Cardiac Arrhythmia Detection From PCG Signals Using Machine Learning

Manoj Athreya A[1], Avani H S[2], Pooja[3], Madhu S[4], K Paramesha[5]
Department of Computer Science and Engineering
Vidyavardhaka College of Engineering, Mysuru, Karnataka, India
[1]manoj2375@gmail.com, [2]avanirao0811@gmail.com,
[3]poojagowda2237@gmail.com,[4]madhu1.mys@gmail.com , [5]paramesha@vvce.ac.in

Abstract: Cardiac Arrhythmia refers to a group of conditions that causes the heart to beat too slow or too fast. It is one of the major problems of the heart which needs to be diagnosed at the earliest, as it takes more time for doctors to detect and provide medication. We find different types of arrhythmias; for slow heartbeat it is called bradycardia, for fast heartbeat it is called tachycardia. During initial stages of Cardiac Arrhythmia, doctors need to carefully examine the heartbeats precisely from different locations of the body. Manually evaluating these fundamental heart sounds (FHSs) for each and every patient is time consuming. Thus automating the procedure by using Machine Learning techniques to classify heart sound recordings would help in overcoming this problem. The objective is to take the Phonocardiogram (PCG) signals for evaluation, convert it to spectrogram images and train a convolution neural network model to predict the outcome. Then given a new PCG recording it will be able to classify as normal or abnormal. Hence the process of detecting arrhythmia is simplified and saves people's lives.

Keywords: Cardiac Arrhythmia, Spectrograms, PCG, Convolution Neural network, Machine Learning, Transfer Learning.

Introduction

A normal resting heart rate for an adult ranges from 60 to 100 beats per minute. Certain circumstances cause the heart to beat in an irregular rhythm, these are medically known as arrhythmias. The heart rhythm is regulated by a node at the top of the head, the sinus node is considered the one that activates an electrical signal traveling through the heart causing the heart to beat and to pump blood in the body. Surplus electrical activity in the upper or lower portion of the heart indicates the heart is not pumping well. Symptoms of Cardiac Arrhythmia are increasing heartbeat called tachycardia and decreasing heartbeat called bradycardia. Few of the other symptoms are anxiety, fatigue,chest pain, lightheadedness or dizziness, sweating and even fainting. Neglecting these symptoms and not treating arrhythmia in the initial stage itself may endup in heart attack and also heart failure. Inorder to classify heartbeats into normal and abnormal and to make life of doctors easy and save lives of people, we came up with an approach. There are many existing systems which will classify heartbeat using electrocardiogram (ECG). But we have come up with a new approach where we will be using phonocardiogram (PCG) to classify heart beat. The basic idea is to classify heartbeat using sound recording of heartbeat. Firstly we will convert heart sound recording (wav file) to spectrogram images and then we will use these spectogramic images to train the model. We are using convolutional neural networks to train the model. After training, our model will be able to predict the given PCG recording as normal or abnormal. The samples were obtained from both abnormal cases and normal subjects, giving a variety of sources of signal. The method was to cover up the data in spectrogram images, and then use a Deep Convolutional Networks preparation. We implemented progressively different algorithms to interpret the images to get a better outcome. We applied, too specific strategies for augmenting results. We started with a simple CNN first, and then shifted to a more layered one In order to further refine and improve the result we use the 'Transfer Learning' strategy. We used the VGG16 model pre-trained, and extracted image attributes, and then flew through certain attributes using our own classifier. We have trained the images from scratch to custom model based on VGG16 model architecture for better results. [1-5]

In the next section, we will give insight into the different research works done on the topic. Section III explains the mathematical model incorporated. Section IV deals with the details of the project creation and its working. Section V deals with the results obtained and Section VI concludes.

II. Literature Survey

Heart Arrhythmia is one of the principal ailments that we have come over in the cardiology field. Its examination and recognition assumes a significant job and to cure the patients. The reference papers alluded in the overview utilized many AI procedures and calculations to decide arrhythmia. They are Gray-Level Co-Occurrence Matrix of ECG, Hidden Markov model, Biomedical sign handling, Empirical Mode Decomposition, Pan Tompkins calculation, Backpropagation neural system, CNN based Generalizable Information Fusion, Multilayer perceptron, two-dimensional profound CNN highlights based exchange learning, time-recurrence joint conveyance of ECG, Bayesian conviction organize, J48 OneR and Naïve Bayes. The dataset considered for the assessment is taken from MIT-BIH arrhythmia database which had ECG chronicles as a noticeable factor which was taken care of as a contribution to the model to assess the outcome.[6]

An automatic cardiac arrhythmia classification system with wearable electrocardiogram, presents a classification and monitoring system for the automatic wearable electrocardiogram (ECG) with stacked denoising auto-encoder (SDAE). To obtain the ECG data, they used a wearable device with wireless sensors, and sent those ECG data to a Bluetooth 4.2 computer. Then the automated cardiac arrhythmia classification system classifies certain ECG details. [7]

Automatic cardiac arrhythmia detection and classification using vector cardiograms and complex networks, seeks to introduce fresh perspectives into the extraction approaches for diagnosis of heart arrhythmias, as well as the ranking schemes. They looked at the probability of usage of Vector Cardiograms (VCG) and Electrocardiograms (ECG) to get appropriate pulse information on table MIT-BIH. They have also used the Support Vector Machine (SVM) and achieved the accuracy of 84.1%. [8-9]

Arrhythmia classification on ECG using Deep Learning, is an approach to signal classification based on intelligent electrocardiograms (ECG) using Deep Learning (DL) is being developed. They use Deep Learning algorithms to identify different forms of arrhythmia. Here they use the Convolutional Neural Network (CNN). A DL algorithm that is effective in signal detection and has 93.6 percent accuracy. [10-12]

Heart rhythm abnormality detection from PCG signal, in the initial step performed Signal Quality Evaluation and Component Extraction in which we investigate numerous data models to identify the connection between the functionality and the outcomes, producing low performance.[6] Using a Logistic Regression-HSMM, audio files are segmented into Systolic and Diastolic phases during the second stage. These systems and diastoles segments are then analysed individually, and the extraction of individual features is carried out. A ton of de-noising is often performed in the segmentation process eliminating the surrounding sounds. This strategy provides 79 percent precision. [13-15]

III. Mathematical Model

Categorical cross entropy is the metric to determine the model's performance on the entire dataset which is logarithmic loss. The formulae for this is given below.

$$\text{log-loss} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(p_{ij}) \qquad \text{................(i)}$$

This can be described as negative; the log likelihood of the model given all the observations are chosen separately from a distribution that places the predicted probability mass on the corresponding class, for all observations. In our case, each recording is already labeled, a set of predicted probabilities is considered as outcome. Here log is the natural logarithm, 'm' is the number of images class labels, 'n' is the number of images in test set i.e 2, $Y_{i,j}$ is 1 if observation belongs to class and 0 otherwise, and $P_{i,j}$ is the predicted probability that observation belongs to given class.

Metric	**Formula**
True positive rate, recall	$\frac{TP}{TP+FN}$
False positive rate	$\frac{FP}{FP+TN}$
Precision	$\frac{TP}{TP+FP}$
Accuracy	$\frac{TP+TN}{TP+FN+TN+FP}$
F-measure	$\frac{2 \,.\, precision \,.\, recall}{precision+recall}$

Since the dataset is imbalanced as the numbers of abnormal samples is less than that of normal samples in the training dataset, accuracy is not only the metric we considered. Hence the model was evaluated on other metrics such as recall which is the ratio of correct positive predictions made out of the actual total that were positive. Precision which can be described as the ratio of correct positive predictions made out of the total positive predictions made. F-Beta score which is given by the weighted average of precision and recall.

IV. Methodology

Dataset

The dataset used for this project is freely available as part of the PhysioNet in Cardiology Challenge 2016 which focuses on automatic classification of normal or abnormal phonocardiogram (PCG) recording. The training data consists of PCG signals of varying length, between 5s to 120s sampled at 2000 Hz and are provided in .wav format. Therefore, we choose the training set to be about 2000 labeled recordings and the validation set to be 686 recordings. The dataset is based on time-frequency features, to capture pitch and intensity of the recordings. Due to its robustness and interpretability, spectrograms are a visual way of representing the signal strength or loudness of a signal. Therefore, raw wav files need parsed into spectrograms first and the class labels are extracted from header (.hea) files. Splitting the dataset into training and testing sets is also automatically taken care of by the script.

To convert .wav recording to spectrogram, we use python code and a few library functions. We import the wav file to the program. We take the data in the form of samples with respect to time and frequency on the x and y axis respectively. We use the function plt.pcolormesh to indicate color to spectrogram which helps to improve the classification, pt.imshow(spectrogram) function to display the spectrogram with respect to each wav file provided as a sample. The below figure 1 and 2 depicts the abnormal and normal spectrogram image after its conversion from wav file.

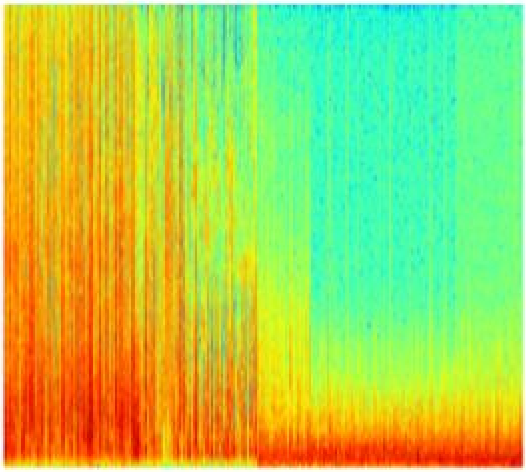

figure 1: An abnormal heartbeat

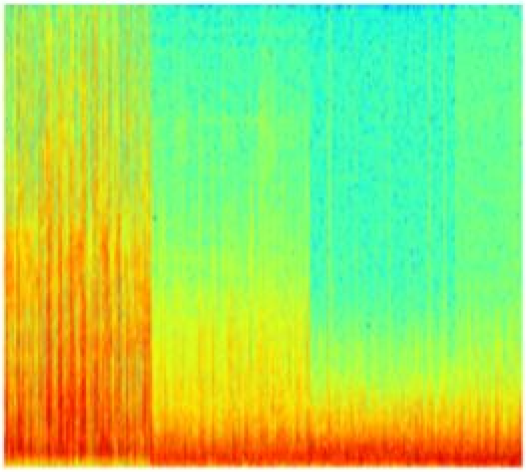

figure 2: A normal heartbeat

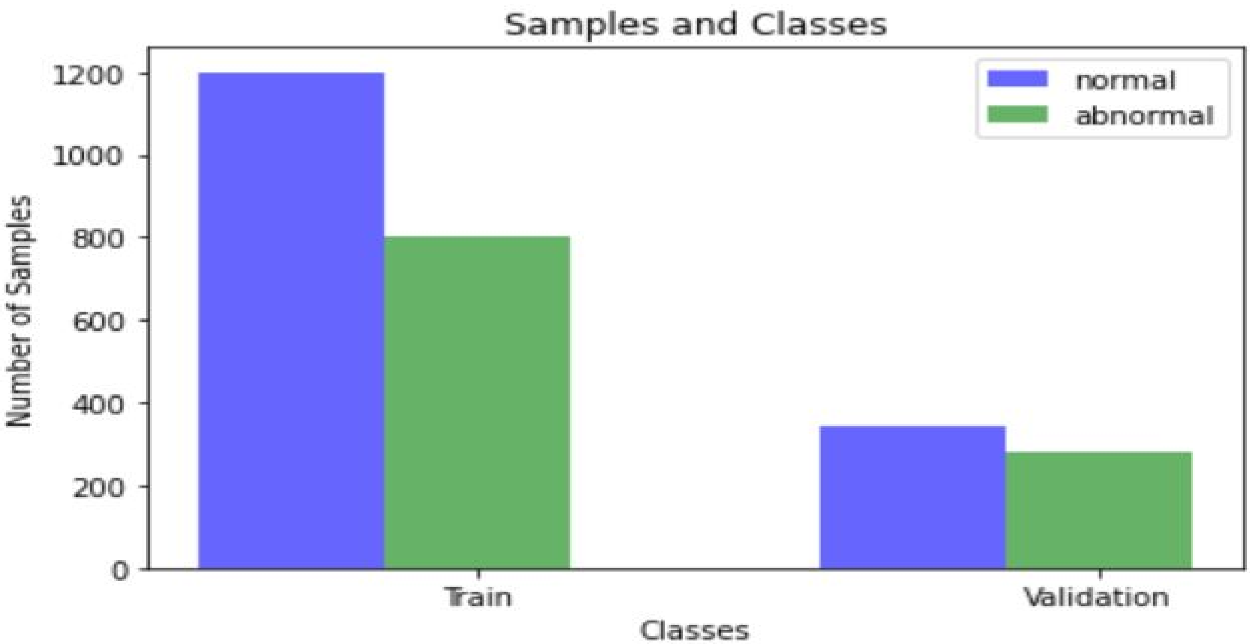

figure 3: The number of samples for each class in the training and validation set

The above figure 3 depicts the bar graph of training and validation samples in the dataset, which are used for training and prediction of the model.

Data Preprocessing

Data Augmentation will be important for our model so that it never sees the exact same picture twice since different spectrograms look similar. This helps prevent overfitting and the model generalizes better. For images, this could be done by rotating the original image, cropping it differently, changing lighting conditions, so for one image we can generate different sub-samples.

For this project, the following data augmentation techniques were applied:

• Resizing: All the images in the dataset were resized to 150x150x3 (width = height = 150 ; colour channel = 3) before feeding into the CNN.

• Normalize: Pixel values for all images were normalized between 0 and 1. This was done by subtracting the minimum pixel (i.e. 0) and dividing by maximum pixel value (i.e. 255). In Keras, this was done by setting the rescale attribute to 1/.255.

• Shear Transformation: Shear Transformation was applied to control the shear intensity of the input images. It was set to 0.2 using the shear-range attribute.

• Zooming: Randomly zooming inside images by setting zoom-range attribute to 0.2

• Flipping: Half of the training inputs were randomly flipped horizontally using the horizontal-flip attribute.

All the augmentation techniques were implemented using Keras Image Data Generator class with the above attributes to provide real-time data augmentation. Apart from these, the Image Data Generator class has the method flow from directory which it takes the path to a directory, and generates batches of augmented data. The data was looped over in batches indefinitely. It further provides us with the number of images in each set. The splitting up of data into training and testing sets was done manually before applying the data preprocessing steps.

Model Architecture

The focus of this project is to classify whether the patient has “normal” or “abnormal” heart sound from the Phonocardiogram (PCG) or heartbeat recordings fed to the model which would help doctors identify and further carry out the medication procedure. This is a supervised learning problem since we already know if the heart sound in the training dataset is normal or abnormal. The basic idea is to convert each heart sound recording (wav file) to a spectrogram image and train a Convolutional Neural Network over those images. Then given a new PCG recording, we will be able to classify it as normal or abnormal. The figure 4 depicts the workflow as mentioned.

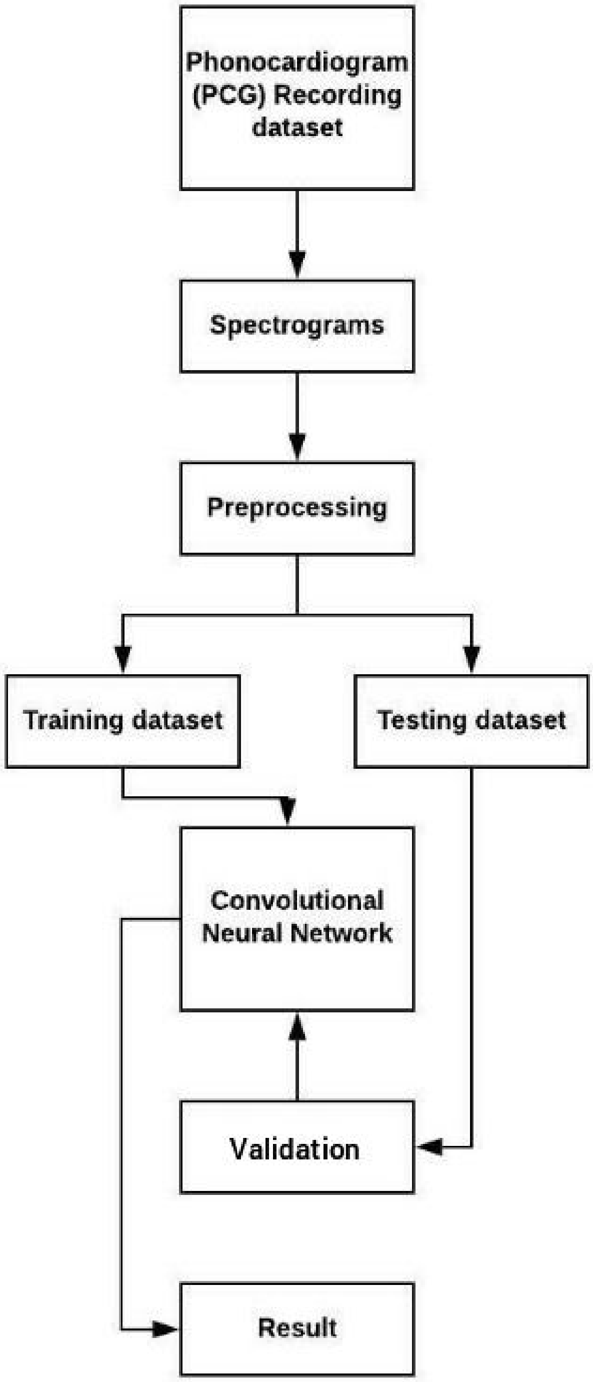

figure 4: Flowchart of model development

Implemented Model

The implemented model is built on a deep convolutional neural network. The stack for each block is as follows:- We first applied zero padding i.e, adding columns and rows of zeros at the top, left, bottom and right side of an image. Then convolutions and activation function are followed by the batch normalization layer. This way the activations of the previous layer at each batch are normalized i.e. mean activation is almost 0 and the activation standard deviation close to 1. After this we applied a max-pooling layer. This architecture has 5 such blocks. This is then followed by two fully-connected layers. We also used dropout to avoid overfitting. In the end we are left with a single unit and hence use sigmoid activation since this is a binary classification. Kernel size for zero padding, filter size, downscale size in each block is the same and is (1,1), (3,3) and (2,2) respectively. The number of filters for 1st, 2nd, 3rd, 4th and 5th layers are 64, 128, 256, 512, 512 respectively. After flattening the image i.e. converting our 3D feature maps to 1D feature vectors we apply the first fully connected layer which has 64 units and activation function. Finally, we apply sigmoid activation to the last unit to classify into normal or abnormal. The model is then saved as a file in (.h5) format and imported to the prediction function. The prediction function takes this model and a new spectrogram image to classify it as normal or abnormal.

We will be using Rectified Linear Unit (ReLU), Exponential Linear Unit (ELU) and Tangent Hyperbolic (Tanh) for the first, second and third model respectively with Sigmoid function at the end of the layers. To improve the original model, we decided to implement a pre-trained model VGG-16 which is commonly known as Transfer Learning. Model compilation is done by tensorflow with its efficient numerical libraries on a GPU. We defined Adam as the optimizer as it is an efficient stochastic gradient descent algorithm. To train the model we call the fit() function on the model and training is carried out over epochs where each epoch is split into batches. After training the model we can evaluate the model by calling the evaluate() function and passing the validation dataset. Thus the results are evaluated for each model and the best accuracy model is used for classification of heartbeats.

Next we implemented a pre-trained model VGG16. This process allows us to use the pre-trained models on datasets with millions of images such as COCO, Imagenet etc. This means instead of creating a model from scratch to solve an equivalent dataset, we can use the model trained on some other dataset as a starting point. The last layer of the network was removed and replaced with custom classifiers i.e. classifying between normal and abnormal. This means we used the network only upto the fully connected layers and removed all successive layers.

Figure 5: Architecture of VGG-16 Model

The architecture for the VGG16 model is shown in figure 5. It is important to note that input dimensions for this model were 150x150 instead of 224x224 as shown in the figure. The architecture for this model is as follows: Each block is stacked with convolution layers and at the end of every block a pooling layer to reduce spatial dimension is applied. The output dimensions are increased by a factor of 2 every block. It should also be noted that zero padding is applied before every convolution layer in each block since it is not shown in the figure. The last three fully connected layers were removed as discussed above.

V. Results

We evaluate our models based on the measures taken like accuracy, recall, precision, loss and F-beta score. We try to minimize the loss function in all the models to achieve better accuracy. Overall, we get a whole simulated model to find the abnormal and normal spectrograms. We ran tests for the machine learning model built using ReLU, ELU, Tanh activation function and the transfer learning model and obtained results for the following measures as shown in the table below.

ML Model	Accuracy	Precision	Recall	F-beta Score	Loss
Model 1 (ReLU)	0.8006	0.7862	0.9175	0.8418	0.5545
Model 2 (ELU)	0.7326	0.6852	0.8875	0.7814	0.6328
Model 3 (tanh)	0.7689	0.7112	0.8913	0.8018	0.6113
Model 4 (Transfer Learning)	0.8119	0.8012	0.9375	0.8651	0.5118

The value above shows the values obtained after we ran tests for the dataset obtained of about three thousand heart recordings. These values show the average of 100 epochs that are run on every model. In this we got the highest result for the model-1 which is for the ReLU activation function.

The graph is plotted for accuracy and loss shown by each model. The accuracy v/s epoch graph has accuracy values on y-axis and epoch values on x-axis. Similarly, for loss v/s epoch graph, loss is plotted on y-axis and epoch on x-axis. It provides an overview of the model behavior for each epoch run on the model and their accuracy rate. The orange line represents the result of the training set and the blue line represents the result of the validation set.

Graph of ReLU Activation Function

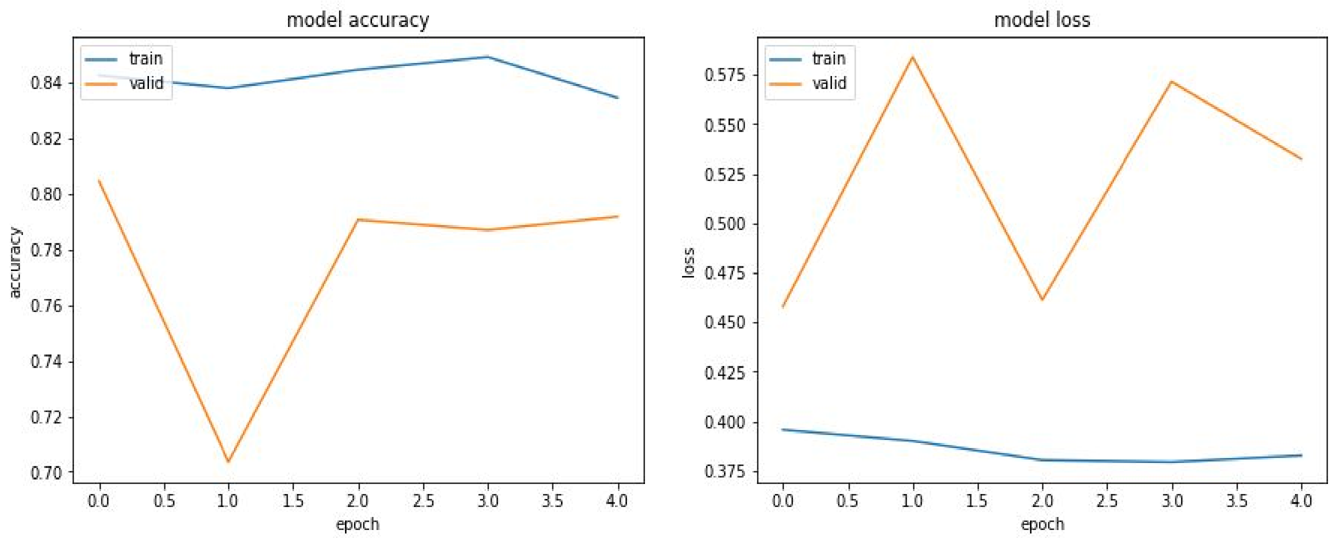

Figure 6: Accuracy and Loss graph of ReLU Activation Function

As shown in figure 6 we can observe that the training accuracy is maintained almost constant between 0.83 and 0.85 with a slight variation. The validation accuracy first decreases and later rises and remains constant throughout. In training loss, we can see constant decrease at 0.375 and the validation loss increase and decrease but it is less than 0.575 which is a good sign.

Graph of ELU Activation Function

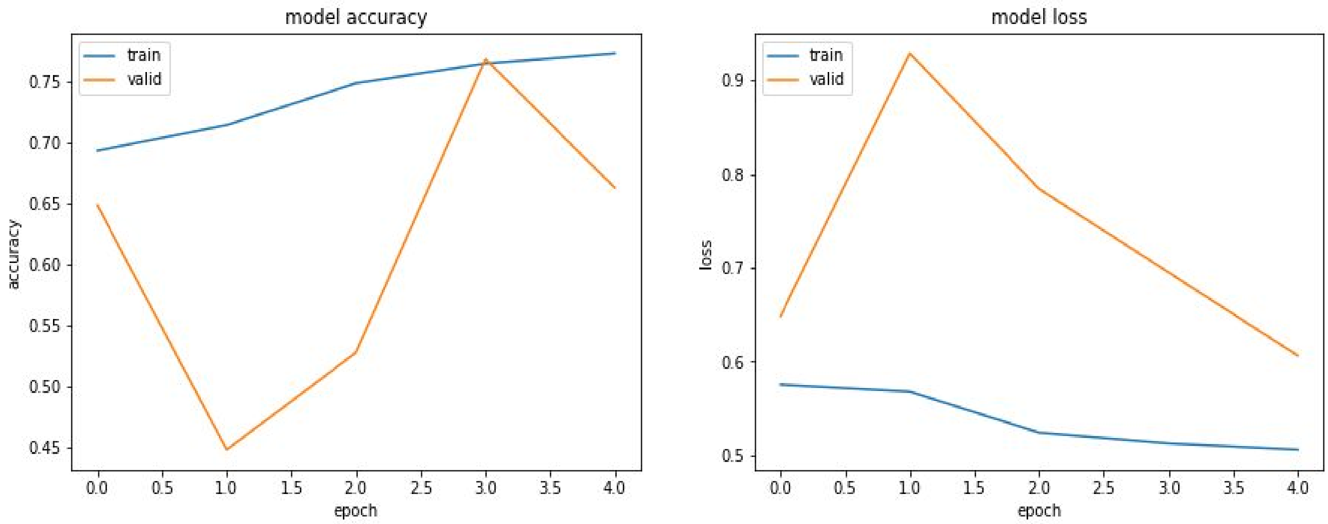

Figure 7: Accuracy and Loss graph of ELU Activation Function

As shown in figure 7 we can observe that the training accuracy increases steadily and validation accuracy has a variation of increase and decrease. The loss in training accuracy is on a decreasing trend and about validation loss it shows a drastic change of increase and then decrease.

Graph of Tanh Activation Function

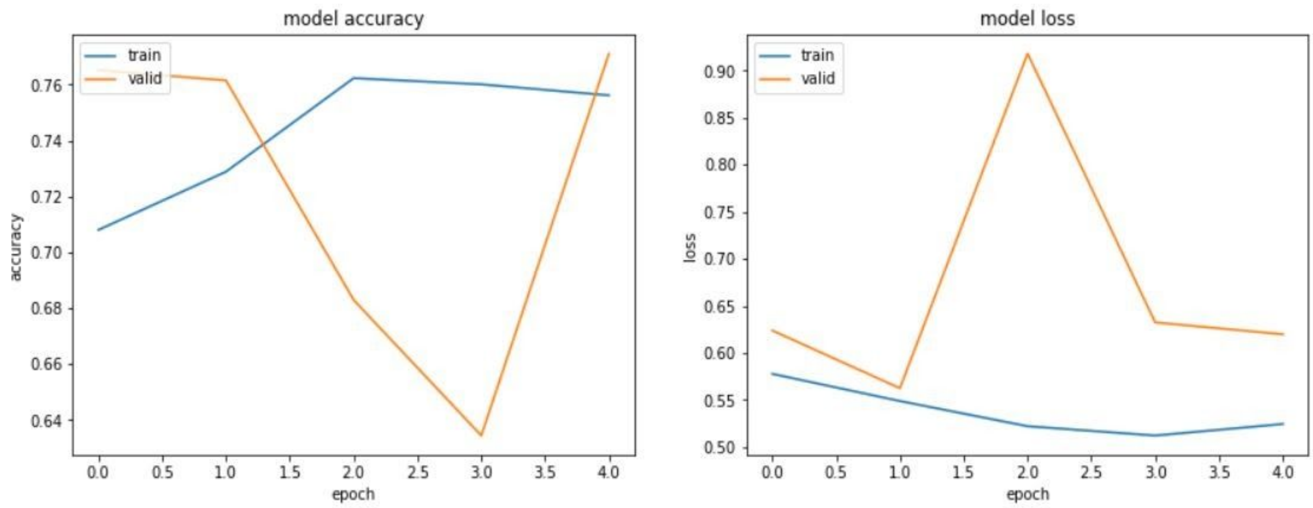

Figure 8: Accuracy and Loss graph of tanh Activation Function

As shown in figure 8 the training accuracy increases upto a point and becomes constant as the validation accuracy has a radical decrease and increase is observed. The training loss decreases upto 0.50 while the validation loss increases to a peak of 0.90 and later decreases.

Graph of Transfer Learning Model

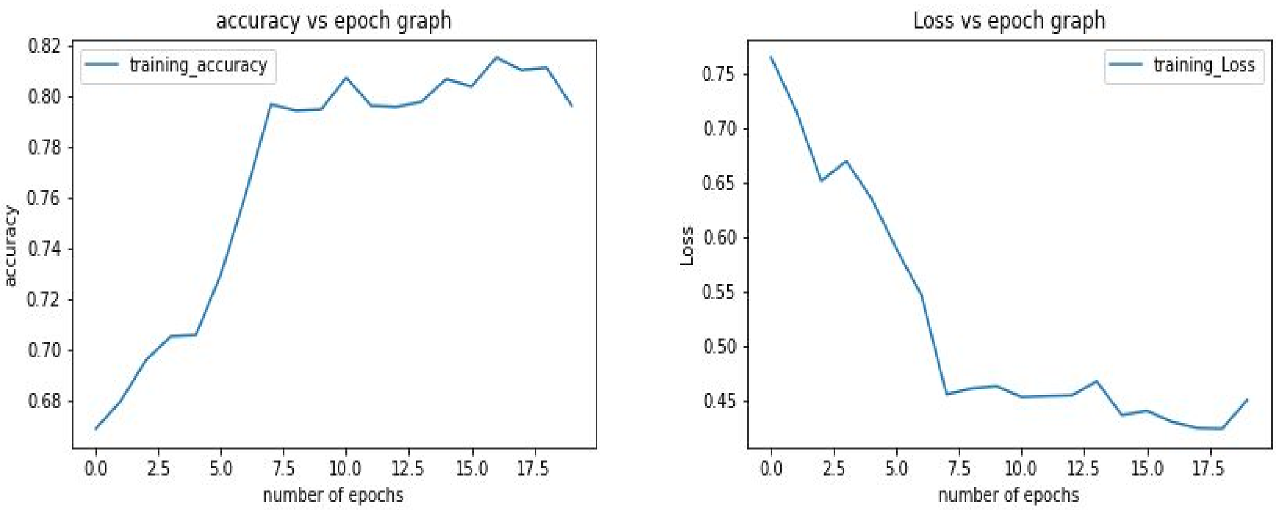

Figure 9: Training Accuracy and Loss graph of VGG-16 Model

As shown in figure 9 the training accuracy increases and is maintained at a constant level while the training loss also decreases as the epochs increase.

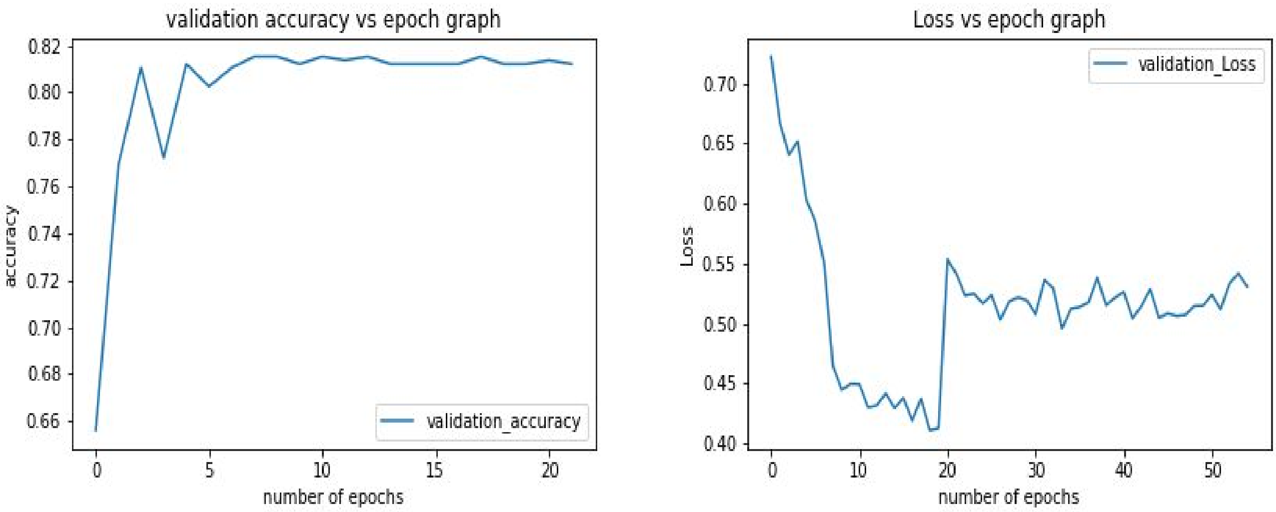

Figure 10: Validation Accuracy and Loss graph of VGG-16 Model

As shown in figure 10 validation accuracy where it increases and maintains a constant level at 0.81. Whereas, the validation loss decreases and has a slight increase later.

Based on the above evaluations, it is obvious that every model has its own merits and flaws. Even though no model could capture the complex features to detect the heartbeats as 'normal' or 'abnormal'. The validation accuracy of 80% and training accuracy of 88% could be considered reasonable. It came close enough to validate the approach that we took while creating the model. It's likely that additional experimentation of fine tuning the model, tweaking and adjusting the hyperparameters could lead to even better results.

VI. Conclusion

The outcome of the paper is to predict the heartbeat given as an input to the machine learning model and to classify it as normal or abnormal. We made use of the dataset which was available as a part of Physio-Net 2016 Cardio challenge. During the training phase of the machine learning models we made use of keras an open-source neural network library written in python and its activation functions such as ReLU (Rectified Linear Unit), ELU (Exponential Linear Unit) and Tanh (Hyperbolic Tangent) and proposed a model to compare their accuracy and precision based on the validation set. Along with this we also implemented, Transfer Learning method to predict the result using the VGG16 architecture which allowed us to use the pre-trained models trained on datasets with millions of images. Based on the observations made and results obtained we got an accuracy of 80.11% from ReLU network of Keras model which is the highest among them and an accuracy of 81.56% from transfer learning model implementation of VGG-16 architecture but when compared with loss VGG-16 performed well than the ReLU network.

References

[1] Weifang Sun, Nianyin Zeng and Yuchao He, “Morphological Arrhythmia Automated Diagnosis Method Using Gray-Level Co-Occurrence Matrix Enhanced Convolutional Neural Network” in IEEE Access, February 2019.

[2] V. Sai Krishna, A. Nithya Kalyani, “Prediction of Cardiac Arrhythmia using Artificial Neural Network” in International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Volume-8, Issue-1S4, June 2019.

[3] Carlos S. Lima, Manuel J. Cardoso, “Cardiac Arrhythmia Detection by Parameters Sharing and MMIE Training of Hidden Markov Models” in 29th Annual International Conference of the IEEE EMB Cité Internationale, Lyon, France, August 23-26, 2007.

[4] Elif Izci, Mehmet Akif Ozdemir, Reza Sadighzadeh, Aydin Akan, “Arrhythmia Detection on ECG Signals by Using Empirical Mode Decomposition” in Proceedings of the third international workshop on advanced issues of e-commerce and web-based information systems, IEEE Proceedings, June 2018.

[5] Ali Isina, Selen Ozdalilib, “Cardiac arrhythmia detection using deep learning” in 9th International Conference on Theory and Application of Soft Computing, Computing with Words and Perception, ICSCCW 2017, 24-25 August 2017, Budapest, Hungary.

[6] Manoj Athreya A, Avani H S, Pooja, Madhu S, K Paramesha, “Detection of Cardiac Arrhythmia Using Machine Learning Algorithms”, International Journal of Recent Technology and Engineering, ISSN: 2277-3878, Volume-8 Issue-4, November 2019.

[7] R Karthik, Dhruv Tyagi, Amogh Raut, Soumya Saxena, Rajesh Kumar M, “Implementation of Neural Network and feature extraction to classify ECG signals”, EP Europace, Volume 21, Issue 8, August 2019.

[8] B. S. Chandra, C. S. Sastry and S. Jana, “Robust Heartbeat Detection from Multimodal Data via CNN-based Generalizable Information Fusion” in Journal of the American Society for Information Science and Technology, 29 June 2019

[9] Anup Das, Francky Catthoor, Siebren Schaafsma, “A rule-based method to model myocardial fiber orientation in cardiac biventricular geometries with outflow tracts” in Communications of the ACM, 13 August 2019

[10] Milad Salem, Shayan Taheri, Jiann-Shiun Yuan, “ECG Arrhythmia Classification Using Transfer Learning from 2-Dimensional Deep CNN Features”, in Communications of the ACM, 21 May 2017.

[11] Sajad Mousavi, Fatemeh Afghah, “Inter- and Intra-Patient ECG Heartbeat Classification for Arrhythmia Detection: A Sequence to Sequence Deep Learning Approach”, ACM arXiv:1812.07421v2, 12 March 2019.

[12] Asim Darwaish, Farid Naït-Abdesselam, “Detection and Prediction of Cardiac Anomalies using Wireless Body Sensors and Bayesian Belief Network”,arXiv:1904.07976v1, ACM proceedings, 16 April 2019.

[13] Rajpurkar, Pranav & Hannun, Awni & Haghpanahi, Masoumeh & Bourn, Codie & Y. Ng, Andrew. “Cardiologist-Level Arrhythmia Detection with Convolutional Neural Networks”, arXiv:1707.01836v1 [cs.CV], ACM proceedings, 6 July 2017.

[14] Kiranyaz S, Ince T, Gabbouj M, “Real-time patient-specific ECG classification by 1-D convolutional neural networks” IEEE Transactions on Biomedical Engineering, Volume 63, pp. 664-675, June 2016.

[15] S. Hong and M. Wu and Y. Zhou and Q. Wang and J. Shang and H. Li and J. Xie, "ENCASE: An Ensemble Classifier for ECG classification using expert features and deep neural networks" Computing in Cardiology, Rennes pp. 1-4, October 2017.