ECG Spectogram Classification by Using CNN Model
ELEC 447/547 Biomedical Signal Processing

September 29, 2022

Contents

List of Figures

List of Tables

1 Introduction

In our project, we use deep learning algorithms to classify ECG signals. Convolutional Neural Network is implemented. Algorithm is trained and signals are labeled as normal, atrial fibrillation, other signals. Detecting atrial fibrillation is a necessity for human life, since it causes health problems including death. Our main objective is, with an accessible analysis of ECG signals, identifying ECG signals if the patient has Atrial Fibrillation or not.

2 Methods

In this section, the motivation behind the project is discussed along with the dataset, approaches and various signal processing & machine learning techniques are discussed.

2.1 Atrial Fibrillation

Heart pumps blood to body, it contracts and relaxes. Heart has four chambers, two upper chambers which are atria and two lower chambers which are ventricles. During each beat, those chambers contract or squeeze. Contraction of the chambers cause blood move along the body. Heart's electrical system controls the timing of the pumps. During regular pumps, when it is working in the regular way, beats are steady. This is called normal sinus rhythm. In normal sinus rhythm, heart rate is between 60 and 100 beats per minute. Atrial Fibrillation (AFib) occurs when beats are not steady, heartbeat is irregular. AFib can lead heart related complications such as stroke, heart failure. In Figure 1, the difference between a signal of normal patient and a signal of a patient with AFib can be examined.

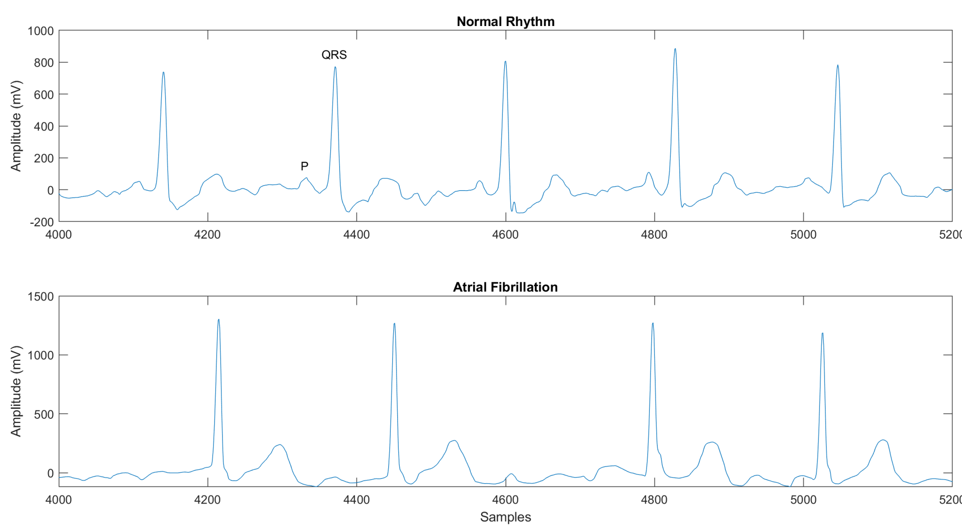

Figure 1: Normal Rhythm vs Atrial Fibrillation

Standard 12-lead ECG is used for diagnosis of AFib. AFib is the most common type of irregular heart beats. In AFib, heart's electrical system does not work properly. Upper chamber of the heart does not beat regularly. Blood flow through the ventricles cannot be done well. This causes irregular heartbeat.

During AFib, rhythm is irregularly irregular and P waves are absent. Isoelectric baseline is absent. Ventricular rate is variable. There may fibrillatory waves which mimic P waves which leads to misdiagnosis.

2.2 Dataset

The dataset that have been used throughout the project is *AF Classification from a Short Single Lead ECG Recording* set, shared on *PhysioNet*. It includes four different ECG signal types which are taken from patients who have Atrial Fibrillation rhythms, normal rhythms along with the noisy signals and others. In Figure 2, samples of each signal type can be examined. The dataset has 8528 recordings with average duration of 32 seconds long.

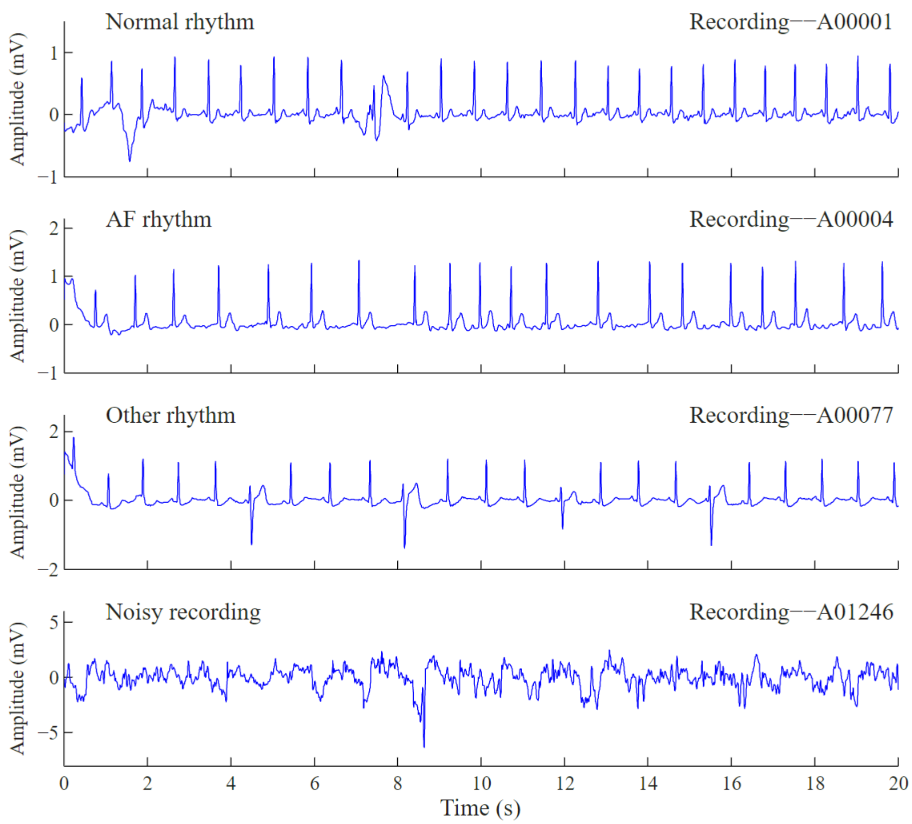

Figure 2: Samples of Each ECG Waveform Types [1]

Signals were sampled as 300 Hz. Although each signal has different sample size, most of the signals has 9000 sample size which can be seen in Figure 3.

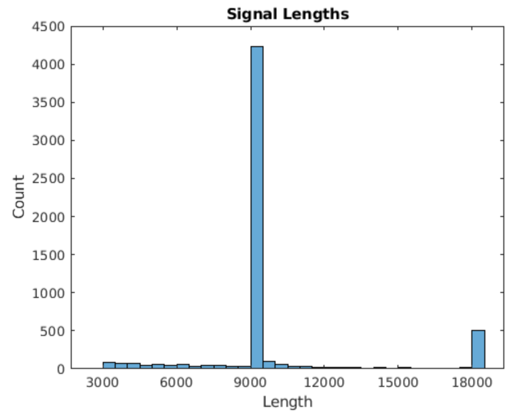

Figure 3: Distribution of Sample Sizes [2]

2.3 Preprocessing and Segmenting

Preprocessing the signals was not necessary since the model knows and labels noisy and other type of signals. Yet, we had to process the dataset due to memory issues.

As mentioned in Figure 3, we have an average of 9000 sample sizes. Therefore, the first approach it was used is segmenting the signals according to 9000^{th} sample. Signals longer than 9000 sample size are divided from 9000^{th} sample and the rest of them added and used as another signals. By this way, we have extended the dataset that we have used. However, we have faced memory issues while training the model. Thus, with the guidance of the professor, we cut out each signal that is longer than average sample size to reduce the feature size of the input.

2.4 Spectogram

Converting the signals into spectograms is the most crucial part of the project. A spectogram is a visual representation of the spectrum of frequencies of a signal as it varies as time goes by. The idea is to take each signal, convert into spectogram images of frequencies by time and provide them as inputs into the implemented model. A sample for spectogram is provided in Figure 4. It shows a sample by normalized frequency spectogram. Another motivation is to try different type of spectograms on the model to examine if the accuracy is affected, or not. Taking the spectograms as images is the fundamental approach. The *spectogram* function of MATLAB returns an array of Fourier transformation of the signal, yet, the implemented model is not capable of handling such input. Thus, we have tried to convert spectograms into images on Python.

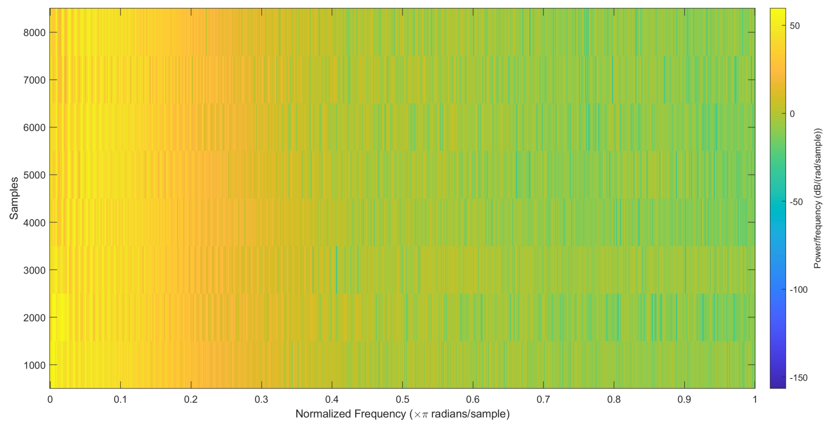

Figure 4: Spectogram of Normal Labeled ECG Signals [2]

2.5 Convolutional Neural Network

As the machine learning model of the project, Convolutional Neural Network (CNN) is implemented. The designed CNN model takes spectogram images as inputs and labels each input as one of the four outputs. A similar idea is presented in the article, *SleepNET*, for EEG signals with 6-channels [3]. We tried to implement for one channel of ECG signal with RGB images. A simple demonstration of the model can be observed in Figure 5.

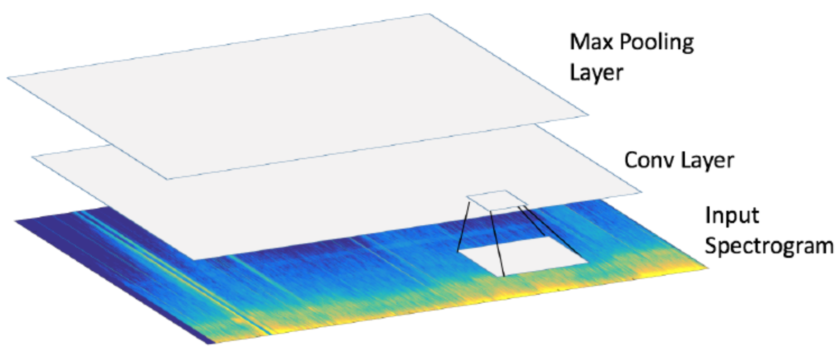

Figure 5: CNN Approach on Spectogram Input [3]

The diagram of implemented model can be examined in Figure 6. Inputs go through three layers. Each layer has a convolutional layer folowed by batch normalization, rectified linear unit (ReLU) and max pooling. According to *SleepNET*, the ideal window size of the convolutional layer is 3-by-3 which gives the most effective performance on the training [3]. Therefore, we have also implemented 3-by-3 window for the convolutional layer. After the three convolutional layer has passed, output of the layers goes through a fully connected layer and a soft-max function followed by classification output. As the training executed successfully, four classed classification of an image is expected.

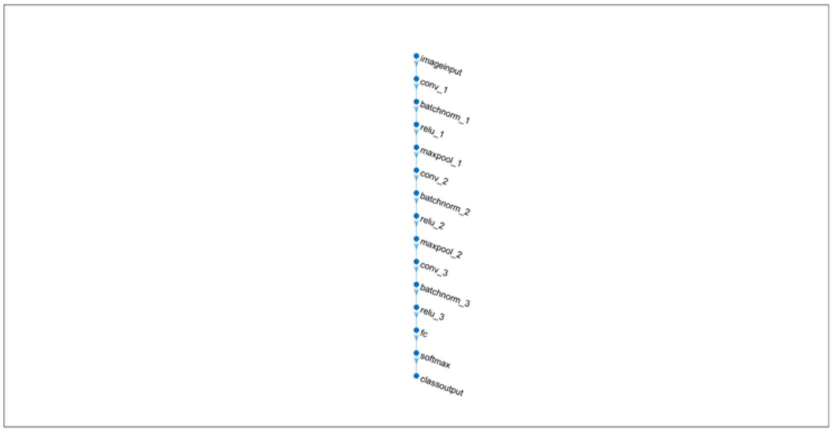

Figure 6: Designed CNN Model

2.6 Final Adjustments and Methods

As mentioned previously, we have faced both technical and memory problems throughout the project. As the results of discussion with the professor, a critical reduction of feature size, signal length and the type of the applied method are committed.

In this manner, we have worked on following setups and approaches.

- 20 signals are used. 60% of them (12 signals) are used as training data while the rest is used as validation.
- After processing the signals on MATLAB, we have moved on Python language. All the implementations, conversions and training are done via Python and TensorFlow and Keras.
- Conversion into the spectograms is done with the he Short-Time Fourier transform of the input signal which contains an estimate of the short-term, time-localized frequency content of the input.
- After converting to spectograms, we have implemented and applied *Continuous Wavelet Transform*, reshaped the output and returned as an RGB image.
- Once we got the RGB image, we have reshaped each image as 64x64x8 to feed into the CNN model.
- Since the number of signals is critically reduced, the model trained for only 3 class labels for the output which are *Normal*, *Atrial Fibrillation* and *Other*.

- CNN model did not change drastically. All the paddings, window sizes and functions stay the same. We removed one of the layers and added two more fully connected layers after flattening the output of the latest max pooling. Briefly, once the input feeded, it passes through two convolutional layers followed by max poolings. After the layers, we are flattening the output and then connecting it to three fully connected layers with softmax activtion function. The final CNN model can be seen in Figure 7.

3 Results and Discussion

The training process is done with 100 epochs. Accuracy and loss of the model through epochs can be examined in Figure 8 and 9.

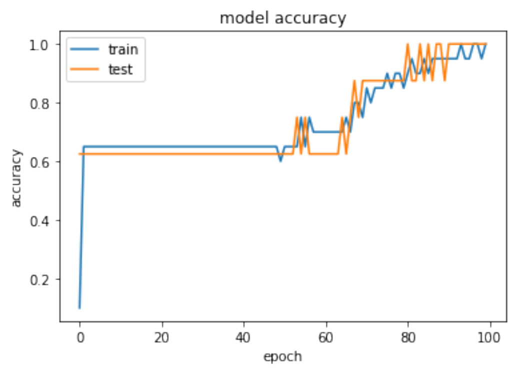

Figure 8: Accuracy through Epochs Graph

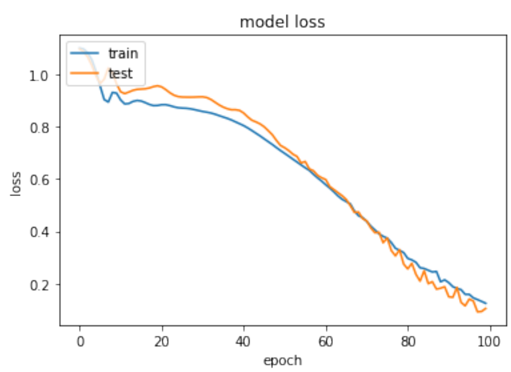

Figure 9: Loss through Epochs Graph

Accuracy is used for measuring performance while the loss function is used for optimizing the model. Loss is interpretation of how well the model is doing in training and test. It is a summation

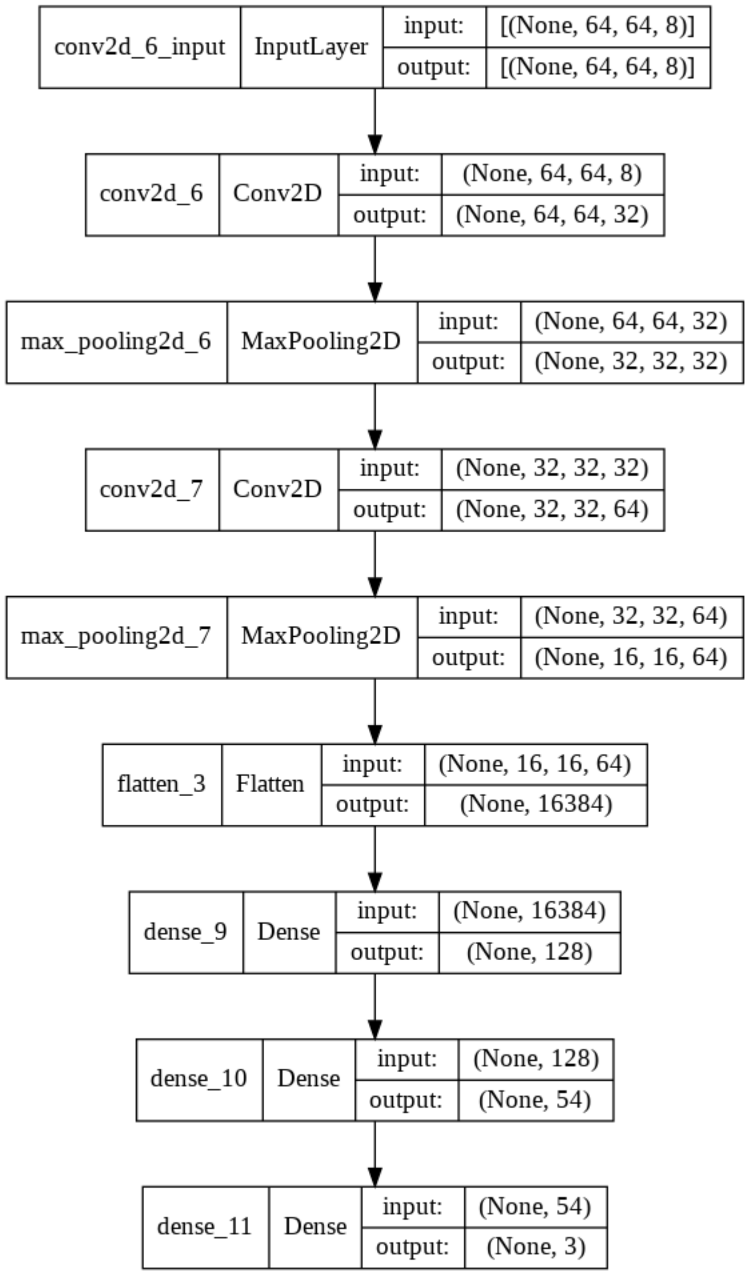

Figure 7: Model Diagram of Finalized CNN

		Accuracy
Epoch Size	**10**	0.625
	20	0.625
	30	0.65
	50	0.75
	75	0.95
	100	0.95

Table 1: Number of Epochs vs Accuracy

of the errors made in each sample of training. Thus, according to our loss graph, Figure 9, we have faced with a greater performance than we expected although we have relatively small sized dataset.

In Table 1, accuracy throughout the changing epoch size can be seen. We have observe that the greater the number of epochs, the better performance of the model. However, as mentioned before, we have used 20 signals, which is crucially low number of feature for such model.

For the solution of lack of training and more accurate model, input data should be extended and applied.

References

[1] G. Clifford, C. Liu, Moody, B. L.-W. Lehman, I. Silva, Q. Li, A. Johnson, and R. Mark, "Dataset: Af classification from a short single lead ecg recording: the physionet computing in cardiology challenge," 2017. Available at https://physionet.org/content/challenge-2017/1.0.0/.

[2] M. MathWorks, "Classify ecg signals using long short-term memory networks." Available at https://www.mathworks.com/help/signal/ug/classify-ecg-signals-using-long-short-term-memory-networks.html.

[3] S. Biswal, J. Kulas, H. Sun, B. Goparaju, M. B. Westover, M. Bianchi, and J. Sun, "Sleepnet: Automated sleep staging system via deep learning," 2017.

[4] A. Rakhecha, "Useful plots to diagnose your neural network." Available at https://towardsdatascience.com/useful-plots-to-diagnose-your-neural-network-521907fa2f45l.