style:italic;
font-size:30px;
```

```
}
</style>
</head>
<body>
<div class="navbar">
<a href="/upload" >Predict</a>
<a href="/info">Info</a>
<a href="/about">Home</a>
<br>
</div>
<br>
<center><b class="pd"><font color="white" size="15" font-family="Comic Sans MS" >ECG
arrhythmia classification using CNN</font></b></center>
<div>
<br>
<p>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.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.
</p>
</center>
</div>
</body>
</html>
```

- Info.html

```
<!DOCTYPE html>
<html>
    <head>
        <style>
        .navbar{
            margin: 0px;
```

```
        padding:10px;
        background-color:red;
        opacity:0.5;
        font-color:black;
        font-family:'Roboto',sans-serif;
        font-style: italic;
        border-radius:15px;
        font-size:30px;
    }
    a{
        color:grey;
        float:right;
        text-decoration:none;
        font-style:normal;
        padding-right:20px;
    }
    a:hover{
        background-color:black;
        color:white;
        border-radius:15px;0
        font-size:30px;
        padding-left:10px;
    }

    img{
        width:550px;
        height:400px;
        padding:10px;
        margin-top:0px;
    }
    img:hover{
        border-radius:100px;
        border-color:grey;
        border-shadow:10px;
    }
    body{
        background-color:none;
        background-size: cover;
    }
    h1{
        font-size:100px;
        text-align:center;
        color:white;
        font-style:italic;
        font-weight:bolder;
    }
    h2{
        font-size:50px;
        text-align:center;
```

```
            color:blue;
            font-style:italic;
            font-weight:bolder;
        }
        div{
            margin-left:50px;
        }
        img{
            width:1100px;
            height:600px;
            padding:10px;
            margin-top:0px;
        }
        img:hover{
            border-radius:100px;
            border-color:grey;
            border-shadow:10px;
        }
        </style>
        <title>Information about Arrhythmia Classification </title>
    </head>
    <body>
        <div class="navbar">
            <a href="/upload" >Predict</a>
            <a href="/info">Info</a>
            <a href="/about">Home</a>
        <br>
        </div>
        <div>
        <h1><u><font color = 'red'>ECG</font></u></h1>
        <h2>Electrocardiogram</h2>
         <div>
        <center>
        <span><img src="../static/normal.jpg" title="normal"></span>
        <span><img src="../static/vf.jpg" title="Ventricular Fibrillation"></span>
        <span><img src="../static/pac.jpg" title="Premature Atrial
Contraction"></span>
        <span><img src="../static/pvc.jpg" title="Premature Ventricular
Contraction"></span>
        <span><img src="../static/rbbb.jpg" title="Right Bundle Branch
Block"></span>
        <span><img src="../static/lbbb.jpg" title="Left Bundle Branch Block"></span>
        </center>
        </div>
        </div>
    </body>
</html>
```

- Base.html

```
<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>Predict</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">
<style>
.bar
{
margin: 0px;
padding:20px;
background-color:white;
opacity:0.6;
color:black;
font-family:'Roboto',sans-serif;
font-style: italic;
border-radius:20px;
font-size:25px;
}
a
{
color:grey;
float:right;
text-decoration:none;
font-style:normal;
padding-right:20px;
}
a:hover{
background-color:black;
color:white;
border-radius:15px;0
font-size:30px;
padding-left:10px;
}
body
```

```
{
    background-image: url("https://i.makeagif.com/media/9-23-2015/PUEusc.gif");
    background-size:cover;
}
</style>
</head>

<body>
<div class="bar">
<a href="/upload" >Predict</a>
<a href="/info">Info</a>
<a href="/about">Home</a>
<br>
</div>
    <div class="container">
       <center> <div id="content" style="margin-top:2em">{% block content %}{%
endblock %}</div></center>
    </div>
</body>

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

</html>
```

- Index6.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;
}

.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!');
            },
        });
    });

});
```

- 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

        pred=model.predict(x)#predicting classes
        y_pred = np.argmax(pred)
        print("prediction",y_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[y_pred])

        return result  #returning 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)
```

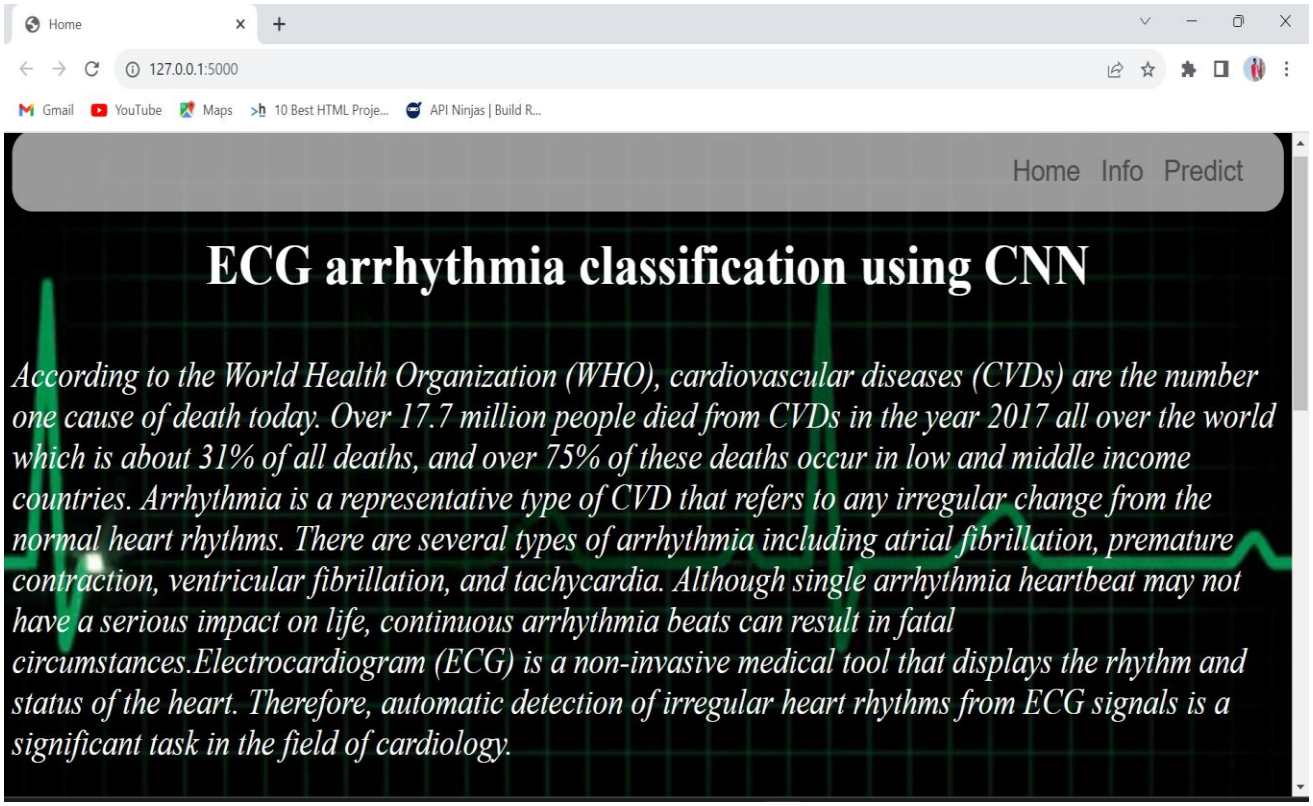
- Run the flask app

```
C:\Users\DILEEP\Desktop\Classification Of Arrhythmia\Flask>python app.py
```

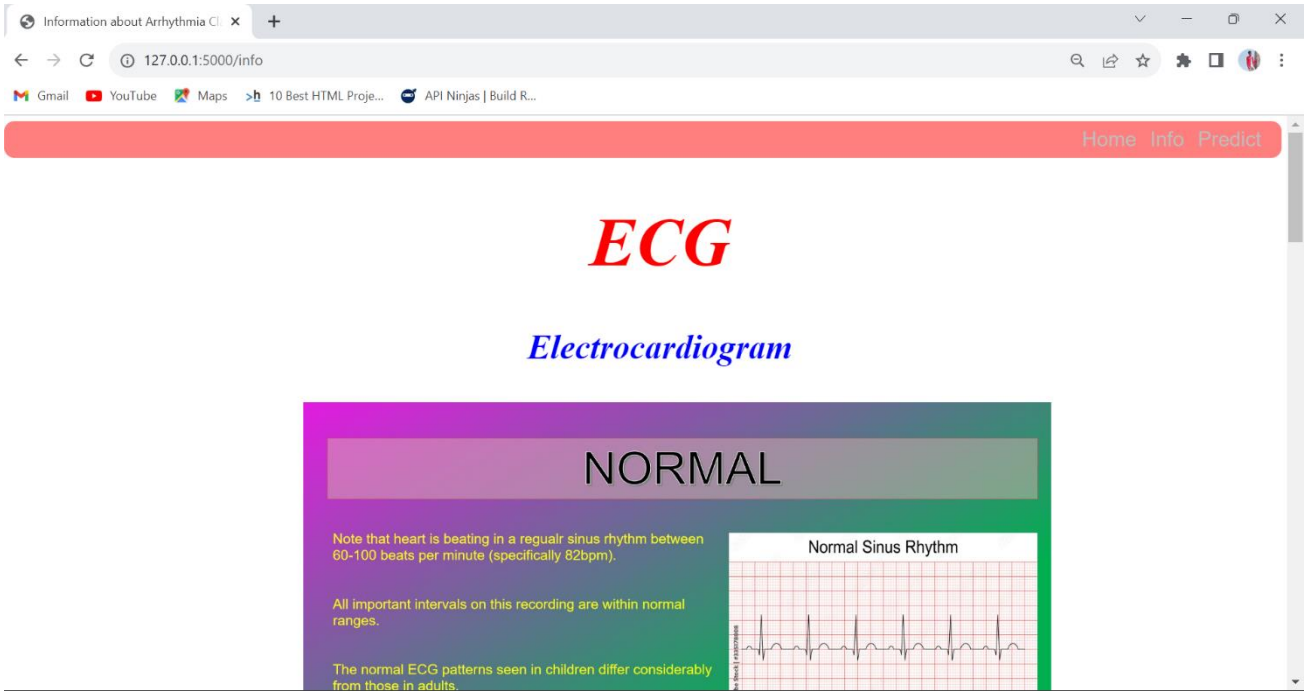
```
* Serving Flask app 'app'
* Debug mode: off
ARNING: This is a development server. Do not use it in a production deployment. Use a product
* Running on http://127.0.0.1:5000
ress CTRL+C to quit
```

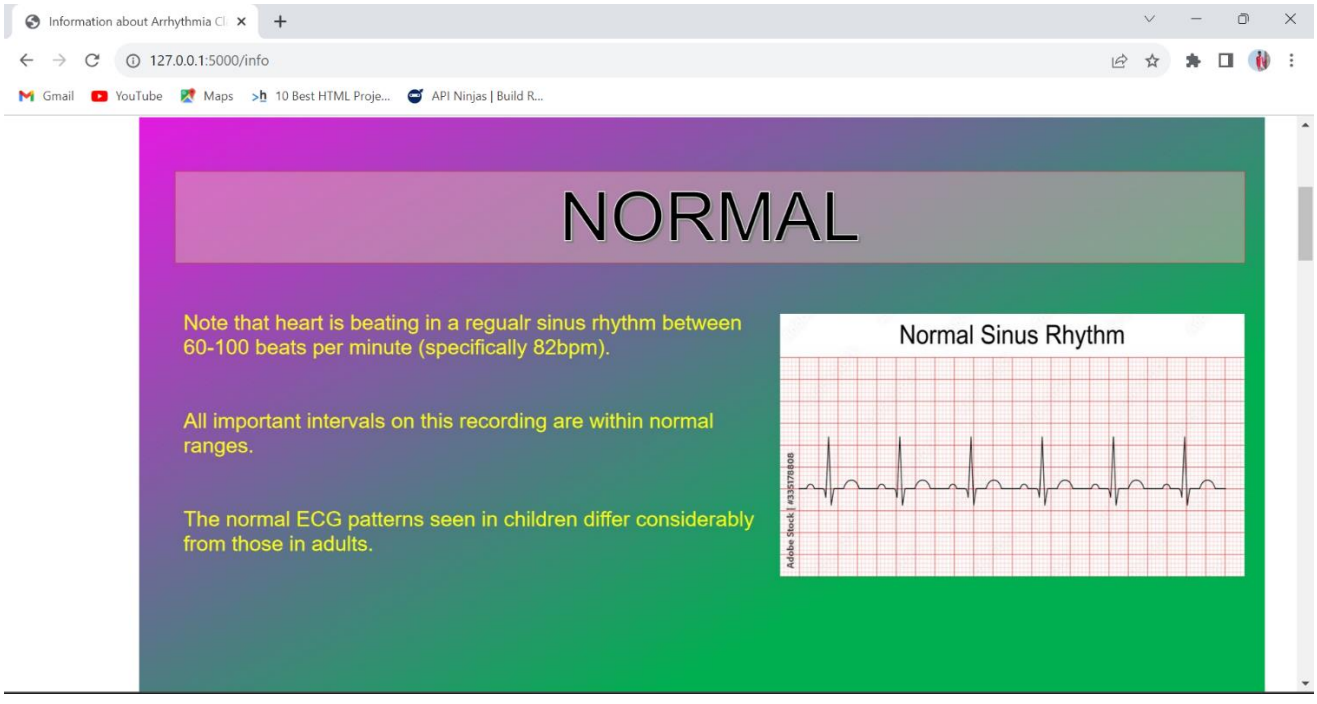
OUTPUT of Flask App Development

About.html



info.html


Information about Arrhythmia Cl
127.0.0.1:5000/info
Gmail
YouTube
Maps
10 Best HTML Proje...
API Ninjas | Build R...
Home
Info
Predict
ECG
Electrocardiogram
NORMAL
Note that heart is beating in a regualr sinus rhythm between 60-100 beats per minute (specifically 82bpm).
All important intervals on this recording are within normal ranges.
The normal ECG patterns seen in children differ considerably from those in adults.
Normal Sinus Rhythm


Information about Arrhythmia Cl
127.0.0.1:5000/info
Gmail
YouTube
Maps
10 Best HTML Proje...
API Ninjas | Build R...
NORMAL
Note that heart is beating in a regualr sinus rhythm between 60-100 beats per minute (specifically 82bpm).
All important intervals on this recording are within normal ranges.
The normal ECG patterns seen in children differ considerably from those in adults.
Normal Sinus Rhythm

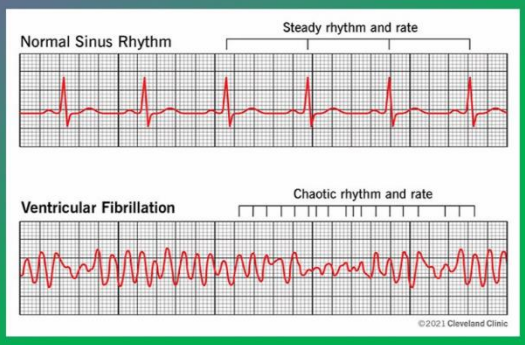
Information about Arrhythmia Cl

127.0.0.1:5000/info

Gmail YouTube Maps 10 Best HTML Proje... API Ninjas | Build R...

VENTRICULAR FIBRILLATION

- A life-threatening heart rhythm that results in a rapid, inadequate heartbeat.
- Ventricular fibrillation (VF) is a rapid, life-threatening heart rhythm starting in the bottom chambers of the heart. it can be triggered by a heart attack.
- Because the heart doesn't pump adeguately during ventricular filbrillation, sustained VF can cause low blood pressure, loss of consciosness or death.
- Emergency treatment includes immediate defibrillation with an automated external defibrillation (AED) and cardiopulmonary resuscitation (CPR). Long-term therapy includes implantable defribrillators and medications to prevent recurrence.



Information about Arrhythmia Cl

127.0.0.1:5000/info

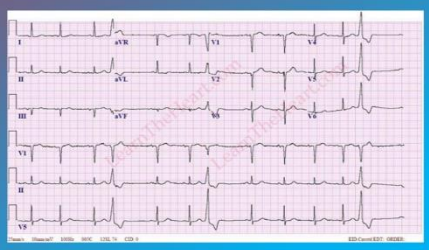
Gmail YouTube Maps 10 Best HTML Proje... API Ninjas | Build R...

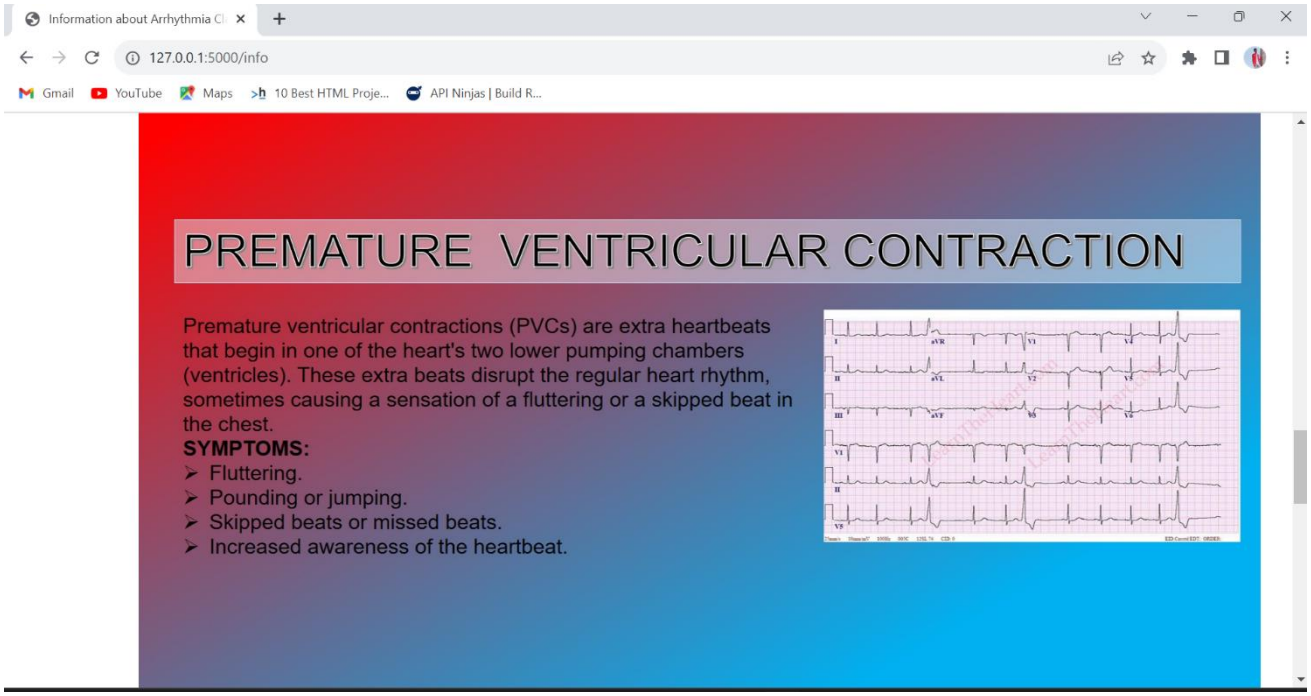
PREMATURE ATRIAL CONTRACTION

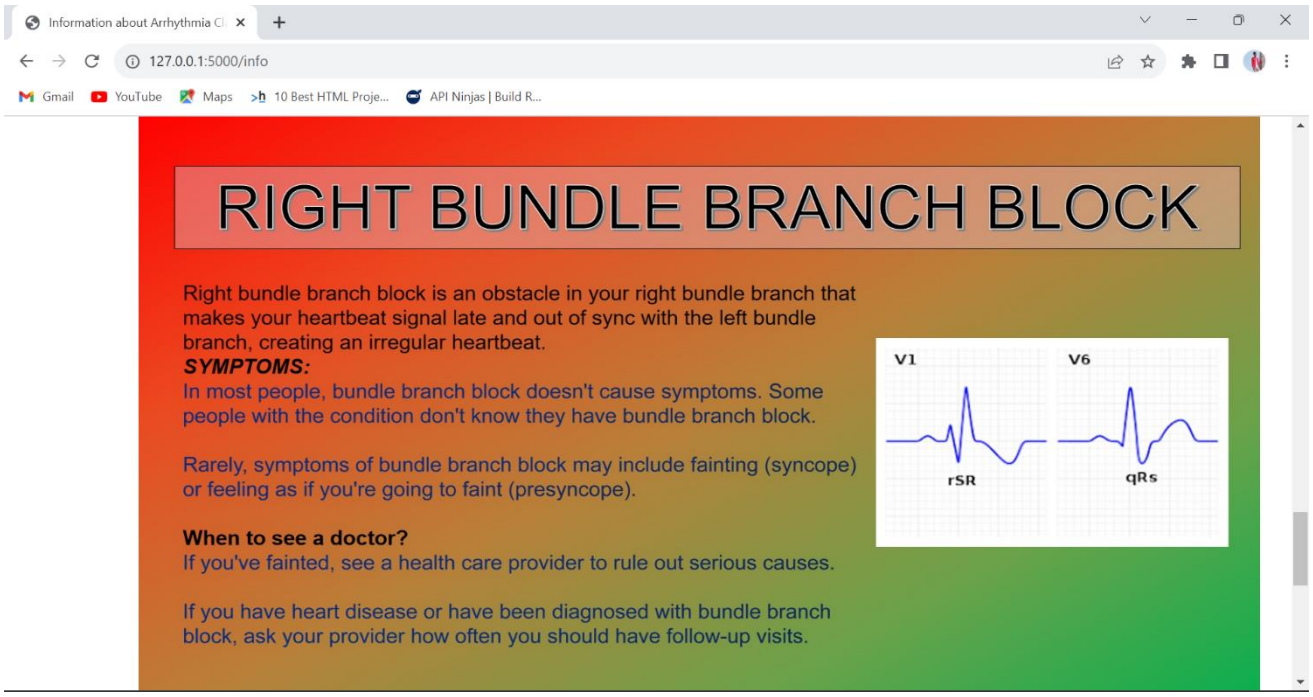
Premature atrial contractions (PACs) are extra heartbeats that start in the upper chambers of your heart. When the premature, or early, signal tells the heart to contract, there may not be much blood in the heart at that moment. That means there's not much blood to pump out.

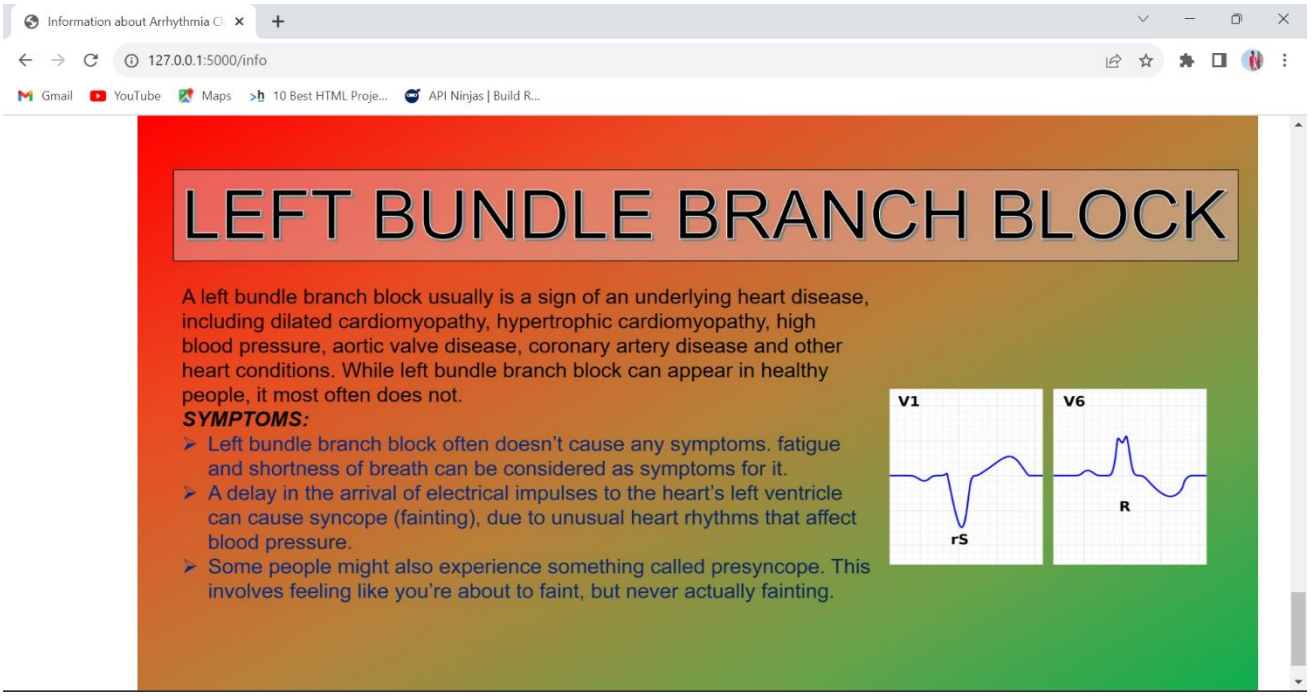
SYMPTOMS:

- You may not have any symptoms. If you do, your symptoms may include heartbeats that:
- Have a lot of force now and then.
- Skip.
- Pound (palpitations).
- You may also have anxiety or shortness of breath.

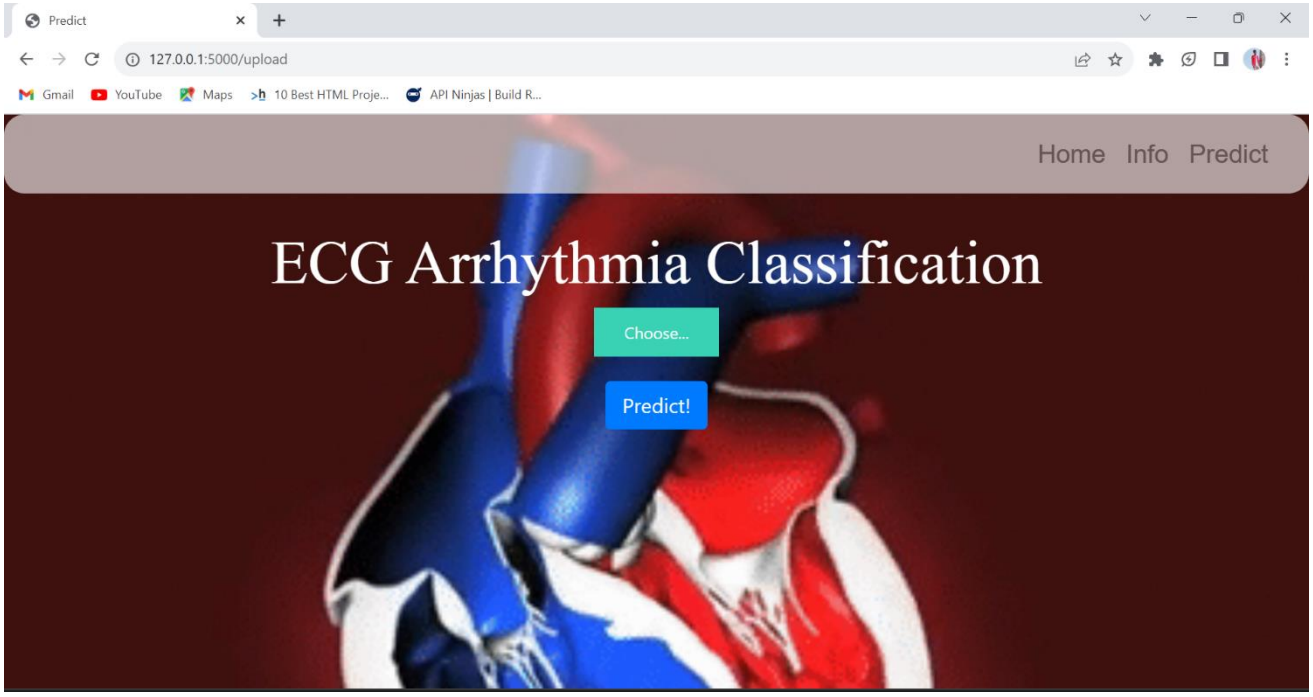


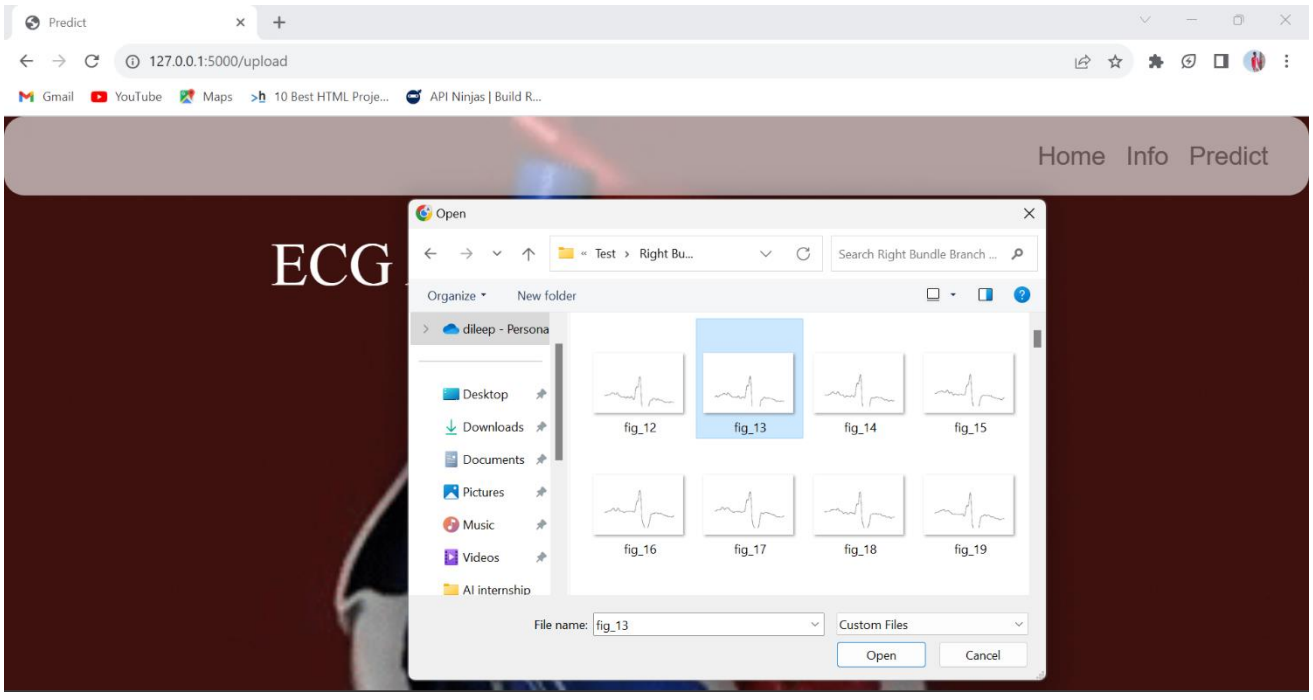

Information about Arrhythmia Cl
127.0.0.1:5000/info
Gmail
YouTube
Maps
10 Best HTML Proje...
API Ninjas | Build R...
PREMATURE VENTRICULAR CONTRACTION
Premature ventricular contractions (PVCs) are extra heartbeats that begin in one of the heart's two lower pumping chambers (ventricles). These extra beats disrupt the regular heart rhythm, sometimes causing a sensation of a fluttering or a skipped beat in the chest.
SYMPTOMS:
Fluttering.
Pounding or jumping.
Skipped beats or missed beats.
Increased awareness of the heartbeat.

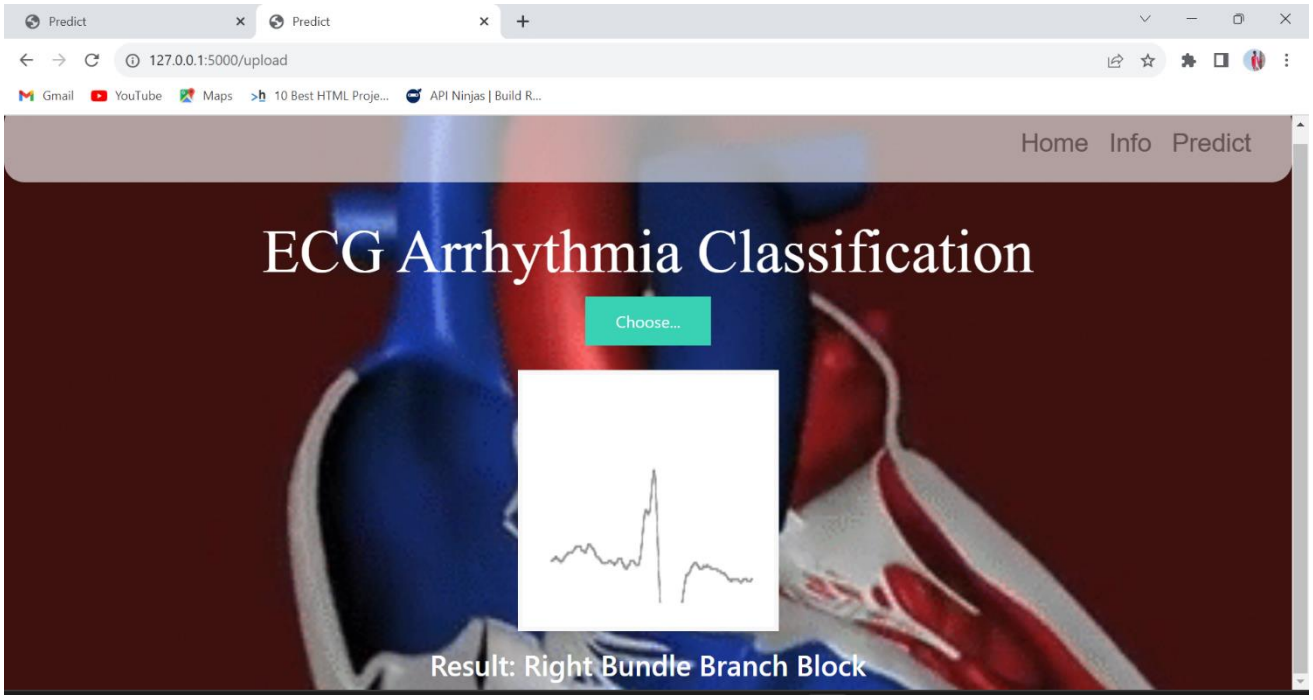

Information about Arrhythmia Cl
127.0.0.1:5000/info
Gmail
YouTube
Maps
10 Best HTML Proje...
API Ninjas | Build R...
RIGHT BUNDLE BRANCH BLOCK
Right bundle branch block is an obstacle in your right bundle branch that makes your heartbeat signal late and out of sync with the left bundle branch, creating an irregular heartbeat.
SYMPTOMS:
In most people, bundle branch block doesn't cause symptoms. Some people with the condition don't know they have bundle branch block.
Rarely, symptoms of bundle branch block may include fainting (syncope) or feeling as if you're going to faint (presyncope).
When to see a doctor?
If you've fainted, see a health care provider to rule out serious causes.
If you have heart disease or have been diagnosed with bundle branch block, ask your provider how often you should have follow-up visits.
V1
V6
rSR
qRs


Information about Arrhythmia Cl
127.0.0.1:5000/info
Gmail
YouTube
Maps
10 Best HTML Proje...
API Ninjas | Build R...
LEFT BUNDLE BRANCH BLOCK
A left bundle branch block usually is a sign of an underlying heart disease, including dilated cardiomyopathy, hypertrophic cardiomyopathy, high blood pressure, aortic valve disease, coronary artery disease and other heart conditions. While left bundle branch block can appear in healthy people, it most often does not.
SYMPTOMS:
Left bundle branch block often doesn't cause any symptoms. fatigue and shortness of breath can be considered as symptoms for it.
A delay in the arrival of electrical impulses to the heart's left ventricle can cause syncope (fainting), due to unusual heart rhythms that affect blood pressure.
Some people might also experience something called presyncope. This involves feeling like you're about to faint, but never actually fainting.
V1
rS
V6
R

Predict.html






Predict
Predict
127.0.0.1:5000/upload
Gmail
YouTube
Maps
10 Best HTML Proje...
API Ninjas | Build R...
Home
Info
Predict
ECG Arrhythmia Classification
Choose...
Result: Right Bundle Branch Block

Advantages, Disadvantages and Applications

Advantages

- The proposed model predicts Arrhythmia in images with a high accuracy rate of nearly 96%
- The early detection of Arrhythmia gives better understanding of disease causes, initiates therapeutic interventions and enables developing appropriate treatments.

Disadvantages

- Not useful for identifying the different stages of Arrhythmia disease.
- Not useful in monitoring motor symptoms

Applications

- It is useful for identifying the arrhythmia disease at an early stage.
- It is useful in detecting cardiovascular disorders

Conclusion

- Cardiovascular disease is a major health problem in today's world.

The early diagnosis of cardiac arrhythmia highly relies on the ECG.

- Unfortunately, the expert level of medical resources is rare, visually identify the ECG signal is challenging and time-consuming.
- The advantages of the proposed CNN network have been put to evidence.
- It is endowed with an ability to effectively process the non-filtered dataset with its potential anti-noise features. Besides that, ten-fold cross-validation is implemented in this work to further demonstrate the robustness of the network.

Future Scope

For future work, it would be interesting to explore the use of optimization techniques to find a feasible design and solution. The limitation of our study is that we have yet to apply any optimization techniques to optimize the model parameters and we believe that with the implementation of the optimization, it will be able to further elevate the performance of the proposed solution to the next level.

References

- https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/
https://www.mathworks.com/help/deeplearning/ref/nnet.cnn.layer.convolution2dlayer.html;jsessionid=0a7e3bc26fabda07a5032030294b
- https://smartinternz.com/externship_dyn/1/artificial-intelligence

Project Links

- **Github Link : https://github.com/Dileep0509/Classification-of-Arrhythmia-by-Using-Deep-Learning-with-2-D-ECG-Spectral-Image-Representation**

- **Demo Link: https://youtu.be/u8bpsNchyxQ**

THE END