ECG Arrhythmia classification using CNN

Introduction

1. Overview:

Cardiovascular disease is one of the major diseases that threaten human life. According to reports by the world health organization, cardiovascular diseases (CVDs) mortality ranks first in all causes of death today.

Arrhythmia is an important group of diseases in cardiovascular disease. Arrhythmia can occur on its own or with other cardiovascular diseases. The diagnosis of arrhythmia mainly depends on the ECG (electrocardiogram). ECG (electrocardiogram) is an important modern medical tool that records the process of cardiac excitability, transmission, and recovery. Automatic detection of irregular heart rhythms from ECG signals is a significant task for the automatic diagnosis of cardiovascular disease.

The main types of arrhythmia are bradyarrhythmias; premature, or extra, beats; supraventricular arrhythmias; and ventricular arrhythmia.

It is very difficult for the human eye to identify the type of arrhythmia ,so we need some automation in the detection of the type of arrhythmia.

1. Purpose:

The classification of electrocardiogram (ECG) signals is very important for the automatic diagnosis of heart disease.  Owing to recent advances in artificial intelligence, it has been demonstrated that deep neural network, which trained on a huge amount of data, can carry out the task of feature extraction directly from the data and recognize cardiac arrhythmias better than professional cardiologists and makes their task easier. In this project we do ECG arrhythmia classification using two-dimensional (2D) deep convolutional neural network (CNN). The time domain signals of ECG, belonging to six heart beat types including normal beat (NOR), left bundle branch block beat (LBB), right bundle branch block beat (RBB), premature ventricular contraction beat (PVC), and atrial premature contraction beat (APC) and Ventricular Fibrillation. It makes the classification of types of arrhythmia easier and more accurate.

Literature survey

1. Existing problem

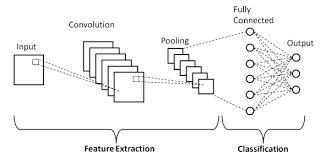
Current medical screening and diagnostic procedures have shifted toward recording longer electrocardiogram (ECG) signals, which have traditionally been processed on personal computers (PCs) with high-speed multi-core processors and efficient memory processing. Later the cardiologists have to study the ECG and come to conclusions about the type of arrhythmia that the patient might make. This method is time consuming and may lead to wrong predictions because of human errors. So, we use convolutional neural networks to classify the data its types and predict accurate results.

1. Proposed solution

The proposed solution is to use a deep learning model that is convolution neural networks which is used in image analysis.The ECG spectrogram images are fed into the proposed deep two-dimensional convolutional neural network (CNN) model. With these obtained ECG spectrogram images, classification of the six ECG types is performed in the 2D-CNN classifier automatically and intelligently. The user has to load an ECG image and the model predicts the type of arrhythmia it is.

Theoretical Analysis

1. Block diagram



b.Software used

Anaconda (jupyter notebook and spyder IDE)

Experimental investigations

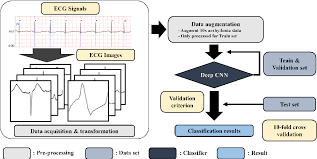
The first step in developing the project is Data Collection. Here train and test folders for all the types of arrhythmia is created. The next step is to preprocess the data by augumenting the image features by using ImageDataGenerator library for train and test data. We then load the dataset and then apply the augumented features to train and test sets.

We then import the required libraries for developing a convolution neural network like Sequential, dense layer ,MaxPooling2D ,flatten and convolution2D.

We then initialize the model , then add the convolution layer, pooling layer and flatten layer which acts as input for the artificial neural network. Then we apply dense layers that includes hidden layers and output layer. Then we configure the learning and save the model.

The next step is to test the model for which we first load the saved model and then load an image after preprocessing and then obtain the prediction.

For the application part we use spyder IDE to develop a flask application and save the model in the folder containing the application. The UI is then built and we can now load an image in the UI and obtain the result.

Flowchart

Results

Using jupyter notebooks the model is trained and tested , the model is then saved to use it in the application.

We use Spyder IDE to develop the flask application . Using the UI the user can load the image by clicking on the choose button and the model will classify the image and display the predicted output.

Advantages

It is easier to predict the results using CNN as it extracts the important features and gives that as the input to ANN which is then fed into dense layers and gives the output.

It gives faster results.

It reduces human effort.

It gives accurate results.

Disadvantages

For smaller amount of training data, DL methods face the overfitting problem since the model highly pay attention to training data and do not generalize well for the test data. Thus, shallow techniques provide better performance on small amount of data samples.

It consumes a lot of time while training the model.

Most of the DL methods are disposed to learn the peculiarities such as the noise of ECG signal leading to inaccurate results. The problem is pronounced with the size of dataset.

Conclusions

The project aims in developing a deep learning method to classify various types of arrhythmia using the ECG using CNN. It is also concluded that the use of a proper type of Deep Learning method can considerably improve the classification performance for the corresponding application. This method also helps in providing faster and accurate results.

Future research trend

According to the best classification methods represented , CNN-based have proven to be effective for arrhythmia classification. Recent trend of research in this scope shows that dynamic classification methods that are capable to learn both short and long term contents of the signal in an efficient way, would be employed for such applications. CNN has shown excellent performance in classifying different types of arrhythmia. This powerful method would be one of the most efficient learning tool for this application.

Bibliography

[Abadi, Barham, Chen, Chen, Davis, Dean, Devin, Ghemawat, Irving, Isard, et al., 2016](https://www.sciencedirect.com/science/article/pii/S2590188520300123#bbib0001)

M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, ..., M. Isard, et al.

**Tensorflow: A system for large-scale machine learning**

12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16) (2016), pp. 265-283

[View Record in Scopus](https://www.scopus.com/inward/record.url?eid=2-s2.0-85014235798&partnerID=10&rel=R3.0.0)[Google Scholar](https://scholar.google.com/scholar_lookup?title=Tensorflow%3A%20A%20system%20for%20large-scale%20machine%20learning&publication_year=2016&author=M.%20Abadi&author=P.%20Barham&author=J.%20Chen&author=Z.%20Chen&author=A.%20Davis&author=J.%20Dean&author=M.%20Isard)

[Acharya, Fujita, Lih, Hagiwara, Tan, Adam, 2017a](https://www.sciencedirect.com/science/article/pii/S2590188520300123#bbib0002)

U.R. Acharya, H. Fujita, O.S. Lih, Y. Hagiwara, J.H. Tan, M. Adam

**Automated detection of arrhythmias using different intervals of tachycardia ecg segments with convolutional neural network**

Information Sciences, 405 (2017), pp. 81-90

[Article](https://www.sciencedirect.com/science/article/pii/S0020025517306539)[Download PDF](https://www.sciencedirect.com/science/article/pii/S0020025517306539/pdfft?md5=99527ecf71f1ac297b80c0b87fcda35f&pid=1-s2.0-S0020025517306539-main.pdf)[View Record in Scopus](https://www.scopus.com/inward/record.url?eid=2-s2.0-85017454416&partnerID=10&rel=R3.0.0)[Google Scholar](https://scholar.google.com/scholar_lookup?title=Automated%20detection%20of%20arrhythmias%20using%20different%20intervals%20of%20tachycardia%20ecg%20segments%20with%20convolutional%20neural%20network&publication_year=2017&author=U.R.%20Acharya&author=H.%20Fujita&author=O.S.%20Lih&author=Y.%20Hagiwara&author=J.H.%20Tan&author=M.%20Adam)

Appendix

1. Source code

Application code

import numpy as np

import os

from tensorflow.keras.models import load\_model

from tensorflow.keras.preprocessing import image

import tensorflow as tf

from flask import Flask , request, render\_template

from werkzeug.utils import secure\_filename

from gevent.pywsgi import WSGIServer

app = Flask(\_\_name\_\_)

model = load\_model("mymodel.h5")

@app.route('/')

def index():

return render\_template('base.html')

@app.route('/predict',methods = ['GET','POST'])

def upload():

if request.method == 'POST':

f = request.files['image']

#print("current path")

basepath = os.path.abspath('')

#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)

preds = model.predict\_classes(x)

print("prediction",preds)

index=['Left Bundle Branch Block','Normal','Premature Atrial Contraction','Premature Ventricular Contractions','Right Bundle Branch Block','Ventricular Fibrillation']

text = "the predicted arrhythmia is : " + str(index[preds[0]])

return text

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug = False,threaded=False)

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>ECG arrhythmia Recognition System</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>

.bg-dark {

background-color: #42678c!important;

}

#result {

color: #0a1c4ed1;

}

</style>

</head>

<body>

<nav class="navbar navbar-dark bg-dark">

<div class="container">

<a class="navbar-brand" href="#">Arrhythmia Recognition Using CNN</a>

</div>

</nav>

<div class="container">

<div id="content" style="margin-top:2em">

<div class="container">

<div class="row">

<div class="col-sm-6 bd" >

<h3>Arrhythmia Recognition </h3>

<br>

<p>Arrhythmia

Also known as Dysrhythmia

An arrhythmia is a problem with the rate or rhythm of the heartbeat. During an arrhythmia, the heart can beat too fast, too slowly, or with an irregular rhythm. When a heart beats too fast, the condition is called tachycardia. When a heart beats too slowly, the condition is called bradycardia.

Arrhythmia is caused by changes in heart tissue and activity or in the electrical signals that control your heartbeat. These changes can be caused by damage from disease, injury, or genetics. Often there are no symptoms, but some people feel an irregular heartbeat. You may feel faint or dizzy or have difficulty breathing.</p>

<img src="https://t4.ftcdn.net/jpg/02/08/48/05/360\_F\_208480571\_Ws4f7xUVkxWmbNcjgGSVxrxEYR3hOpFk.jpg" style="height:150px"class="img-rounded" alt="ecg">

</div>

<div class="col-sm-6">

<div>

<h4>Please upload an ECG image</h4>

<form action = "http://localhost:5000/predict" id="upload-file" method="post" enctype="multipart/form-data">

<label for="imageUpload" class="upload-label">

Choose...

</label>

<input type="file" name="image" id="imageUpload" accept=".png, .jpg, .jpeg">

</form>

<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>

<div class="loader" style="display:none;"></div>

<h3>

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

</h3>

</div>

</div>

</div>

</div>

</div>

</div>

</body>

<footer>

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

</footer>

</html>

UI Window Screenshot

